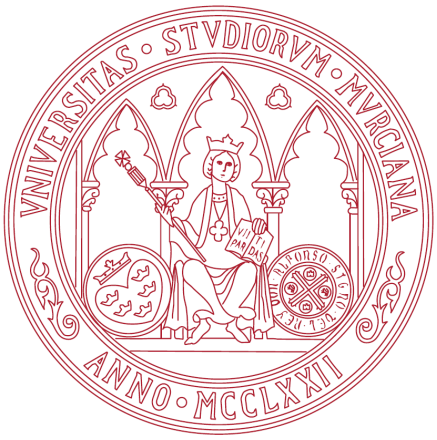

UNIVERSIDAD DE MURCIA

FACULTAD DE INFORMÁTICA

Integración de información biomédica basada en tecnologías semánticas avanzadas

Dña. María del Carmen Legaz García

2015

Integración de información biomédica basada en tecnologías semánticas avanzadas

Tesis doctoral presentada por María del Carmen Legaz García
dentro del Programa de Doctorado en Informática

Dirigida por el Doctor

Jesualdo Tomás Fernández Breis

Departamento de Informática y Sistemas
Facultad de Informática
Universidad de Murcia

2015

Agradecimientos

Primero, quiero dar las gracias a mi director, Jesualdo Tomás Fernández Breis, por darme la oportunidad de comenzar en el mundo de la investigación, por confiar en mí y por toda su ayuda en el desarrollo de esta tesis.

Por aceptarme sin dudar en sus respectivos grupos de investigación y dedicarme su tiempo y atención para enriquecer este trabajo, gracias a Christopher Chute y Cui Tao, de la *Division of Biomedical Statistics and Informatics* en *Mayo Clinic*; a Stefan Schulz y Catalina Martínez, del *Institute for Medical Informatics, Statistics and Documentation* en *Medical University Graz*, y a Ronald Cornet, del *Medical Informatics Department (KIK)* en *AMC*.

Realizar una tesis requiere mucho tiempo en el laboratorio. No me puedo quejar, no creo que se pueda tener mejor compañía y ambiente de trabajo que en nuestro rincón del 1.03 izquierda. Gracias a los actuales *moradores*, Manuel, Miguel, Jojo, Lucia, Mario, Pilar, Omar, Teddy y Astrid; y a todos los que estuvieron en algún momento, Juani, Fran, Consuelo, Cati, Jose y muchos más. Ha sido un placer compartir este camino con vosotros.

Porque de alguna manera están relacionados con esta tesis, gracias a Sophie, Bethany, Justin, Vickram y Tiffany, por hacer de Rochester la gran experiencia que fue; a Cati y Jose (de nuevo), por hacer de Graz más murciano; y a María, Ángel y Sophie, por alegrarme (tren mediante) Ámsterdam.

Gracias a mi madre, que es la que más se preocupa por mí cuando estoy de estancia, y a mi hermana Flori, que es la que más me visita. Y gracias a Jesús, que cree más en mí de lo que ni yo ni nadie creerá jamás.

También agradecer a todo aquel, familia o amigo, que siempre ha tenido palabras de apoyo, y que aun no sabiendo lo que son las *tecnologías semánticas avanzadas*, abrirá esta tesis, sólo porque es mía.

Por último, esta tesis ha sido posible gracias a la Fundación Séneca - Agencia de Ciencia y Tecnología de la Región de Murcia, a través de la Beca-Contrato Predoctoral de Formación de Personal Investigador 15555/FPI/10, así como a través de las Ayudas para Estancias Cortas en Centros Externos 18605/EFPI/12, 18805/EFPI/13 y 19575/EFPI/14.

Índice

Índice de figuras

Índice de Tablas

Bloque I

Introducción y estado del arte

Capítulo 1

Introducción

La informática biomédica se define como el campo científico interdisciplinar que se ocupa del almacenamiento, recuperación, intercambio y uso óptimo de la información biomédica, sus datos y conocimiento para la investigación científica, resolución de problemas y toma de decisiones con el objetivo principal de mejorar la salud humana [1]. El ámbito de la informática biomédica se puede dividir en cuatro áreas de investigación: bioinformática, informática de imágenes biomédicas, informática clínica e informática de la salud pública. Estas cuatro áreas y la colaboración intensiva entre ellas son claves para lograr la práctica de la medicina traslacional, es decir, trasladar resultados científicos del laboratorio a la práctica clínica sobre pacientes específicos y sobre la población, pasando primero por su validación a través de ensayos clínicos [2; 3].

Para conseguir el objetivo final de la medicina traslacional se deben superar una serie de retos que se agrupan en tres barreras traslacionales identificadas: (1) trasladar las innovaciones realizadas en laboratorio a la validación en ensayos clínicos, (2) que estas pruebas lleven finalmente a adoptar esos nuevos métodos en los pacientes y la población, llevando potencialmente a (3) establecer nuevas políticas específicas [4]. La informática biomédica trata de traspasar estas barreras con el uso combinado de las propuestas de sus cuatro subáreas. Las distintas colaboraciones entre estas subáreas se pueden organizar en dos categorías: (1) bioinformática traslacional e (2) informática de la investigación clínica.

La bioinformática traslacional utiliza y extiende los conceptos y métodos de la bioinformática para facilitar trasladar los resultados biológicos de laboratorio a nuevos hallazgos en la práctica clínica [5], es decir, se centra en traspasar la primera barrera. La informática de la investigación clínica se centra en las técnicas de la informática biomédica que permiten traspasar

la segunda y tercera barrera, es decir, llevar las innovaciones de los ensayos clínicos a mejorar el cuidado de los pacientes y llevar con éxito los nuevos hallazgos a la población, integrándolos de forma adecuada en un sistema sanitario eficiente. Se centra en la información en el ámbito de la investigación clínica, que incluye investigación del mecanismo de las enfermedades, intervenciones terapéuticas, ensayos clínicos, desarrollo de nuevas tecnologías, epidemiología, estudios de conducta e investigación de resultados y de servicios sanitarios [6].

El amplio ámbito de aplicación de las ciencias biomédicas hace que la informática biomédica tenga que gestionar una cantidad creciente de datos. Las bases de datos que almacenan información sobre diferentes entidades biológicas crecen a un ritmo constante. Actualmente, según el informe de 2015 de *Molecular Biology Database Collection* [7] existen más de 1500 bases de datos biológicas, las cuales utilizan distintos formatos de representación para los datos biológicos. Por lo tanto, la recuperación y gestión de datos no es fácil para los científicos porque necesitan saber: (1) qué recursos tienen disponibles y contienen la información deseada; (2) cómo esos recursos pueden ser accedidos y consultados; (3) el significado de los tipos de datos y campos usados en cada recurso. Por otro lado, en el ámbito clínico y médico, la llegada de las historias clínicas electrónicas (HCE) contribuye a que haya más datos disponibles para el procesado por ordenador y promueve el uso secundario de los mismos, aunque también genera nuevos problemas. El uso secundario de los datos clínicos incluye actividades como identificación de grupos de estudio clínico, evaluación de la calidad de la asistencia sanitaria, investigación comparativa efectiva, privacidad de datos y desidentificación/reidentificación, metodología de identificación de fenotipos y modelado predictivo [8]. Algunos de estos usos secundarios requieren combinar datos que normalmente están distribuidos entre varios sistemas clínicos, lo que requiere acceso completo, comunicación y entendimiento de la información con independencia de su origen, es decir, interoperabilidad semántica entre dichos sistemas. La falta de interoperabilidad semántica se convierte en una razón de ineficiencia dentro del sistema de salud [9; 10] y tiene un coste de mil millones de dólares en los Estados Unidos anualmente [11].

Por lo tanto, la necesidad de colaboración de las distintas disciplinas de la biomedicina para alcanzar la medicina traslacional requiere formas de representación de la información biomédica que permitan una recuperación y tratamiento de los datos eficiente.

En el ámbito clínico, varios estándares y especificaciones han surgido en los últimos años, entre los que destacan aquellos basados en la arquitectura dual (por ejemplo CEN/ISO 13606 [12], openEHR [13], HL7 [14] y CEM [15]),

donde se distinguen dos niveles de modelado. El modelo de información proporciona los elementos básicos genéricos para estructurar la información de la HCE, mientras que los modelos clínicos especifican escenarios de registro de información mediante la definición de restricciones sobre el modelo de información. Tanto en openEHR como en CEN/ISO 13606, los modelos clínicos se llaman arquetipos y han sido considerados prometedores para compartir datos clínicos de forma escalable y formal [9]. Sin embargo, los modelos clínicos suelen estar representados utilizando lenguajes como ADL, con orientación sintáctica, lo que dificulta la explotación semántica de los mismos. Además, la variedad de estándares disponibles propicia que distintas instituciones utilicen diferentes modelos, con lo que se necesitan soluciones adicionales que permitan la comunicación entre distintos estándares.

En el lado técnico, la Web Semántica [16] describe una nueva forma de contenido web significativo para un ordenador y ha sido propuesta como espacio tecnológico en el que los datos biomédicos pueden ser integrados y explotados [17]. Hay diferentes tecnologías básicas para el éxito de la Web Semántica, entre las que la tecnología clave es la ontología. Una ontología representa una vista común, compartible y reutilizable de un dominio de aplicación [18]. El hecho de que las máquinas conozcan el significado del contenido permite el uso de razonamiento automático en la Web Semántica, lo que permite inferir nueva información o comprobar la consistencia lógica del contenido. Además, la comunidad de la Web Semántica desea alcanzar la Web de Datos [19], que conectaría semánticamente conjuntos de datos distribuidos por Internet. Más específicamente, la iniciativa de Linked Open Data [20] persigue la publicación y compartición de conjuntos de datos biomédicos usando formatos semánticos.

Existen varias propuestas de uso de tecnologías de la Web Semántica en la representación y gestión de la información biomédica. Numerosas ontologías biomédicas han sido propuestas, siendo Gene Ontology [21] una de las más utilizadas para la anotación de resultados biológicos, además de surgir iniciativas que regular la creación de nuevas ontologías, OBO Foundry [22], y repositorios de gestión de dichas ontologías como BioPortal [23], que contiene más de 400 ontologías biomédicas y vocabularios controlados. Esfuerzos recientes promueven el uso de tecnologías semánticas para la representación de datos biomédicos, por ejemplo la plataforma RDF del EBI [24] o Bio2RDF [25]. Propuestas de representación de modelos clínicos y datos de los estándares de HCE han demostrado su utilidad para la interoperabilidad entre dichos estándares [26; 27] y la validación de los modelos clínicos [28].

La mayoría de propuestas de representación semántica de datos realizan una transformación de los datos específica para cada caso de uso, lo que

requiere un esfuerzo por parte de las instituciones ya que necesitan tener conocimientos avanzados en tecnologías y procedimientos de transformación. Existen herramientas para facilitar la generación automática o manual de conjuntos de datos biomédicos en formatos semánticos, sin embargo, la generación automática simplemente realiza una transformación sintáctica del formato tradicional, mientras que las herramientas que proporcionan una transformación semi-automática o manual son poco flexibles y están muy centradas en formatos de entrada concretos.

Por todo ello, en esta tesis se proponen soluciones para la transformación semántica de conjuntos de datos biomédicos guiada por una arquitectura ontológica, lo que permitirá obtener una representación semántica correcta del conjunto de datos de entrada. El proceso definido es genérico y no depende del formalismo usado para la captura de los datos. Este proceso forma parte de una metodología de integración que permite integrar conjuntos de datos heterogéneos para su explotación combinada, creando repositorios semántico y conjuntos de datos abiertos siguiendo los principios de Linked Open Data. Finalmente, una plataforma de gestión de información biomédica permite la gestión, integración y explotación de modelos, datos clínicos y recursos biomédicos externos, haciendo uso de representaciones semánticas de los recursos y de los métodos de transformación e integración.

1.1 Organización del documento

El documento se divide en cuatro bloques principales constituidos de una serie de capítulos. En el bloque I se ofrece una visión general del estado del arte de aquellos aspectos relevantes para esta tesis y define los objetivos que se persiguen. En el bloque II se describen las aportaciones resultado del trabajo en esta tesis, para cada aportación se describirá el método creado, las herramientas desarrolladas y la validación de los mismos. El bloque III ofrece las conclusiones finales de la tesis, una discusión de aspectos relevantes del trabajo realizado, una descripción de posibles futuras vías de trabajo y el listado de publicaciones científicas y contribuciones a congresos derivadas de la tesis. Por último, el bloque IV incluye un resumen en inglés de toda la tesis.

Los seis primeros capítulos pertenecen al bloque I. El capítulo 1, en el que nos encontramos, proporciona una breve introducción a la tesis y su organización.

El capítulo 2 ofrece una visión sobre los formatos de representación de la información biomédica. El capítulo se estructura en la información estrictamente clínica y la información procedente de la bioinformática. En la primera

parte se introducen los estándares y especificaciones de la HCE basados en la arquitectura de dos niveles, y por lo tanto, se pone atención en la definición de modelos clínicos en las distintas propuestas de estandarización y en las herramientas de gestión de modelos y datos clínicos existentes. En la segunda parte del capítulo aparecen los formatos más comunes de representación de los conjuntos de datos biológicos, y se nombran algunas de las bases de datos más relevantes. Al final del capítulo se comentan las terminologías, detallando algunas de las más utilizadas en la codificación de información biomédica, como es SNOMED-CT.

El capítulo 3 comenta el espacio tecnológico de la Web Semántica, destacando RDF, ontologías, el lenguaje OWL y las propuestas de Linked Data. Además, presenta una introducción a técnicas de ingeniería ontológica, buenas prácticas en la construcción y reutilización de ontologías y metodologías útiles en la gestión de ontologías y contenido semántico, como es la comprobación de similitud y la anotación. La última parte del capítulo presenta el uso de las tecnologías de la Web Semántica en el ámbito biomédico, y se presentan ontologías ampliamente utilizadas, como Gene Ontology, propuestas como OBO Foundry, BioPortal, recursos biomédicos presentes en la Web de Datos y diferentes propuestas y proyectos de aplicación de ontologías en estándares de HCE.

El capítulo 4 presenta la necesidad de la transformación semántica de información, muestra los pasos comunes de las metodologías, técnicas y lenguajes de definición de alineamientos y herramientas existentes.

El capítulo 5 describe las propuestas más comunes de integración de datos, la aplicación de la web semántica a los procesos de integración y técnicas relacionadas, como el alineamiento de ontologías, además de la definición de los problemas más comunes relacionados con estos procesos.

El capítulo 6 y último del bloque I describe los objetivos de la tesis y la metodología de investigación seguida.

El bloque II está formado por cuatro capítulos. El capítulo 7 presenta un modelos de transformación para la obtención de repositorios semánticos a partir de conjuntos de datos de entrada en formatos tradicionales. Presenta los componentes del modelo, así como diferentes casos de aplicación del mismo. La última parte del capítulo presenta la herramienta que implementa este modelo de transformación.

En el capítulo 8 se presenta un modelo de integración de recursos heterogéneos basado en la aplicación del modelo de transformación a un modelo de salida ontológico. El capítulo presenta el diseño del modelo de integración, las características del modelo de salida global y de los procesos de transformación e integración.

El capítulo 9 presenta la plataforma de gestión, explotación e integración de información biomédica. Se presentan todos los métodos para la gestión de modelos y datos clínicos, la utilización de los métodos presentados en los dos capítulos anteriores y diferentes usos de los repositorios creados. La última parte del capítulo presenta la interfaz de la plataforma creada.

El capítulo 10 presenta los distintos escenarios en los que los resultados de esta tesis han sido utilizados, así como un caso de uso completo de gestión de información biomédica.

El bloque III está constituido por un único capítulo 11, en el que se agrupan las conclusiones finales, la discusión del trabajo realizado, las posibles vías futuras y el listado de publicaciones científicas y contribuciones a congresos derivados de este trabajo.

El bloque IV incluye el capítulo 12 con un resumen breve de la tesis en inglés.

Capítulo 2

Representación de información biomédica

La investigación biomédica tiene un amplio ámbito de aplicación, lo que supone que la informática biomédica debe trabajar con datos generados de forma continuada y creciente. En el ámbito clínico, el uso creciente de la historia clínica electrónica (HCE) hace que los datos clínicos de un paciente estén distribuidos entre distintos sistemas. Por otro lado, el número de bases de datos bioinformáticas que se ponen a disposición de los investigadores crece de forma constante, las cuales son producidas y gestionadas por cada institución, con lo que tienen la característica de ser heterogéneas. La necesidad de colaboración de las distintas disciplinas de la biomedicina para alcanzar la medicina traslacional requiere intercambiar estos datos entre los distintos dominios de aplicación y por lo tanto la búsqueda de formas de representación que permitan una recuperación y tratamiento de los datos eficiente.

En este capítulo se presentan distintos tipos y formas de representación de información biomédica, distinguiendo entre las dos principales categorías de la informática biomédica. Por una lado la informática de la investigación clínica, concretamente la investigación clínica sobre pacientes específicos y relacionada con el cuidado de su salud, cuya información relacionada se representa por medio de estándares de historia clínica electrónica (HCE) y por otro lado la información biológica y médica utilizada en bioinformática traslacional. En general todos los tipos de representación aquí expuestos tratan de mejorar la integración de información y su interoperabilidad semántica, definiendo ésta como la capacidad de los sistemas de compartir, interpretar y explotar información independientemente del origen de la misma [29].

2.1 Historia Clínica Electrónica

La historia clínica tradicional en papel surgió en el siglo XIX como un cuaderno de laboratorio personalizado para cada paciente [30]. En él, los médicos podían registrar sus observaciones y planificaciones, y usarlo como recordatorio de detalles importantes en la siguiente sesión con sus pacientes. En las siguientes décadas este historial intentó adaptarse a los nuevos requisitos surgidos de los cambios en asistencia sanitaria y medicina. Sin embargo, hoy en día se considera a esta historia tradicional totalmente inadecuada para cumplir las necesidades de la medicina moderna, debido entre otras cosas a su incapacidad para soportar la comunicación entre distintos profesionales médicos.

La informatización de la historia clínica de los pacientes surge para dar solución a los problemas asociados al historial clínico tradicional. Esta versión informatizada se conoce como Historia Clínica Electrónica (HCE), se define como un repositorio de información, relacionada con la asistencia sanitaria de un paciente, en formato digital, almacenado de forma segura y que permite el acceso a usuarios autorizados. La necesidad de acceder de forma integrada a la información clínica completa de un paciente, la cual puede estar distribuida entre distintos sistemas de información, cada uno utilizando su propia representación, ha provocado que desde el principio de la década de los 90 se realizara una inversión considerable en la investigación y desarrollo de sistemas de información clínica, con el propósito de lograr una representación estándar e interoperable de la HCE. En general, los proyectos de investigación surgidos en diferentes países tienen los siguientes objetivos en común [31]:

- Involucrar a los pacientes en el uso de su HCE.
- Definir el tipo de información a representar en los historiales.
- Seleccionar estándares y vocabularios de representación.
- Desarrollar una infraestructura y políticas de seguridad.
- Producir sistemas de HCE abiertos, estandarizados e interoperables para fomentar su intercambio y facilitar su gestión.

El proyecto europeo *Good European Health Record* (GEHR) [32] fue un proyecto de tres años financiado por el programa *European Health Telematics research*, con la participación de 7 países europeos y que dio como resultado el desarrollo de una arquitectura de historia clínica de modelo dual que constituye la base de los actuales estándares CEN/ISO 13606 y openEHR.

2.1.1 Arquitectura del modelo dual

La arquitectura del modelo dual [33] surgió como contraposición a la arquitectura de un nivel en la que se basan la mayoría de los sistemas de información.

En la arquitectura de un nivel las entidades de negocio se modelan directamente en modelos software y de base de datos a través del modelado de casos de uso u otras técnicas de desarrollo software. Los sistemas basados en esta arquitectura crean la información como instancias de las entidades de negocio, y la guardan, transmiten y le dan formato para que sea manejable por los usuarios. Tanto la base de datos, como el software y la interfaz gráfica de usuario se desarrollan siguiendo un enfoque orientado a objetos o relacional, describiendo formalmente la semántica del sistema de información. Esta aproximación es adecuada cuando la complejidad del dominio es baja y no requiere continuos cambios en su definición. Sin embargo, en sistemas de información que manejan dominios más complejos y bajo constante cambio, como es el caso del dominio clínico, la aproximación de un nivel conlleva problemas, por ejemplo, los dominios muy grandes son complejos de definir ya que se requiere que todos los conceptos estén en el modelo final, y la interoperabilidad es difícil de conseguir ya que esta requeriría que cada sistema involucrado en la comunicación haga tanto sus modelos como su software compatibles con los del resto de sistema.

La alternativa es la arquitectura de modelo dual, donde en lugar de capturar toda la información requerida en un modelo de datos enorme, se define una separación clara entre información y conocimiento. En esta arquitectura la información se estructura siguiendo un modelo de referencia, que representa conceptos genéricos y estables, mientras que el conocimiento se representa usando un modelo de conocimiento. En el dominio de las HCE, el modelo de referencia contiene los conceptos genéricos que estructuran la información contenida en la historia clínica, mientras que el modelo de conocimiento se utiliza para definir escenarios de registro de información clínica restringiendo las estructuras definidas por el modelo de referencia. El uso del término arquetipo se ha extendido en algunos estándares y especificaciones para referirse a estos escenarios, por lo que también utilizan el término modelo de arquetipos para referirse al modelo de conocimiento. Esta separación permite representar con el modelo de referencia los conceptos genéricos y estables del dominio, mientras que los arquetipos se definen combinando y restringiendo las entidades del modelo de información. De esta forma, en los sistemas de HCE, las bases de datos y el software solo dependen de los conceptos genéricos y estables del modelo de información y cualquier actualización en la naturaleza de la información solo requiere la actualización de los arquetipos,

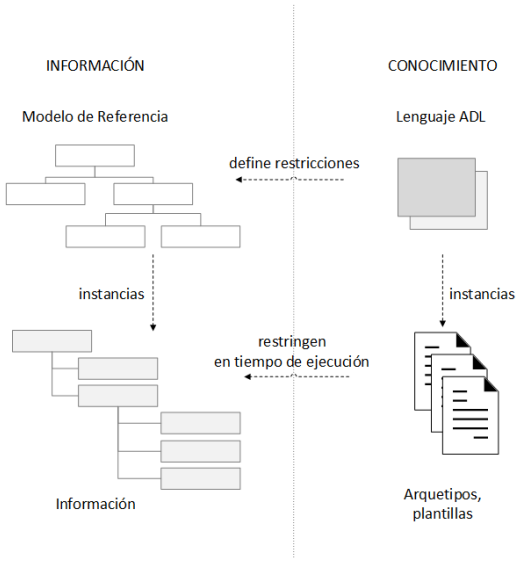

Figura 2.1: Meta-arquitectura del modelo dual

haciendo a los sistemas flexibles a los cambios. Además, la separación permite que el sistema se construya de forma rápida por expertos en sistemas de información mientras que los arquetipos son desarrollados por expertos clínicos.

Esta arquitectura provoca nuevas relaciones entre información y modelos, como se puede ver en la figura 2.1 [34]. En esta figura, la información (abajo-izquierda) representa instancias de un modelo de objetos (arriba-izquierda), al igual que ocurre en los sistemas de información tradicionales. Sin embargo, en el modelo de dos niveles, la semántica del dominio la proporcionan los arquetipos (abajo-derecha). Los arquetipos se definen formalmente en un lenguaje de arquetipos genérico y se relacionan con el modelo de referencia de forma que su semántica viene dada a través de la restricción de las clases definidas en éste. Así, los datos también están definidos conforme a la definición del modelo de referencia.

En las últimas décadas se han hecho grandes esfuerzos para desarrollar estándares de HCE basados en la arquitectura del modelo dual. En las siguientes secciones presento en más detalle algunos de estos estándares y especificaciones.

2.1.2 Estándares y especificaciones

2.1.2.1 ISO 13606

El estándar CEN/ISO 13606 [12] es una norma europea desarrollada por el Comité Europeo de Normalización (CEN) aprobada como estándar ISO internacional. Esta norma especifica el diseño interno y el modo en el que se debe comunicar la HCE para dar soporte a la interoperabilidad semántica basándose en la arquitectura de modelo dual. El objetivo de este estándar es definir una arquitectura de información rigurosa y estable para comunicar parte o toda la HCE de un paciente a cualquier sistema, repositorio o aplicación que necesite acceder o proporcionar dichos datos. Algunos proyectos y desarrollos en los que se hace uso de este estándar pueden consultarse en [35]. Este estándar se compone de cinco partes. Las dos primeras, (1) modelo de referencia y (2) modelo de arquetipos, corresponden a los dos niveles de la arquitectura dual. También define (3) arquetipos de referencia y lista de términos, (4) seguridad y (5) especificación de interfaz.

- El modelo de referencia contiene las estructuras básicas genéricas para representar cualquier información de la HCE.
- El modelo de arquetipos permite definir conceptos clínicos utilizando artefactos llamados arquetipos. Estos permiten combinar y restringir las entidades del modelo de referencia para definir los conceptos clínicos. También define un lenguaje para la representación y comunicación de arquetipos denominado ADL.
- Arquetipos de referencia y lista de términos define una lista de términos que dan valor a los atributos del modelo de referencia. Además muestra ejemplos de cómo representar información clínica codificada usando los modelos de referencia de openEHR y HL7 versión 3.
- Seguridad describe la metodología a seguir para controlar el acceso a los datos de la HCE.
- La especificación de la interfaz proporciona la arquitectura de comunicación para proporcionar los datos de la HCE a sistemas y servicios.

El modelo de referencia se divide en cuatro paquetes: extracto, demográfico, soporte y tipos primitivos. El paquete extracto (EXTRACT) define los componentes que forman la HCE:

- Extracto de HCE (EHR_EXTRACT): clase contenedor de toda o parte de la HCE de un paciente.

- Carpeta (FOLDER): organización de más alto nivel de la HCE, suele representar episodios clínicos o especialidades médicas. Permite definir jerarquías opcionales de carpetas y contiene componentes de tipo composición.

- Composición (COMPOSITION): representa el conjunto de anotaciones asociadas a una única sesión clínica o documento. Las composiciones se organizan en carpetas.

- Sección (SECTION): encabezamientos que agrupan las entradas asociadas a una única sesión clínica (COMPOSITION) y que ayudan a mejorar la presentación y navegación por la información. Permiten definir jerarquías.

- Entrada (ENTRY): información registrada en la HCE como resultado de una acción clínica, una observación, una interpretación clínica o un propósito clínico. También se conoce como declaración clínica.

- Clúster (CLUSTER): es una estructura de datos que permite organizar la información en series, tablas o árboles.

- Elemento (ELEMENT): representa el nivel más bajo de la jerarquía de la HCE. Contiene un único valor que debe ser instancia de alguno de los tipos de datos definidos por el estándar.

La figura 2.2 muestra cómo se relacionan todos estos componentes. Las clases principales que se utilizan para construir la jerarquía de datos de la HCE dentro de un EHR_EXTRACT son de tipo RECORD_COMPONENT. Esta es una clase abstracta que es superclase de todos los nodos concretos en la jerarquía de la HCE: FOLDER, COMPOSITION, SECTION, ENTRY, CLUSTER, ELEMENT, y también superclase de otras clases abstractas: CONTENT e ITEM.

El modelo de referencia cumple con los requisitos publicados en ISO/TS 18308 [36] para asegurar que se preserva el significado de las HCE cuando se comunican entre sistemas clínicos heterogéneos.

2.1.2.2 OpenEHR

La fundación openEHR [13] es una fundación independiente sin ánimo de lucro creada por investigadores y socios industriales con experiencia en el proyecto GEHR [32]. Ha creado una serie de especificaciones para sistemas de HCE influyendo con su trabajo significativamente en otros estándares

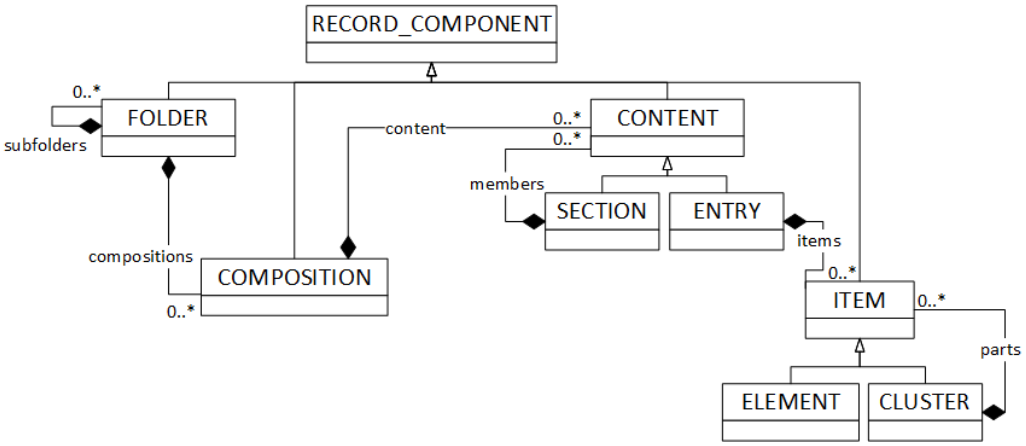

Figura 2.2: Fragmento del modelo de referencia de ISO 13606

como CEN 13606 y HL7. OpenEHR apuesta por el modelo dual como arquitectura de HCE y comparte con CEN/ISO 13606 el modelo de arquetipos. La especificación openEHR define tres paquetes principales: (1) modelo de referencia, (2) modelo de arquetipos y (3) modelo de servicios. Como ya se ha dicho, el modelo de arquetipos es el mismo que en CEN/ISO 13606 y el modelo de servicios incluye definiciones de los servicios básicos en el entorno de la informática médica centrados en la HCE. El modelo de referencia, del que podemos ver un fragmento en la figura 2.3 se componen de los siguientes modelos de información:

- Soporte (Support): describe los conceptos más básicos requeridos por el resto de modelos de información. Se compone de los paquetes Definiciones (Definitions), Identificación (Identification), Terminología (Terminology) y Medida (Measurement). La semántica que definen permite al resto de modelos de información usar identificadores y tener acceso a servicios de conocimiento como terminologías y otros datos de referencia.

- Tipos de datos (Data Types): conjunto de tipos de datos definidos para ser usados por el resto de modelos de información.

- Estructuras de datos (Data Structures): define las estructuras genéricas de tipo tabla (ITEM_TABLE), lista (ITEM_LIST), árbol (ITEM_TREE), elemento único (ITEM_SINGLE) e historia (HISTORY).

- Común (Common): define varios conceptos que se reutilizan en los demás paquetes del modelo de información, como LOCATABLE, AR-

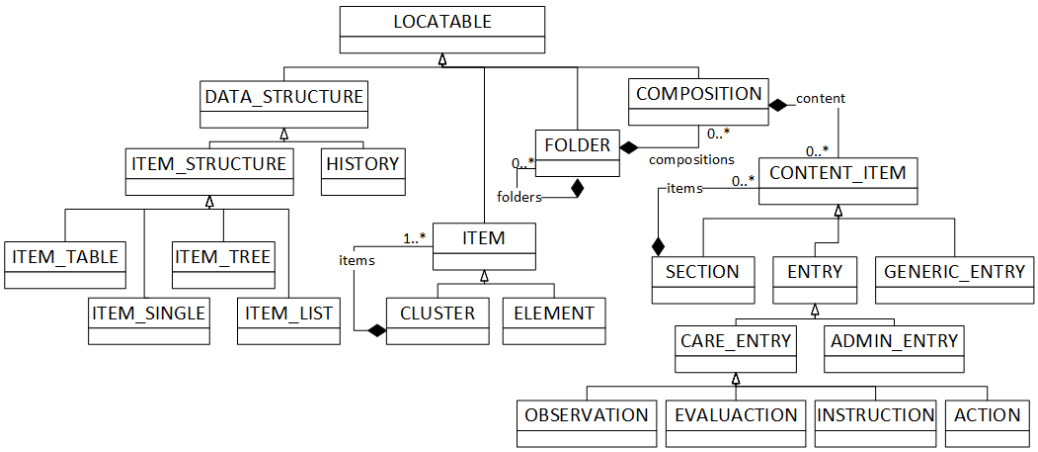

Figura 2.3: Fragmento del modelo de referencia de openEHR

CHETYPED, ATTESTATION, PARTICIPATION, etc.

- Seguridad (Security): define la semántica de control de acceso y privacidad para la información contenida en la HCE.

- EHR: define la semántica y el contexto de los conceptos COMPOSITION, SECTION y ENTRY, que se corresponden con las clases del mismo nombre en ISO EN 13606.

- Extracto EHR (EHR Extract): define cómo se construye un extracto EHR a partir de COMPOSITION, datos demográficos y de control de acceso.

- Integración (Integration): define la clase GENERIC_ENTRY usada para representar datos externos o en formato libre como una estructura de árbol.

- Demográfico (Demographics): define conceptos genéricos para identificar al paciente, sus datos de contacto, etc.

- Workflow: modelo de información marcado como trabajo futuro, define modelos para describir la semántica de procesos, como aquellos que son resultado de la ejecución de guías clínicas.

2.1.2.3 HL7

Health Level Seven (HL7) [14] es una organización internacional de desarrollo de estándares. Su objetivo principal es la interoperabilidad de la información clínica y administrativa. Para ello crea estándares para el intercambio, manejo e integración de HCE. HL7 comenzó como un estándar de mensajería en el que la interoperabilidad de los datos del paciente se basaba en el intercambio de mensajes. Sin embargo, carencias identificadas en este estándar, como la ausencia de un modelo de datos consistente, la falta de metodologías formales y de roles de aplicación y usuario bien definidos, así como su poca precisión, llevaron a HL7 a definir HL7 v3.

HL7 v3 es una familia de estándares que cuenta con un modelo de información, Reference Information Model (RIM), que es la raíz de todos los modelos de información y estructuras desarrolladas como parte del proceso de desarrollo del estándar. El proceso de desarrollo del estándar HL7 v3 es una metodología dirigida por modelos en la que se desarrolla una red de modelos relacionados entre sí que describen los aspectos estáticos y de comportamiento de los requisitos y diseño de los estándares HL7, así como la semántica subyacente y las reglas de negocio que los gobiernan.

RIM proporciona una vista estática de la información necesaria para los estándares HL7 v3. Incluye clases y diagramas de máquinas de estado y se acompaña de guiones gráficos, modelos de interacción, modelos de tipos de datos, y modelos de terminologías entre otros. Todo esto proporciona una vista completa de los requisitos y diseño de los estándares HL7. Las clases, atributos, máquinas de estado y relaciones de RIM se usan para derivar modelos de información específicos del dominio que se transforman por medio de un conjunto de procesos de refinamiento de restricciones a un modelo estático de la información de un estándar HL7. A través de las reglas de gobierno se obtienen los modelos de información derivados de RIM que se refinan para obtener especificaciones de estándares HL7. Las reglas requieren que a partir de los modelos derivados sea posible obtener la parte de RIM de la que se obtuvieron y que su semántica y reglas de negocio sean consistentes respecto a las de RIM. Todos los estándares de la familia HL7 v3 obtienen su información y contenido semántico de RIM.

El núcleo principal de RIM, utilizado para expresar el contenido clínico y administrativo de la atención sanitaria, se compone de seis clases principales (ver figura 2.4):

- *Act*: representa las acciones que son ejecutadas y que deben documentarse durante la gestión y administración de la atención sanitaria.
- *Participation*: expresa el contexto de una acción, como quién la realiza,

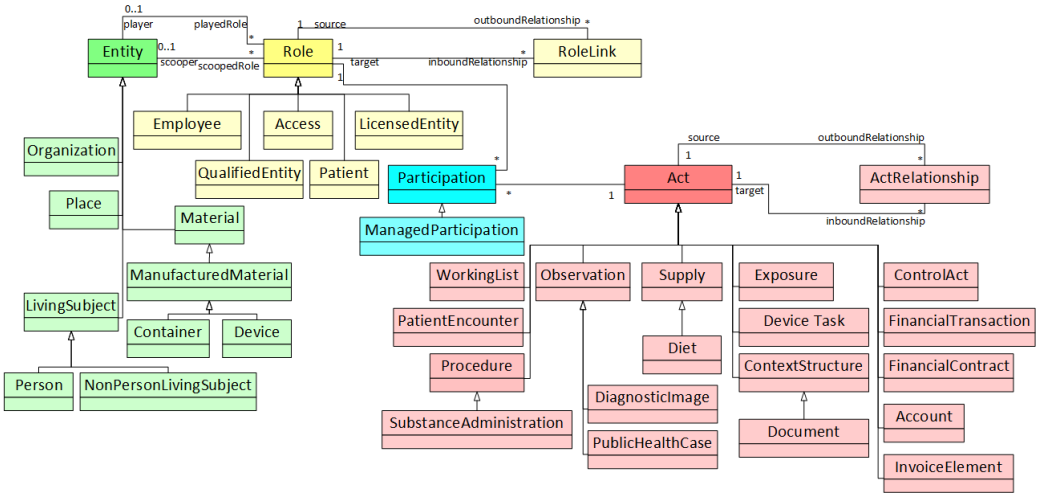

Figura 2.4: Fragmento del modelo de referencia RIM

para quién la realiza, dónde se realiza, etc.

- *Entity*: representa las cosas y personas que son de interés y forman parte de la asistencia clínica, por ejemplo, una persona o una organización.
- *Role*: establece el papel de las entidades que participan en acciones de la asistencia sanitaria.
- *ActRelationship*: representa la relación de una acción con otra, como la relación entre una orden de observación y el evento de observación que se produce como consecuencia.
- *RoleLink*: representa una dependencia entre dos *Roles*.

Uno de los estándares que forma parte de la familia de HL7 v3 es CDA [37], un estándar que especifica la estructura y semántica de documentos clínicos. Un documento CDA es un objeto de información definido y completo que puede incluir texto, imágenes, y otros contenidos multimedia. El objetivo de este estándar es promover el intercambio de la información clínica, ha sido creado con el deseo de codificar en mayor profundidad las afirmaciones clínicas textuales encontradas en los informes clínicos, y hacerlo de forma que permita la comparación de contenido entre documentos creados en diferentes sistemas de información [38].

Un documento puede ser enviado dentro de un mensaje HL7 y puede existir independientemente de un mensaje. Los componentes principales de la

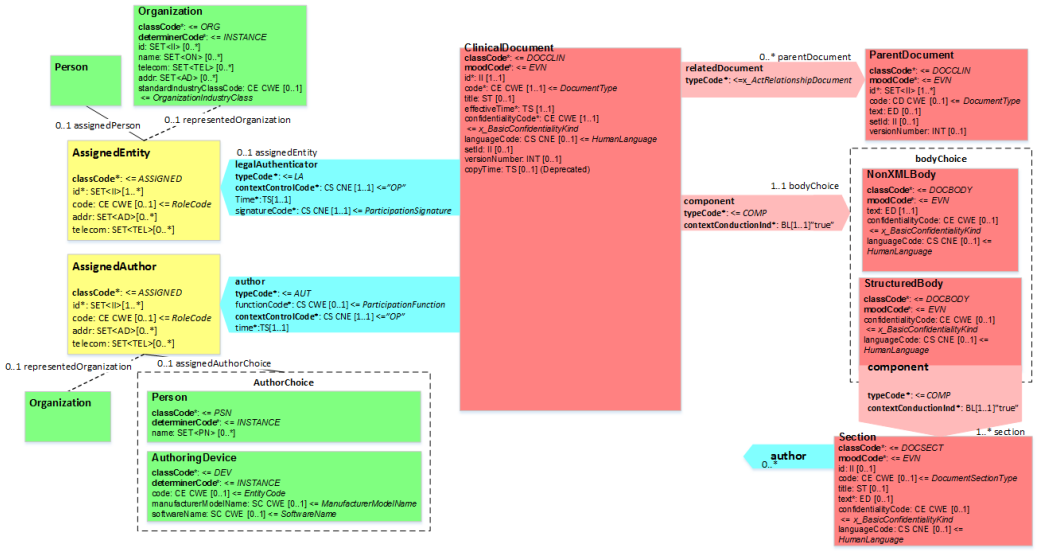

Figura 2.5: Fragmento del modelo de objetos de CDA v2 mostrando una porción de la cabecera y sus relaciones con el cuerpo del documento

especificación CDA son la cabecera (header) y el cuerpo (body). La figura 2.5 [38] muestra parte de la cabecera y su relación con el cuerpo del documento. La cabecera expresa el contexto en el cual el documento es creado, identifica y clasifica el documento. Contiene información sobre el paciente, el autor del documento, la creación del documento, de autenticación, etc. El cuerpo contiene la información clínica detallada, organizada en secciones y cuyo contenido puede ir codificado usando vocabularios estandarizados.

CDA tiene tres niveles de definición de documentos, siendo el nivel 1 el que menos estructura proporciona, y los niveles 2 y 3 los que proporcionan más estructura. El nivel 1 se compone de cabecera, cuerpo y datos sin estructura, que pueden ser PDF, un fichero .doc, o una imagen escaneada. En el nivel 2 la definición del documento puede ser una sección XML, cada una con un código identificativo. En el nivel 3 dentro de cada sección, entradas específicas son identificadas a través de su codificación con alguna terminología. Así pues, la arquitectura CDA abarca desde datos no estructurados hasta datos clínicos totalmente codificados.

Los documentos CDA se definen en XML y el esquema CDA se obtiene a partir del modelo de objetos de CDA R2 (CDA release 2). El modelo de objetos CDA R2 es un diagrama técnico de la especificación CDA, el cual se deriva a partir de RIM.

Fast Healthcare Interoperability Resources (FHIR) [39] es una nueva ge-

neración de estándares de HL7 para el intercambio de información clínica de forma electrónica. FHIR se basa en los estándares de HL7 v2, HL7 v3 y CDA. Trata de simplificar la implementación sin sacrificar la integridad de la información, y hace uso de modelos lógicos y teóricos existentes para proporcionar un mecanismo de intercambio de información entre aplicaciones clínicas.

Los componentes básicos de FHIR son recursos. Todo contenido intercambiable se define como un recurso, y todos los recursos comparten un conjunto común de metadatos, una sección legible para humanos, y una forma común de ser definidos y representados, construyéndose a partir de tipos de datos que definen patrones de elementos comunes y reutilizables.

La filosofía de FHIR es construir una conjunto base de recursos que por sí solos o combinados, satisfagan la mayoría de casos de uso comunes. Los recursos pretenden definir los contenidos de información y estructuras para el conjunto de información núcleo que se comparte entre la mayoría de implementaciones. Mientras que en HL7 v3, el modelado está basado en restricciones, en FIHR se basa en composición. Aunque en FIHR la mayoría de recursos y tipos de datos provienen de RIM y los tipos de datos de ISO, como en HL7 v3, algunos de sus recursos manejan contenido que está fuera del ámbito de RIM y algunos tipos de datos han sufrido cambios que no están soportados por el modelo de tipos de datos de HL7 v3.

2.1.2.4 CEM

Clinical Element Model (CEM) [15] es una propuesta de la empresa Intermountain Healthcare para representar modelos clínicos detallados a la que más tarde se unió la compañía GE Healthcare/Caradigm. El objetivo de esta propuesta es asegurar representación, interpretación e intercambio de datos entre fuentes heterogéneas de forma inequívoca. Esta estrategia de modelado ha sido adoptada dentro del Strategic Health IT Advanced Research Project (SHARP) [40], enfocado a mejorar la calidad, seguridad y eficiencia de la atención sanitaria a través de las tecnologías de la información. Dentro de este proyecto, el área cuatro pretende aumentar la seguridad y mejorar los resultados médicos de los pacientes a través del uso secundario de la HCE. SHARP usa CEM como estrategia de modelado global para representar modelos detallados de datos clínicos junto a las instancias de datos que conforman a esos modelos. CEM también sigue la arquitectura de dos niveles, está formado por el modelo abstracto de instancias que proporciona la estructura genérica para representar datos clínicos y el modelo abstracto de restricciones, que define cómo restringir este modelo para definir modelos clínicos específicos. Los modelos clínicos que se definen con la estrategia CEM

se clasifican en una categoría estructural básica, que recoge los atributos comunes en un modelo de clases.

El modelo abstracto de instancias o modelo de instancias define la estructura básica para representar modelos clínicos. La figura 2.6 muestra la estructura de este modelo. Se compone de las siguientes partes:

- Type: su valor es definido por el modelo abstracto de restricciones, actúa como identificador del modelo clínico además de indicar a qué categoría estructural básica pertenece.
- Key: indica un código controlado que describe lo que el modelo clínico intenta definir.
- Value choice: representa el cuerpo del modelo clínico. Puede estar formado por un tipo de dato simple (propiedad *data*) o por otro modelo clínico (propiedad *items*). CEM utiliza los tipos de datos de HL7 v3.
- Qualifiers y modifiers: modelos clínicos que modifican el significado del modelo clínico al que pertenecen. Un modelo clínico actuando de *qualifier* añade información adicional al modelo que lo contiene mientras que si actúa como *modifier*, modifica el significado del modelo.
- Attribution: se utiliza para definir una acción y el quién, dónde, por qué y cuándo de esa acción.

Un modelo clínico CEM utiliza el modelo abstracto de restricciones para definir las restricciones necesarias sobre los atributos del modelo de instancias. Así, se puede restringir el atributo *Key* a tener un código específico y un valor dentro de un dominio concreto, el atributo *Data* a un tipo de dato concreto, o la cardinalidad del atributo *Item*. Todas las restricciones se agrupan en una colección nombrada que crean un tipo de modelo clínico. Estos tipos se identifican por un nombre, pueden ser creados a partir de otro tipo definido y todos pertenecen a una categoría funcional. Las categorías funcionales disponibles son las siguientes:

- Statement: representa una aseveración completa sobre algún aspecto, característica o condición del paciente. Un modelo clínico restringido a *statement* tiene significado por sí solo en el historial del paciente. Existen dos tipos de *statement*, (1) *simple statement*, cuyo valor es un tipo de datos simple, es decir en el cuerpo *value choice* utiliza la propiedad *data* Data y (2) *compound statement*, cuyo cuerpo está formado por una colección de modelos clínicos, es decir, en el *value choite* utiliza la propiedad *items*.

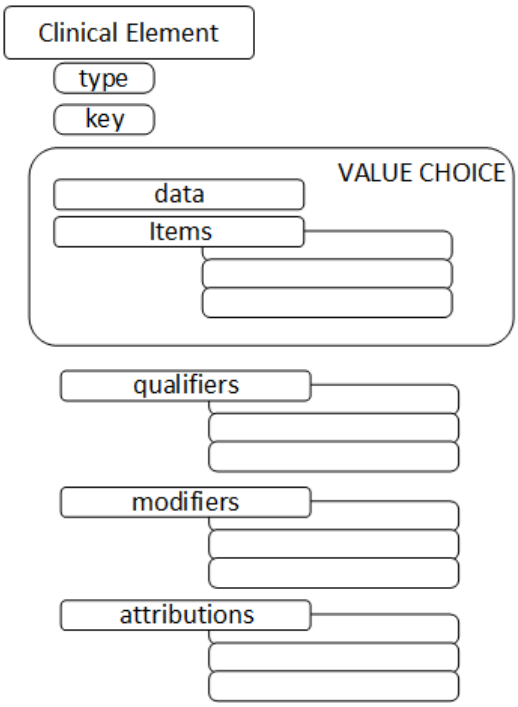

Figura 2.6: Modelo abstracto de instancia de CEM

- Component: representa un modelo clínico que solo tiene sentido dentro de otro modelo clínico de tipo *statement* o *association*, es decir, no tiene entidad por sí solo en el historial clínico del paciente. Un ejemplo sería una fecha o un aparato de medición. Existen dos tipos de *component*, (1) *simple component*, cuyo valor es un tipo de dato simple y (2) *compound component*, cuyo valor es una colección de otros *components*. Un componente también puede ser usado como *qualifier* de otro modelo clínico, añadiendo información adicional al mismo.

- Modifier: esta categoría es equivalente a *component* pero añade la restricción que son modelos clínicos que solo pueden ser usados como modificadores (*modifiers*) de otra instancia.

- Attribution: al igual que *modifier*, esta categoría es equivalente a *component* pero solo puede ser utilizada como *attribution* de otro modelo clínico.

- Por último existen asociaciones (*association*) que representan colecciones de *statement* y *component*, estas asociaciones pueden ser *panel*, *collection*, *semantic link* y *annotations*.

- Panel y Collection: representan una agrupación de entradas clínicas. Es una colección de *statement* o de *association*. La diferencia entre las dos es que *panel* representa una relación muy fuerte entre los componentes que lo forman mientras que *collection* representa una relación más débil.

- Semantic Link: es una asociación que representa una relación semántica fuerte entre dos o más modelos clínicos. Los modelos clínicos involucrados en un *semantic link* tienen un rol definido en la relación.

- Annotation: es una asociación que añade información adicional a un modelo clínico sin modificar su significado.

2.1.3 Modelos clínicos

Los modelos clínicos se utilizan para especificar escenarios de registro de información clínica siguiendo un modelo de conocimiento. Por ejemplo, pueden ser utilizados para registrar datos clínicos sobre una prueba de laboratorio, la medida de la presión sanguínea, la prescripción de un medicamento, etc. Los modelos clínicos se construyen de forma similar en los distintos estándares y especificaciones de HCE, restringiendo las entidades del modelo de información o modelo de referencia. Las diferencias entre los modelos clínicos de distintos estándares se deben al formalismo utilizado para especificar los modelos, el contenido y estructura del modelo de información y el tipo de restricciones que utilizan. Para cada estándar o especificación aquí revisado, los datos clínicos se representan siguiendo la representación de los modelos clínicos, y comúnmente se almacenan como ficheros XML.

2.1.3.1 Arquetipos en CEN/ISO 13606 y openEHR

CEN/ISO 13606 y openEHR comparten el modelo de arquetipos, donde los arquetipos se definen como restricciones sobre el modelo de información.

Los arquetipos se definen a través del lenguaje Archetype Description Language (ADL) [41]. Un documento ADL es un fichero de texto estructurado, cuya estructura es independiente de un dominio o estándar particular. Esta flexibilidad permite que la misma estructura sintáctica pueda utilizarse para especificar arquetipos basados en distintos modelos de referencia.

ADL utiliza dos sintaxis, cADL, para la definición de restricciones, y dADL, para la definición de datos. Un arquetipo en ADL incluye cuatro secciones principales, cabecera (*header*), descripcion (*description*), definición (*definition*) y ontología (*ontology*). La sintaxis cADL se utiliza en la parte definición del arquetipo y la sintaxis dADL en el resto.

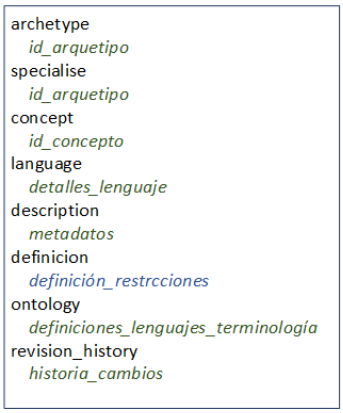

Figura 2.7: Estructura de un arquetipo en ADL

La figura 2.7 muestra gráficamente las partes de un arquetipo. Las cuatro primeras forman la cabecera del arquetipo, e indican el nombre del arquetipo (*archetype*), si se trata de la especialización de otro (*specialise*), el concepto del dominio que modela (*concept*) y el idioma original del arquetipo, así como las traducciones disponibles (*language*). La siguiente sección (*description*) proporciona detalles del autor del arquetipo y la organización a la que pertenece, detalles de uso recomendado del arquetipo, prohibiciones, palabras clave y estado de desarrollo. La sección definición (*definition*) es la única escrita en cADL y es la que expresa las restricciones sobre el modelo de referencia. Todos los conceptos incluidos en esta sección pueden codificarse utilizando un sistema propio o una terminología disponible en la comunidad científica. La sección ontología (*ontology*) proporciona descripciones textuales para cada elemento que aparece en la sección *definition* y para cada enlace hacia otra terminología. También se incluyen las traducciones de cada elemento en los diferentes idiomas del arquetipo. En la última sección (*revision_history*) se puede incluir la historia de su evolución.

La figura 2.8 muestra el fragmento de la definición ADL de un arquetipo openEHR para registrar una reacción alérgica a una sustancia. La cabecera incluye el nombre del arquetipo, *openEHR-EHR-EVALUATION.adverse_reaction.v1*, el concepto clínico que representa, *Adverse reaction*, y el idioma original (inglés). En la sección *description* se da información del autor, fecha, y detalles como palabras clave que definen al arquetipo, (*reaction, allergy, allergic, adverse,...*). En la sección *definition* se describe la reacción adversa (*EVALUATION[at000]*) como una jerarquía de elementos (*ITEM_TREE[at0001]*). Los elementos registran la sustancia que ha provocado la reacción (*ELEMENT[at0002]*) y la información de la reacción alérgica

(*CLUSTER[at0009]*), compuesta por un conjunto de elementos que para cada sustancia específica (*ELEMENT[at0010]*) informan cómo se ha manifestado la reacción (*ELEMENT[at0011]*) y el tipo de reacción (*ELEMENT[at0016]*). Finalmente en la sección *ontology* se muestra la definición de los conceptos que aparecen en la parte de definición del arquetipo.

```
archetype (adl_version=1.4)
    openEHR-EHR-EVALUATION.adverse_reaction.v1
concept
    [at0000]    -- Adverse Reaction
language
    original_language = <[ISO_639-1::en]>
description
    original_author = <
        ["name"] = <"Heather Leslie">
        ["date"] = <"2010-11-08">
    >
    details = <
        ["en"] = <
            keywords = <"reaction", "allergy", "allergic", "adverse",...>
    >>
    ...
definition
 EVALUATION[at0000] matches {    -- Adverse Reaction
  data matches {
   ITEM_TREE[at0001] matches { -- Tree
    items cardinality matches {1..*; unordered} matches {
     ELEMENT[at0002] matches {   -- Substance/Agent
      value matches {
       DV_TEXT matches {*}
     }}
     CLUSTER[at0009] occurrences matches {0..*} matches {    -- Reaction Event
      items cardinality matches {1..*; unordered} matches {
       ELEMENT[at0010] occurrences matches {0..1} matches {--Specific Substance/Agent
        value matches {
         DV_TEXT matches {*}
       }}
       ELEMENT[at0011] occurrences matches {0..*} matches { -- Manifestation
         value matches {
          DV_TEXT matches {*}
       }}
       ELEMENT[at0016] occurrences matches {0..1} matches { -- Reaction Type
        value matches {
         DV_TEXT matches {*}
       }}
       ...
       }}}}
       ...
    }
ontology
 term_definitions = <
  ["en"] = <
    items = <
     ["at0000"] = <
      text = <"Adverse Reaction">
      description = <"A harmful or undesirable, unexpected effect associated with exposure
      to any substance or agent, or a medication at therapeutic or sub-therapeutic doses.">
     >
      ["at0002"] = <
         text = <"Substance/Agent">
          description = <"Identification of a substance, agent, or a class of substance,
          that is considered to be responsible for the Adverse Reaction.">
     >...
 >>>
```

Figura 2.8: Fragmento de arquetipo openEHR para registrar una reacción alérgica

La figura 2.9 muestra el fragmento de la definición ADL del arquetipo CEN/ISO 13606 para registrar una reacción alérgica a una sustancia. La parte de cabecera es similar al arquetipo en openEHR, incluye el nombre del arquetipo, *CEN-EN13606-ENTRY.adverse.v1*, el concepto clínico que representa, *Adverse reaction*, y el idioma original (inglés). En la sección *description* se da información del autor, fecha, y detalles como palabras clave que definen al arquetipo, (*reaction, allergy, allergic, intolerance*). En la sección *definition* se describe la reacción adversa (*ENTRY [at0000]*) como una entrada estructurada de *ELEMENTs* y *CLUSTERs*. Se registra la sustancia que ha provocado la reacción (*ELEMENT [at0003]*) y los detalles de la reacción alérgica (*CLUSTER [at0019]*), como la sustancia específica (*ELEMENT[at0032]*), la categoría de la reacción (*ELEMENT[at0015]*), la severidad de la reacción (*ELEMENT[at0023]*) y una descripción textual de la reacción (*ELEMENT[at0022]*). Algunos de los elementos se recogen como entradas textuales, pero otros se indican como valores codificados, por ejemplo, la categoría de la reacción se codifica localmente, con los términos "at0016", "at0017", "at0018", "at0030", los cuales se definen en la parte ontología del arquetipo. En este caso "at0016" corresponde a la categoría de intolerancia (*Intolerance*).

```
archetype
        CEN-EN13606-ENTRY.adverse.v1
concept
        [at0000] -- Adverse reaction
language
        original_language= <[ISO_639-1::en]>
description
    original_author = <
                ["date"] = <"23/04/2006">
    >
    details = <
        ["en"] = < keywords= <"reaction", "allergic", "allergy", "intolerance">>>
...
definition
 ENTRY [at0000] occurrences matches {1..1} matches { -- Adverse reaction
  items matches {
   CLUSTER [at0002] occurrences matches {1..1} matches
   {--Structure
    parts cardinality matches {0..*; unique} matches {
     ELEMENT [at0003] occurrences matches {1..1} matches { -- Agent
      value matches {SIMPLE_TEXT occurrences matches {1..1} matches {*}}
     }
     CLUSTER [at0019] occurrences matches {0..*} matches { -- Reaction detail
      parts cardinality matches {1..*; unique} matches {
       ELEMENT [at0032] occurrences matches {0..1} matches { -- Specific substance
        value matches {SIMPLE_TEXT occurrences matches {1..1} matches {*}}
       }
       ELEMENT [at0015] occurrences matches {0..1} matches { -- Reaction category
        value matches {
         CODED_TEXT occurrences matches {1..1} matches {
          codedValue matches {
           CD  matches {
            codeValue matches {"at0016", "at0017", "at0018", "at0030"}
             codingSchemeName matches {"local"}
       }}}}}
       ELEMENT [at0023] occurrences matches {0..1} matches { -- Reaction severity
        value matches {
         CODED_TEXT occurrences matches {1..1} matches {
          codedValue matches {
           CD  matches {
            codeValue matches {"at0024", "at0025", "at0026"}
             codingSchemeName matches {"local"}
       }}}}}
       ELEMENT [at0022] occurrences matches {1..1} matches { -- Reaction description
        value matches {SIMPLE_TEXT occurrences matches {1..1} matches {*}}
        }
 }}}}}}
ontology
    term_definitions = <
        ["en"] = <
            items = <
                ["at0016"] = <
                    text = <"Intolerance">
                    description = <"Leads to unpleasant symptoms which are sufficient to
                    avoid use in the future">
                >
    >>>
```

Figura 2.9: Fragmento de arquetipo CEN/ISO 13606 para registrar una reacción alérgica

2.1.3.2 Modelos clínicos CEM

Los modelos CEM comenzaron definiéndose en CEML (Clinical Element Modeling Language), un lenguaje que utiliza XML como sintaxis, pero actualmente se definen utilizando CDL (Constraint Definition Language), un lenguaje con sintaxis propia creado por la compañía GE Healthcare.

Los modelos CEM definidos en CDL se componen de una cláusula de *copyright*, cláusulas *import*, comentario cabecera, nombre del modelo y cuerpo del modelo. La cláusula *copyright* es un comentario e indica los derechos de copia del modelo. Un CEM puede hacer referencia a otros modelos CEM, bien extendiéndolos o restringiéndolos, o utilizándolos como *qualifier*, *modifier*, *attribution* o *item*. Cuando esto ocurre se debe definir una cláusula *import* por cada modelo CEM al que se hace referencia. Después de la cláusulas *import*, el comentario cabecera da información del uso del modelo, descripción textual del mismo, autores, fecha de creación, versión, notas y otros comentarios. Por último el nombre del modelo y el cuerpo definen el modelo CEM específico.

La figura 2.10 muestra el modelo CEM en CDL para registrar una alergia. Después del comentario indicando el *copyright* del modelo, las cláusulas *import* añaden al modelo CEM el resto de modelos a los que éste hace referencia en la definición del modelo. El comentario cabecera informa de que el modelo es *Allergy*, su utilización, autor y fecha de creación. En la definición del modelo *model* se da nombre al mismo, *Allergy*, y se indica que es una extensión del modelo existente *PrescribingGuidance*, el cual ha sido incluido con una cláusula *import*. La figura 2.11 muestra el modelo CEM CDL para *PrescribingGuidance*. Como vemos, se utiliza como modelo padre para documentar alergias, intolerancias y terapias fallidas. *PrescribingGuidance* se define como un modelo de tipo *statement*, por lo que *Allergy* también lo es, en concreto se trata de un *compound statement*. El cuerpo del modelo en *Allergy* está formado por cuatro *items*, que indican la sustancia que ha provocado la alergia *Substance*, la categoría a la que pertenece dicha sustancia *SubstanceCategory*, las reacciones que esta sustancia ha provocado *ReactionToSubstance* y los ingredientes que compone la sustancia *Ingredient*. Después añade calificadores que aportan más información, como la fecha en que comenzó el problema *DateOfLastOccurrence*, la fecha en la que se resolvió *DateOfResolution*, si el paciente está informado de su problema *PatientInformedInd*, y en caso de que se produzca, la causa de muerte *CauseOfDeathInd*. El modelo permite incluir en el registro de una alergia un modificador *NegationInd*, que en caso de estar presente niega lo registrado por el modelo, es decir, la existencia de la alergia. Por último, un campo *status* permite indicar el estado del registro, este campo obtiene sus valores de la variable GuidanceStatus_VALUESET_ECID, perteneciente a una terminología local, y cuyos

valores pueden ser *Active, Deleted, Inactive, Ruled Out*. El hecho de que este modelo extienda a *PrescribingGuidance* hace que incluya todos los modelos definidos en este último.

```
/*
 *  Copyright © 2009-2011 General Electric Company
 *  All Rights Reserved
 */
import ReactionToSubstance;
import SubstanceCategory;
import PrescribingGuidance;
import Substance;
import DateOfLastOccurrence;
import DateOfResolution;
import PatientInformedInd;
import CauseOfDeathInd;
import NegationInd;
import Ingredient;

/**
 *  Allergy
 *  Allergy is for documenting true allergies.
 *  @author CTilley
 *  @createdate 11/26/2008
*/
model Allergy extends PrescribingGuidance {
  key code(Allergy_KEY_ECID);
  item Substance substance card(1);
  item SubstanceCategory substanceCategory card(1);
  item ReactionToSubstance reactionToSubstance card(0-M);
  item Ingredient ingredient card(0-M);
  qualifier DateOfLastOccurrence dateOfLastOccurrence card(0..1);
  qualifier DateOfResolution dateOfResolution card(0..1);
  qualifier PatientInformedInd patientInformedInd card(0..1);
  qualifier CauseOfDeathInd causeOfDeathInd card(0..1);
  modifier NegationInd negationInd card(0..1);
  status domain (GuidanceStatus_VALUESET_ECID);
}
```

Figura 2.10: Modelo clínico CEM para registrar una alergia

```
/*
 *  Copyright © 2009-2011 General Electric Company
 *  All Rights Reserved
 */
import Documented;
import ReportedReceived;
import Verified;
import Uncertainty;
import Subject;
import Comment;
import Aggregate;

/**
 *  PrescribingGuidance
 *  PrescribingGuidance is the top node CEM for documenting Prescription guidance.
 *  This will be restricted to allow documentation of allergies (allergy),
 *  intolerances (intolerance), failed therapies (therapy)and pharmacogenetics.
 *
 */
partial model PrescribingGuidance is statement {
  key domain(PrescribingGuidance_KEY_VALUESET_ECID);
  qualifier Aggregate aggregate card(0..1);
  modifier Subject subject card(0..1);
  modifier Uncertainty uncertainty card(0..1);
  attribution Verified verified card(0..1);
  attribution ReportedReceived reportedReceived card(0..1);
  attribution Documented documented card(0..1);
}
```

Figura 2.11: Modelo clínico CEM Prescribing Guidance

2.1.3.3 Plantillas CDA de HL7

En el estándar CDA de HL7 se utilizan plantillas (templates) para restringir la especificación de CDA (el modelo de objetos CDA R2) dentro de una implementación particular y para proporcionar conjuntos de reglas válidas que comprueben la consistencia de las restricciones. Los documentos CDA cumplen las restricciones de una plantilla para representar los datos clínicos. La figura 2.12 muestra el documento que registra una alergia siguiendo la plantilla *Allergy Intolerance*. En esta plantilla se definen las alergias y las intolerancias como tipos especiales de problemas, por lo que se registran como elementos CDA de tipo *observation*. Después se indican los identificadores de las plantillas utilizadas para modelar el documento. El identificador "1.3.6.1.4.1.19376.1.5.3.1.4.6" corresponde a la plantilla *Allergy Intolerance*, esta plantilla cumple con las restricciones de la plantilla *Problem Entry*, cuyo identificador es "1.3.6.1.4.1.19376.1.5.3.1.4.5", que a su vez cumple con las restricciones de la plantilla *Problem Observation*, cuyo identificador es "2.16.840.1.113883.10.20.1.28". Los documentos CDA que sigan la plantilla

Allergy Intolerance deben tener un campo <code>, que representa el tipo de sustancia o agente que provoca la alergia, y si el documento establece la existencia o no de una alergia o una intolerancia. La plantilla *Allergy Intolerance* puede contener varios <entryRelationship>, que se utilizan para describir eventos adversos provocados por la alergia; registrar la severidad de la alergia; registrar el estado del problema alérgico, si está activo, en remisión, resuelto, etc.; todo esto se registra mediante entradas <observation>que deben cumplir con la plantilla *Problem Entry*. Un <entryRelationship>también se utiliza para hacer comentarios, utilizando la entrada <act>. El documento de la figura 2.12 incluye los campos obligatorios de la plantilla *Problem Entry*, <id>, para asignar un identificador a la observación específica recogida; el tiempo efectivo <effectiveTime>, que indica el intervalo de tiempo en el que la observación es válida, el valor <low>es el momento más temprano en el que es conocida la existencia de la condición, mientras que el valor <high>indica el momento en el que la observación deja de ser cierta; y por último el elemento <text>se utiliza para descripciones textuales del problema, incluyendo fechas y comentarios.

```
<?xml version="1.0" encoding="UTF-8"?>
<observation xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="urn:hl7-org:v3"
xsi:schemaLocation="urn:hl7-org:v3 CDA.xsd" classCode="OBS" moodCode="EVN">
  <templateId root="2.16.840.1.113883.10.20.1.28"/>
  <templateId root="1.3.6.1.4.1.19376.1.5.3.1.4.5"/>
  <templateId root="1.3.6.1.4.1.19376.1.5.3.1.4.6"/>
  <id root="2055171060"/>
  <code codeSystem="2.16.840.1.113883.5.4" codeSystemName="ObservationIntoleranceType"/>
  <text/>
  <statusCode code="completed"/>
  <effectiveTime>
    <low value="2011"/>
    <high value="2011"/>
  </effectiveTime>
  <entryRelationship>
    <observation classCode="OBS" moodCode="EVN">
      <templateId root="2.16.840.1.113883.10.20.1.28"/>
      <templateId root="1.3.6.1.4.1.19376.1.5.3.1.4.5"/>
      <templateId root="2.16.840.1.113883.10.20.1.54"/>
      <id root="840816419"/>
      <code codeSystem="2.16.840.1.113883.6.96" codeSystemName="SNOMEDCT"/>
      <text/>
      <statusCode code="completed"/>
      <effectiveTime>
        <low value="2011"/>
        <high value="2011"/>
      </effectiveTime>
    </observation>
  </entryRelationship>
  <entryRelationship>
    <observation/>
  </entryRelationship>
  <entryRelationship>
    <observation/>
  </entryRelationship>
  <entryRelationship>
    <act/>
  </entryRelationship>
</observation>
```

Figura 2.12: Ejemplo CDA para una intolerancia alérgica

2.1.3.4 Recurso FHIR

Los recursos en FHIR pueden ser representados en formato XML o JSON. La figura 2.13 muestra el recurso para representar una reacción adversa específica a una sustancia. El recurso registra uno o varios identificadores para el registro. Después recoge información general, fecha en la que ocurre la reacción, datos del paciente (utilizando un recurso "Paciente") y datos de la persona que realiza el registro de la reacción adversa (utilizando otro recurso). Los siguientes dos campos dan información detallada de la reacción, concretamente los síntomas registrados y la sustancia que los provoca. Para los síntomas se indica un campo código y la severidad de la reacción, para el código se utiliza la terminología ICD-10, mientras que para la severidad se utilizan un conjunto de valores de FHIR (severa, seria, moderada, menor).

```
<AdverseReaction xmlns="http://hl7.org/fhir">
 <!-- from Resource: extension, modifierExtension, language, text, and contained -->
 <identifier><!-- 0..* Identifier External Ids for this adverse reaction --></identifier>
 <date value="[dateTime]"/><!-- 0..1 When the reaction occurred -->
 <subject><!-- 1..1 Resource(Patient) Who had the reaction --></subject>
 <didNotOccurFlag value="[boolean]"/><!-- 1..1 Indicates lack of reaction -->
 <recorder><!-- 0..1 Resource(Practitioner|Patient) Who recorded the reaction --></recorder>
 <symptom>  <!-- 0..* What was reaction? -->
  <code><!-- 1..1 CodeableConcept E.g. Rash, vomiting --></code>
  <severity value="[code]"/><!-- 0..1 severe | serious | moderate | minor -->
 </symptom>
 <exposure>  <!-- 0..* Suspected substance -->
  <date value="[dateTime]"/><!-- 0..1 When the exposure occurred -->
  <type value="[code]"/><!-- 0..1 drugadmin | immuniz | coincidental -->
  <causalityExpectation value="[code]"/><!-- 0..1 likely | unlikely | confirmed | unknown -->
  <substance><!-- 0..1 Resource(Substance) Presumed causative substance --></substance>
 </exposure>
</AdverseReaction>
```

Figura 2.13: Ejemplo de recurso Adverse Reaction en FHIR

Para la sustancia se indica la fecha en la que el paciente estuvo expuesta a ella; el tipo de exposición, utilizando un conjunto de valores de FHIR (administración de medicamento, inmunización, por coincidencia); la probabilidad de que el tipo de exposición cause una reacción, con un conjunto de valores de FHIR (probable, improbable, confirmada ,desconocida); y se indica la sustancia usando un recurso específico para la misma.

2.1.4 Gestión de información clínica

Los arquetipos clínicos se consideran prometedores en la consecución de la interoperabilidad semántica dentro de la HCE [9]. Sin embargo, no existen demasiadas herramientas que permitan explotar arquetipos y datos basados en los mismos en entornos de interoperabilidad semántica. En esta sección veremos algunas de las herramientas existentes.

2.1.4.1 Clinical Knowledge Manager

La herramienta Clinical Knowledge Manager (CKM) [42] es un repositorio y aplicación cuyo objetivo principal es el desarrollo colaborativo, gestión y publicación de recursos de conocimiento. Los recursos de conocimiento actuales de CKM se componen de arquetipos, plantillas y conjuntos de terminologías orientados a historias clínicas y conjuntos de terminologías.

CKM (ver figura 2.14) proporciona un entorno de gobernanza y publicación, en el cual las versiones, ciclo de vida, dependencias, meta-datos y recursos de conocimiento pueden ser gestionados de forma coherente. La aplicación

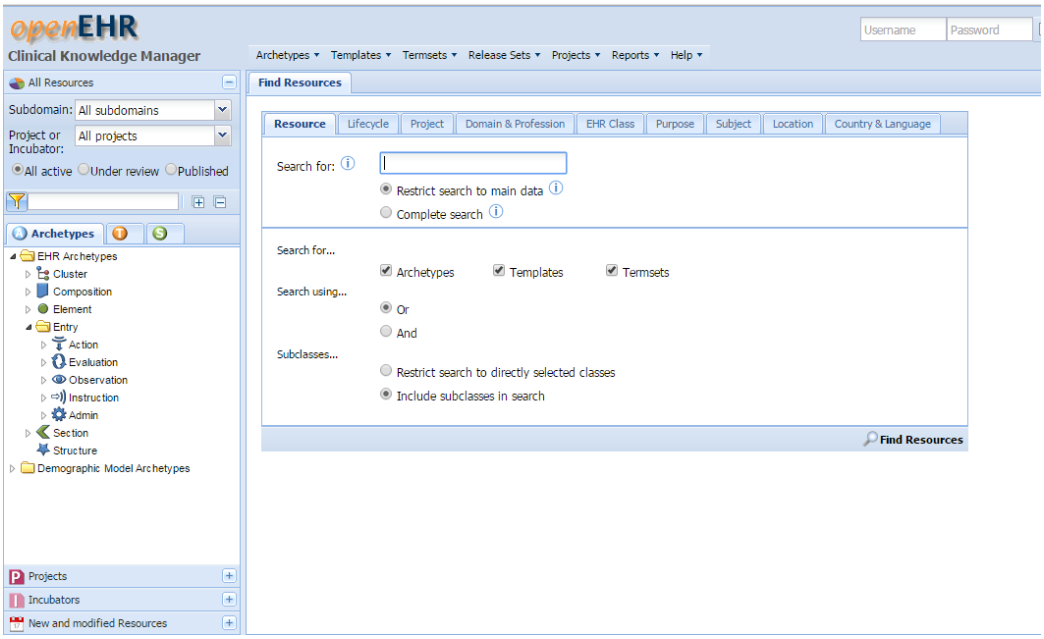

Figura 2.14: Interfaz de Clinical Knowledge Manager

está dirigida a profesionales médicos y expertos en informática médica, que pueden trabajar de forma colaborativa en el desarrollo de nuevos recursos y realizar revisiones exhaustivas de los mismos. El trabajo de gestionar versiones y notificaciones lo realiza la aplicación, mientras que los usuarios son los que se ocupan de la creación y gestión y evaluación de nuevos recursos.

El entorno de gestión de CKM permite crear subdominios y proyectos. Los subdominios permiten organizar los recursos por librerías u organizaciones que los utilicen. Los proyectos se usan para facilitar la colaboración formal, publicación, distribución y mantenimiento de los recursos principales de CKM para un propósito clínico o escenario específico y se crean bajo un subdominio.

La aplicación permite los roles de usuario, revisor, editor y administrador, cada uno con las siguientes funciones:

Un usuario puede:

- Obtener una vista general de todos los arquetipos a través de un navegador, de un diagrama de mapa mental y de un buscador que permite buscar por contenido textual en los arquetipos.
- Visualizar un arquetipo concreto en formato ADL, HTML o visualizar su mapa mental, y descargarlo.
- Consultar el estado de un arquetipo, que puede variar desde draft (bo-

rrador), cuando es un arquetipo nuevo; team review (en revisión), cuando el proceso de revisión se inicia; published (publicado), cuando el equipo de revisores considera que se ha llegado a una versión final; obsolete (obsoleto), cuando un arquetipo utilizado ya no debe ser utilizado.

- Consultar el historial de revisión de un arquetipo.
- Ofrecerse voluntario para la revisión de un arquetipo en estado borrador.
- Crear lista de alertas de arquetipos para realizar un seguimiento de los mismos.
- Crear discusiones o contribuir a las discusiones sobre el desarrollo de arquetipos.
- Recibir notificaciones cuando se crea un nuevo arquetipo.
- Convertirse en revisor o traductor.

Los revisores forman parte del equipo de desarrollo de arquetipos. Para los arquetipos que tienen asignados pueden comprobar sus modificaciones, actualizar el arquetipo a una nueva versión revisada, participar en todo el proceso de revisión y comunicarse con el resto de miembros del equipo.

Los editores, además de toda la funcionalidad anterior, pueden:

- Crear nuevos arquetipos en el repositorio.
- La creación de nuevos arquetipos requiere añadirles metadatos para que pueden ser consultados y recuperados a partir de estos.
- Asignar arquetipos a proyectos concretos.
- Gestionan los proyectos, los detalles y miembros del mismos.
- Invitar usuarios y revisores al proyecto.
- Los arquetipos pasan por varios ciclos de revisión. Cuando uno ha terminado, el editor proporciona retroalimentación a los revisores con respecto a los cambios realizados.
- Obtener una vista general del estado de proyectos, arquetipos y procesos de revisión.
- Publicar los arquetipos que están listos.

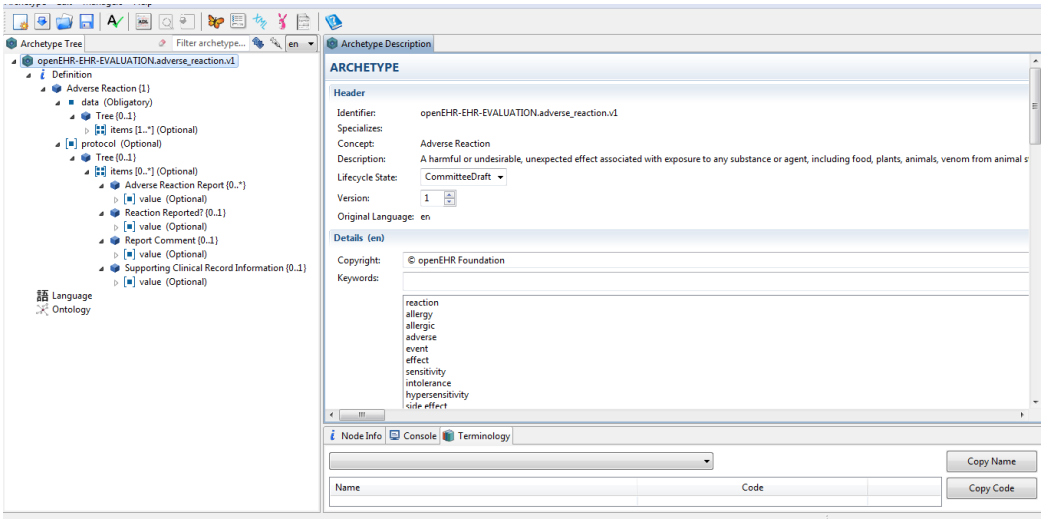

Figura 2.15: Interfaz de LinkEHR Archetype Editor

Los administradores, además de toda la funcionalidad anterior, pueden crear y gestionar conjuntos estables de arquetipos, crear nuevos proyectos e incubadoras y asignar editores a los mismos, manejar los usuarios.

2.1.4.2 LinkEHR Normalization Platform

LinkEHR [43] es una plataforma formada por distintos módulos, que persigue la interoperabilidad semántica de los sistemas de información clínica, creando una capa adicional que permite la edición de arquetipos, la normalización de datos usando arquetipos y la visualización de instancias de datos estructuradas según los arquetipos (extractos HCE). LinkEHR está formada por cinco módulos distintos:

LinkEHR Archetype Editor (ver figura 2.15), editor de arquetipos, facilita la definición de arquetipos. Trabaja con independencia de un estándar o modelo de referencia particular y permite enlazar los arquetipos con terminologías médicas. Trabaja con estándares CEN/ISO 13606, openEHR, HL7 CDA, CCR y CDISC ODM.

LinkEHR Studio, herramienta de normalización de datos que genera extractos de HCE según un arquetipo a partir de datos existentes. Es decir permite la transformación de datos no normalizados a datos arquetipados a través de la definición de correspondencias entre el esquema origen de los datos y el esquema destino. Estas correspondencias se traducen a consultas sobre las instancias de datos de entrada, que son transformadas a instancias

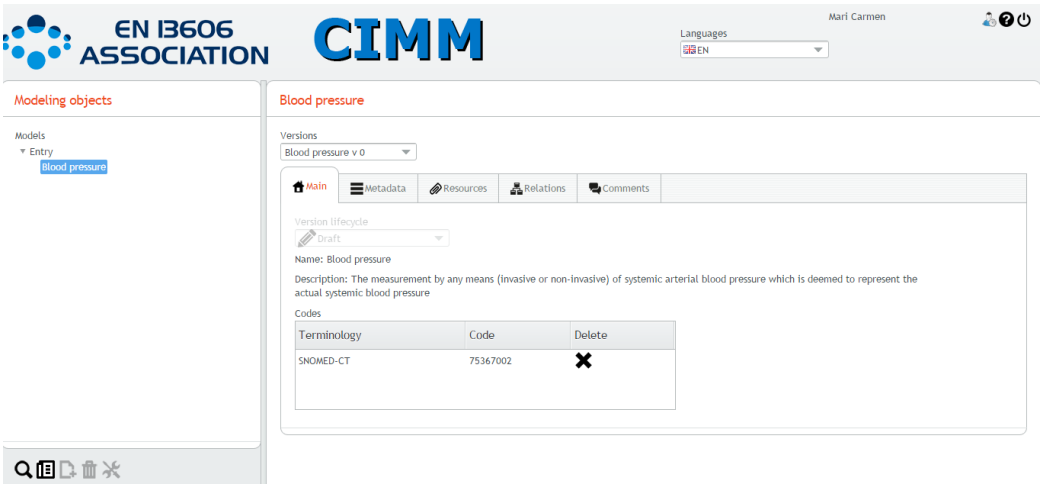

Figura 2.16: Interfaz de CIMM

según el esquema de salida.

LinkEHR Integration Engine, permite acceder a múltiples fuentes de datos heterogéneas y distribuidas, realizar una consulta sobre ellas e integrar los resultados en un documento XML.

LinkEHR Extract Server, servidor que se coloca en cada fuente de datos, recibe las peticiones de información, extracción de datos, y normalización mediante el uso de scripts XQuery previamente generados con LinkEHR Editor.

LinkEHR Viewer, visor de HCE basado en web para información clínica estandarizada.

2.1.4.3 Clinical Information Model Manager

La herramienta Clinical Information Model Manager (CIMM) [44] (ver figura 2.16), publicada por la asociación EN ISO 13606 [12], trata de convertirse en una plataforma de referencia para la publicación y gestión de arquetipos EN ISO 13606. La funcionalidad disponible se basa en los privilegios de usuario. Los nuevos registros tiene acceso de solo lectura, mientras que los usuarios avanzados tienen habilidad de editar y subir nuevos recursos a los modelos clínicos disponibles.

CIMM crea una entrada por cada modelo, donde se agrupan todos los recursos relacionados con el mismo, y permite la edición de los arquetipos utilizando LinkEHR Archetype Editor.

2.2 Repositorios bioinformáticos

La bioinformática es el estudio de cómo la información es representada y analizada en sistemas biológicos, especialmente información obtenida a nivel de moléculas. Mientras que la informática clínica se ocupa de la gestión de la información relacionada con el cuidado de la salud, la bioinformática se enfoca a la gestión de la información relacionada con las ciencias de la biología básica subyacentes, sin embargo, la colaboración entre las dos disciplinas es importante para la medicina traslacional [45].

El origen de la bioinformática se puede establecer a principios de los 60, donde tres factores contribuyeron al comienzo de la biología computacional [46]: (1) la secuenciación de una proteína completa, insulina, por parte de Frederick Sanger y su equipo [47] en la década que va de 1945 a 1955, confirmó que cada proteína estaba caracterizada por una estructura primaria única y llevó a la aparición, en los siguientes años, de nuevos métodos de secuenciación más eficientes. Esto provocó, a principios de los 60, la aparición de una colección de secuencias de aminoácidos utilizada como fuentes de datos en nuevas investigaciones. Esta colección iba en aumento, haciéndose cada vez menos manejable, y surgieron nuevos problemas en su uso que eran imposibles de resolver sin los ordenadores. (2) Con la demostración de la existencia de una estructura primaria en las proteínas y la aparición de métodos de secuenciación, surgió la idea de que macromoléculas como las proteínas transportan información. Esta idea, que relacionaba información con moléculas, creó un vínculo a nivel conceptual entre la biología molecular y las ciencias de la computación. (3) La disponibilidad de ordenadores de gran velocidad, surgidos durante la Segunda Guerra Mundial, junto a la aparición de lenguajes de programación de alto nivel, permitieron a los científicos realizar aprovechar el potencial del ordenador y las nuevas aplicaciones para realizar tareas imposibles para un humano, como el análisis filogenético de las secuencias de aminoácidos. Sin embargo, es a partir de la década de los 90 cuando la bioinformática comenzó su etapa de expansión. En esta época, las técnicas de secuenciación de ADN comenzaron a aplicarse a grandes proyectos sobre genomas (como el Proyecto del Genoma Humano [48]), lo que se unió a la aparición de supercomputadores y la generalización de Internet. En los siguientes años, las investigaciones y proyectos surgidos en torno a los genomas generaron grandes volúmenes de datos, las base de datos sobre biología molecular comenzaron a crecer exponencialmente y los métodos de la bioinformática se hicieron indispensables para manejar, analizar y explotar todos esos datos. La publicación de datos experimentales en el ámbito de la genómica se vio fomentada por los *Principios de las Bermudas* [49],

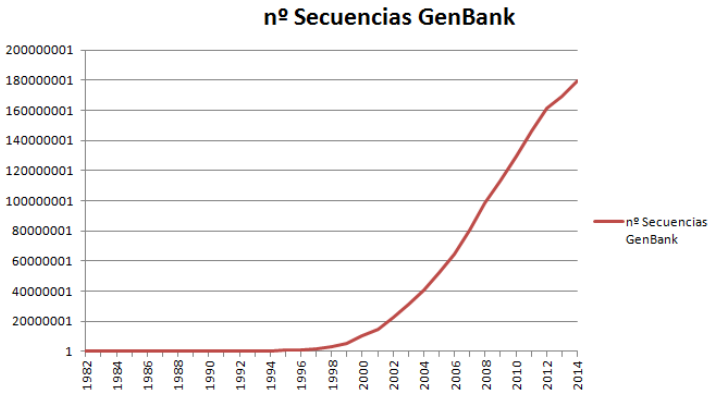

Figura 2.17: Estadísticas del número de secuencias en GenBank de 1982 a 2014

un acuerdo de los líderes del Proyecto del Genoma Humano para poner a disposición de la comunidad científica toda la información sobre secuencias genómicas humanas, de forma pública, 24 horas después de su generación. Hasta ese momento, la norma imperante era no poner los datos a libre disposición de la comunidad hasta su aparición en una publicación científica. Con esto se trató de favorecer el progreso de las investigaciones sobre las secuencias de datos y llevó a un aumento considerable de las bases de datos genómicas y la aparición de otras nuevas. La figura 2.17 muestra el aumento en el número de secuencias almacenadas en GenBank, una base de datos de secuencias genéticas del NIH y de todas las colecciones anotadas de secuencias de ADN disponibles públicamente [50]. Como vemos, a partir de 1998 creció el aumento anual en numero de secuencias.

En general, la información que se maneja en bioinformática se caracteriza por [51]:

- Gran cantidad de datos: como ya se ha comentado, la cantidad de proyectos de secuenciación hacen que los datos disponibles sigan aumentando.

- Complejidad: una entidad biológica contiene muchas relaciones con otras entidades, por ejemplo, una proteína tiene asociada una secuencia, función, procesos en los que está involucrada, enfermedades, etc.

- Volatilidad: el conocimiento que se tiene sobre las entidades biológica cambia y se va incrementando debido a que se continua investigando sobre ellas.

- Distribución: el hecho de que haya una gran cantidad de recursos disponibles hace que la información de una misma entidad esté repartida en distintas bases de datos.
- Heterogeneidad: además de que la información está distribuida, la mayoría de las bases de datos no utilizan una nomenclatura uniforme para las entidades biológicas, por lo que el mismo recursos puede estar identificado de distinta forma.

2.2.1 Tipos de bases de datos

Podemos clasificar las bases de datos biológicas atendiendo al ámbito del dominio de la información y al grado de curación de los datos. Al clasificarlas según el ámbito de su dominio podemos distinguir:

- Bases de datos de nucleótidos, como por ejemplo GenBank [50], una colección anotada de todas las secuencias de ADN disponibles públicamente.
- Bases de datos de proteínas, como UniProt [52], un repositorio con información funcional de proteínas y anotaciones.
- Bases de datos de estructuras de proteínas, como PDB [53], un repositorio de estructuras biológicas macromoleculares con información 3D de las mismas.
- Bases de datos de genomas y mapas, como Genome [54], que agrupa información de genomas, incluyendo secuencias y mapas.
- También podemos incluir en esta categoría las bases de datos especializadas en un organismo concreto, es decir, pueden incluir información del ámbito de las categorías anteriores, pero especifica de una organismo, como por ejemplo Mouse Genome Database (MGD) [55], que integra todo tipo de información biológica sobre el ratón.

Atendiendo al grado de curación, las bases de datos pueden ser curadas o no curadas. Las bases de datos no curadas contienen datos no revisados, que pueden ser redundantes e incompletos. Cuando los datos no curados se completan y organizan, los datos referentes a una misma entidad se agrupan en el mismo registro y expertos los revisan y anotan con información adicional, estamos hablando de bases de datos curadas. UniProt se divide en dos bases de datos principales, Swiss-Prot, que contiene los datos curados y TrEMBL que contiene datos no curados que simplemente están anotados automáticamente pero no revisados.

2.2.2 Almacenamiento y representación de bases de datos biológicas

Los resultados de investigación en biología están disponibles en diversos formatos. Es común encontrarlos en lenguaje natural como parte de publicaciones científicas. MEDLINE es un ejemplo de base de datos bibliográfica de la Librería Nacional de Medicina Americana (National Library of Medicine, NLM) [56] que contiene alrededor de 21 millones de referencias a artículos científicos sobre las ciencias en la vida, con enfoque en la biomedicina. El acceso a esta base de datos se hace a través de PubMed [57], que enlaza la bibliografía biomédica procedente de MEDLINE, junto a otras revistas científicas y libros en línea. Cuando los resultados se encuentran en bases de datos estructuradas, los formatos más comunes son ficheros de texto plano, ficheros estructurados, bases de datos relacionales y bases de datos basadas en grafos.

2.2.2.1 Ficheros de texto plano

En este tipo de representación los datos se almacenan en ficheros de texto por medio de pares atributo-valor y separados entre sí por caracteres especiales. Ejemplos de formatos de representación basados en texto son el formato FASTA [58] para la representación de secuencias de nucleótidos o aminoácidos. En este formato las secuencias se representan mediante varias líneas de caracteres, donde cada carácter representa un nucleótido o aminoácido y una línea cabecera que identifica la secuencia. El formato FASTQ [59] es otro formato basado en texto para representar secuencias resultado de métodos de secuenciación de nueva generación (Next-Generation Sequencing NGS) y sus puntuaciones de calidad, la secuencia se representa como en el formato FASTA, a la que se le añade una línea adicional con las puntuaciones de calidad, un símbolo representando la puntuación de cada una de las letras que representa un nucleótido. Otro formato muy común es el General Feature Format (GFF) [60], un formato estandarizado para representar anotaciones de características sobre una secuencia. En este formato cada anotación se define por una serie de campos, identificados en la cabecera y cada anotación se almacena en una línea del fichero donde el valor de cada campo está separado por tabulaciones. La base de datos GenBank [50] organiza sus entradas como ficheros de texto plano.

2.2.2.2 Ficheros estructurados

En este tipo de representación los datos se estructuran mediante etiquetas. El formato más común es el XML. Muchos proyectos han propuesto estandarizaciones de datos biológicos basadas en XML como medio de facilitar la integración de datos. OrthoXML y SeqXML [61] son dos ejemplos de representación basada en XML de un proyecto del Stockholm Bioinformatics Centre [62] que trata de organizar y estandarizar conjuntos de datos de proteomas y representar las relaciones de ortólogos resultado del estudio de dichos proteomas. Dos genes se definen como ortólogos cuando perteneciendo a diferentes especies ambos derivan de un mismo gen en su último ancestro común. Este tipo de relación es interesante de identificar porque es probable que los genes ortólogos tengan la misma función. Algunas bases de datos que se representan usando OrthoXML o que tienen disponible sus datos también en este formato son InParanoid [63], PANTHER [64], PhylomeDB [65], MGD Mouse Genome Database [55]. Otros proyectos son ProML [66], para la especificación de secuencias, familias y estructuras de proteínas, BIOML [67] para la estandarización de anotaciones de secuencias de biopolimeros, Chado-XML [68] es una correspondencia a XML directa del esquema de base de datos relacional Chado [69], diseñado para representar datos biológicos como secuencias, genotipos o filogenias.

2.2.2.3 Bases de datos relacionales

En este tipo de almacenamiento los datos se estructuran siguiendo el modelo relacional propuesto por Codd [70]. Los principales elementos de este modelo son las relaciones, que se representan como tablas y modelan los conceptos del dominio. Los atributos de la relación representan las columnas de la tabla. Cada relación está formada por un conjunto de tuplas, donde cada tupla tiene un conjunto de valores para los atributos de la relación. El modelo relacional tiene la ventaja de proporcionar un lenguaje de consulta, SQL (Structured Query Language) que facilita el acceso a los datos almacenados en la base de datos.

La base de datos FlyBase [71] proporciona acceso a una representación Chado de sus datos. La ontología Gene Ontology (GO) [21], una ontología de funciones moleculares, procesos biológicos y componentes celulares, tiene una representación en formato relacional, GO Database [72].

2.2.2.4 Bases de datos basadas en grafos

Las bases de datos basadas en grafos se presentan como alternativa a las bases de datos relacionales. Formalmente, un grafo es una colección de vértices y aristas, a los vértices se les llama nodos, y las aristas representan las relaciones que los conectan. En un grafo, se representan las entidades de un dominio como nodos, y las relaciones entre entidades como conexiones entre los nodos.

En estas bases de datos el modelo de datos es un grafo, en el cual las relaciones cobran mucha importancia. Por ello, presentan ventajas respecto a las bases de datos relacionales a la hora de representar dominios en los que los datos están muy relacionados entre sí y que requieren frecuentes cambios de esquema. Por un lado, la consulta de datos muy relacionados entre sí requiere la unión de varias tablas en las bases de datos relacionales, lo que deteriora el rendimiento de las consultas con el incremento de datos. En cambio, las bases de datos basadas en grafos presentan un buen rendimiento en las consultas que además tiende a permanecer constante [73]. Por otro lado, los grafos admiten de forma natural la adición de nuevas relaciones, nodos o nuevos subgrafos a su estructura, sin perturbar las consultas o la funcionalidad existente.

Estas características hacen a las bases de datos basadas en grafos especialmente adecuadas para representar información de dominios donde las relaciones entre los datos o su topología tienen la misma importancia, o más, que los datos en sí. El dominio de la bioinformática se caracteriza por información muy relacionada entre sí, dónde además la información es muy relevante, como es el caso de las relaciones entre proteínas, las redes metabólicas o los mapas genéticos. Un ejemplo de la utilidad del uso de grafos lo podemos ver en el ámbito del estudio de la estructura de proteínas. La estructura tridimensional de las proteínas es clave para entender su función y evolución. Analizar estructuras tridimensionales plegadas que son estables proporciona conocimiento en la estabilidad de plegamiento y sobre su función y ayuda a la predicción de estructura de proteínas a partir de secuencias de aminoácidos. La representación de estructuras de proteínas por medio de grafos ha demostrado su utilidad en el análisis de éstas [74].

Un ejemplo de base de datos biológica basada en grafos lo encontramos en Bio4j [75], una base de datos bioinformática que incluye la mayoría de los datos disponibles en UniProt KB [76], Gene Ontology [21], UniRef [77], RefSeq [78], NCBI Taxonomy [79] y ExPASy Enzyme DB [80].

2.3 Terminologías biomédicas

Las terminologías clínicas se consideran fundamentales para garantizar la interoperabilidad de la información clínica [10], mientras que si nos centramos en el dominio estrictamente biológico, los vocabularios controlados son artefactos imprescindibles en el proceso de anotación de secuencias genómicas. Sin embargo, la definición de terminología y otros términos utilizados en su mismo ámbito no está clara, en muchos casos diferentes autores suelen utilizar la misma palabra de forma diferente [81]. A continuación doy la definición de una serie de conceptos cuyo uso se suele intercambiar y confundir entre sí:

- Vocabulario controlado: lista de elementos usados con un propósito concreto, como el de reducir la ambigüedad en los sistemas de información o el de evitar errores ortográficos.
- Sistema de identificadores (códigos): lista de códigos utilizados para referenciar de forma inequívoca las entidades. Se suelen utilizar en léxicos, ontologías y tesauros.
- Léxico: listado de unidades lingüísticas en un lenguaje específico que pueden ir enlazadas a un vocabulario controlado o una ontología. Normalmente incluye información lingüística como sinónimos, términos preferentes, categoría gramatical, etc.
- Ontología: en el ámbito de los sistemas de información se define como un modelo formal y explícito de los conceptos de un domino concreto, con sus relaciones y restricciones.
- Clasificación: organización de entidades en clases para un propósito concreto.
- Tesauro: sistema de términos organizado para navegar entre los términos siguiendo las relaciones "término más general" y "termino más restringido".

Dadas todas estas definiciones, podemos definir una terminología como cualquiera de los conceptos anteriores en distintas combinaciones. Cuando las terminologías incluyen identificadores o códigos se conocen como sistemas de codificación.

En el ámbito biomédico, el uso de terminologías intenta superar la heterogeneidad inherente a la información generada en dichos dominios, proporcionando un vocabulario estandarizado que sirve para codificar los conceptos

almacenados en HCE y bases de datos biológicas. El recurso más significativo creado con este propósito es el Sistema de Lenguaje Médico Unificado (UMLS) [82] que integra y distribuye terminologías en los ámbitos biológicos y clínicos para promover la creación de sistemas de información biomédica más eficiente e interoperables.

2.3.1 Terminologías clínicas

Una terminología clínica se define como los términos estandarizados y sus sinónimos que registran hallazgos, circunstancias, eventos e intervenciones relacionados con pacientes, con el detalle suficiente para dar soporte a la atención clínica, ayuda a la decisión, resultados de investigación y mejora de la calidad, y pueden ser alineados de forma eficiente con clasificaciones más amplias para propósitos administrativos, de regulación, de supervisión y de requisitos fiscales [83]. El uso de estas terminologías permite la representación consistente de la información clínica dentro de las HCE, permite comparar los datos de los pacientes, mejorando la efectividad y eficiencia de la atención sanitaria [83]. Por lo tanto, el uso conjunto de terminologías junto a estándares de HCE mejoran la recuperación y reutilización de la información clínica, siendo un paso más hacia la consecución de la interoperabilidad semántica de la misma.

El uso de terminologías en las HCE lo encontramos en los modelos clínicos. Lenguajes como ADL permiten anotar los conceptos clínicos en los arquetipos con términos locales, estos términos, representados con *códigos AT* son los que aparecen en la sección ontología del arquetipo, mencionada en la sección 2.1.3. Opcionalmente, los elementos del arquetipo se pueden enlazar a términos de una terminología externa, codificando los términos del arquetipo y facilitando la recuperación de la información. A estos enlaces a terminologías externas en los arquetipos se les conoce como *term-bindings*. El número de terminologías clínicas que pueden ser utilizadas para codificar conceptos en las HCE es muy amplio [31], lo que puede ser un inconveniente a la hora de extraer y comparar los datos contenidos en el historial clínico.

En el dominio clínico nos encontramos con gran variedad de terminologías clínicas con distintos propósitos. La clasificación internacional de enfermedades (ICD) [84] es una de las terminologías más antiguas. Promovida por la Organización Mundial de la Salud (OMS), es utilizada internacionalmente para codificar diagnósticos, en estudios epidemiológicos y para la monitorización de la incidencia y prevalencia de enfermedades en la población. LOINC (Logical Observation Identifiers Names and Codes) [85] es otra terminología que proporciona un conjunto de nombres de códigos universales para

identificar resultados de laboratorio y pruebas clínicas. SNOMED CT (Systematized Nomenclature of Medicine - Clinical Terms) [86] es la terminología clínica multilingüe, codificada, de mayor amplitud, precisión e importancia en el mundo, utilizada para codificar datos clínicos. Se trata de un estándar internacional y actualmente es distribuida por la International Health Terminology Standards Development Organisation (IHTSDO). Esta terminología se ha desarrollado de forma colaborativa para conseguir que sea lo suficientemente diversa para poder ser utilizada por cualquier profesional médico alrededor del mundo. Surgió como la unión de dos terminologías de referencia en la atención sanitaria, SNOMED RT (Systematized Nomenclature of Medicine Reference Terminology) desarrollada por el Colegio Americano de Patólogos y Clinical Terms v3, desarrollada por el Servicio Nacional de Salud de Reino Unido (NHS).

SNOMED CT fue diseñada para permitir una representación efectiva de la información en las HCE, está reconocida como la terminología líder para ser usada en las HCE. Entre sus características a destacar:

- Sus términos cubren la mayoría de conceptos clínicos usados en el historial de los pacientes.
- Capacidad para expresar distintos niveles de detalle clínico en las entradas del historial del paciente usando expresiones que contienen uno o más identificadores de concepto.
- Contiene relaciones entre conceptos que permiten recuperar de forma consistente información clínica relacionada entre sí.
- Soporta la representación de variantes del lenguajes y dialectos, conjunto de valores, jerarquías alternativas y mapeos a clasificaciones.

La terminología está formada por conceptos, descripciones y relaciones organizados de forma jerárquica que permiten representar cualquier tipo de información clínica a cualquier nivel de detalle y permiten una recuperación y reutilización eficiente de la información basándose en el significado de la misma.

2.4 El objetivo de la interoperabilidad semántica

Las propuestas de estándares, especificaciones, terminologías y formalismos buscar ofrecer una representación de la información biomédica que permita

su correcto almacenamiento, recuperación, explotación y compartición. Uno de los objetivos principales es el de la interoperabilidad semántica, ser capaz de intercambiar e interpretar la información independientemente del sistema de origen. Sin embargo, la variedad de propuestas de representación y terminologías disponibles dificultan la consecución de este objetivo, pues distintos sistemas e instituciones utilizan distintos modelos de representación y terminologías.

Los arquetipos han sido considerados prometedores para las tareas de intercambio de datos clínicos de forma formal y escalable [9], sin embargo, existen pocas herramientas dedicadas a la explotación de arquetipos y la representación más común de estos utiliza el lenguaje ADL, el cual tiene una orientación sintáctica. Los lenguajes con orientación sintáctica dificultan la realización de actividades semánticas requeridas para obtener la interoperabilidad semántica [87], por ejemplo, actividades como comprobar la corrección semántica de la definición de un arquetipo, es decir, que las restricciones de un arquetipo sean compatibles respecto a su arquetipo padre, o comprobar que dos arquetipos son equivalentes, requieren mucho más trabajo en un representación orientada a sintaxis. ADL permite el enlace a terminologías externas, pero la explotación conjunta del contenido del arquetipo y de las terminologías se complica debido a la propia naturaleza del lenguaje.

El uso de XML, muy extendido para los recursos biológicos, tiene asociados problemas similares a los de ADL. Su uso está pensado para mejorar el intercambio e integración de los datos, estandarizando la definición de algún dominio biológico. Sin embargo, el lenguaje puede ser utilizado en más de una forma para definir la misma información, haciendo incompatibles representación que a priori definen la misma información. En [88] se muestran que XML es un lenguaje poco flexible para el dominio biomédico, donde el conocimiento que se tiene de los datos es muy volátil, ya que está sujeto a cambios debido al desarrollo de nuevos experimentos que aportan nueva información.

2.4.1 Iniciativas de interoperabilidad semántica

Varias iniciativas internacionales tratan de proporcionar soluciones y promover directrices que permitan la consecución de la interoperabilidad semántica.

2.4.1.1 Clinical Information Modeling Initiative (CIMI)

La iniciativa internacional CIMI (Clinical Information Modeling Initiative) tiene el objetivo de promover un formato común para la interoperabilidad de los modelos de información clínica [89]. La motivación de CIMI está fundada en que los modelos clínicos cada vez tienen un papel más importante en los

sistemas de información sanitarios, especialmente controlando la entrada y visualización de la información, por lo que su reutilización ha despertado el interés de organismos de estandarización e instituciones en el ámbito de la informática médica, que son los que integran CIMI.

El objetivo principal de CIMI es crear un repositorio compartido de licencia abierta de modelos de información clínica detallados. Estos modelos utilizarán formalismos aprobados, un conjunto común de tipos de datos base y contendrán enlaces explícitos a terminologías estándar. En CIMI se define un modelo lógico como el modelo independiente de lenguajes de programación o tipos de bases de datos específicos, que muestra las relaciones estructurales entre los elementos de modelado y cuyos elementos codificados tienen enlaces explícitos con valores codificados permitidos. Un modelo lógico debe dar soporte a las consultas para recuperar instancias de datos.

CIMI propone utilizar ADL 1.5 como formalismo de punto de partida junto al Archetype Object Model (AOM) de ISO EN 13606 parte 2, además de desarrollar un perfil UML junto a estereotipos UML y especificaciones y transformaciones XMI, utilizando UML 2.0 y OCL como lenguaje de restricciones, con la intención de que los usuarios de los modelos puedan convertirlos a formatos locales. La estrategia de implementación de CIMI es realizar (bajo demanda) mapeos entre sus modelos lógicos e implementaciones particulares, como pueden ser recursos y perfiles FHIR, HL7 v3, etc. CIMI ha seleccionado SNOMED-CT y LOINC como terminologías clínicas de referencia, siendo SNOMED-CT la principal.

2.4.1.2 SemanticHEALTH

El proyecto europeo SemanticHEALTH estableció en su informe final una hoja de ruta para lograr la interoperabilidad semántica de la HCE donde se destaca el papel esencial del estándar ISO EN 13606 y la arquitectura de modelo dual [10]. El informe final define varios niveles de interoperabilidad entre sistemas de HCE:

- Nivel 0: ausencia de interoperabilidad.
- Nivel 1: nivel más básico, se consigue cuando la comunicación de la HCE se realiza a nivel de documentos que pueden ser accedidos a través de Internet. A este nivel se le denomina de interoperabilidad sintáctica y técnica y para cumplirlo es suficiente con que los documentos sean legibles.
- Nivel 2: se denomina de interoperabilidad semántica parcial. En este nivel existe una comunicación de la HCE de forma estructurada, y aun-

que mucho texto está presente como texto libre, información relevante está codificada por un sistema de codificación internacional. Sin embargo, los sistemas participantes deben realizar un esfuerzo de adaptación de la información a sus repositorios locales.

- Nivel 3: se denomina de interoperabilidad semántica completa. Cualquier extracto de HCE pueda ser importado y combinado con datos locales de un sistema sin necesidad de especificar correspondencias previas con los sistemas locales.

Para alcanzar la interoperabilidad semántica completa el informe final de SemanticHealth recomienda el uso de la arquitectura de modelo dual (ISO EN 13606), compartir una biblioteca de estructuras de datos clínicas (arquetipos) y el uso terminologías clínicas de forma consistente, preferiblemente la terminología SNOMED-CT.

2.4.1.3 SemanticHealthNet

La red europea SemanticHealthNet (SHN) [90] sigue las recomendaciones de su proyecto predecesor, SemanticHealth, en la búsqueda de integrar más estrechamente los modelos de información, tal como se utilizan en la HCE, así como terminologías y ontologías, para mejorar la interoperabilidad semántica. Tiene como objetivo desarrollar un proceso organizativo y de gobierno paneuropeo, escalable y sostenible, para la interoperabilidad semántica del conocimiento clínico, que permita optimizar el uso de sistemas de HCE a través de distintas instituciones para el cuidado de los pacientes y la investigación clínica y sanitaria.

SHN acepta la coexistencia de diferentes estándares para la HCE y proponen soluciones complementarias para alcanzar al interoperabilidad semántica. Afirma que las ontologías deberían jugar un papel fundamental en la consecución de la interoperabilidad semántica y propone una infraestructura semántica basada en un framework ontológico [91] junto a un conjunto de patrones ontológicos de contenido [92] que utilizan este framework como referencia. El framework consiste en tres tipos de ontologías: una ontología de alto nivel, una ontología de entidades de información y una ontología del dominio clínico. Una ampliación de este framework se verá en el siguiente capítulo.

Capítulo 3

Web Semántica y la información Biomédica

3.1 Web Semántica y sus tecnologías

La Web Semántica [16] es una evolución de la Web tradicional en la que se da significado bien definido a la información de manera que ésta pasa de estar diseñada para ser utilizada por humanos a que también pueda ser manipulada por ordenadores. Para que esto sea posible, los ordenadores deben tener acceso a información estructurada junto a conjuntos de reglas de inferencia que les permitan realizar razonamiento automático.

En la Web Semántica se aporta significado semántico a los datos para que la información sea fácil de utilizar y permita a aplicaciones avanzadas explotar ese significado y así, las acciones de búsqueda, intercambio, integración y manejo de la información sean más efectivas.

Tim Berners-Lee propuso una primera arquitectura por capas para implementar la Web Semántica (véase figura 3.1).

Las capas Unicode y URI (Uniform Resource Identifier) [93] aseguran la identificación inequívoca de la información y su correcta representación en cualquier idioma. Para ello propone el uso de URI como identificador y de Unicode como estándar de codificación de caracteres.

El lenguaje XML [94] junto a los espacios de nombres (NameSpaces) y el XML Schema [95], ampliamente usados en la Web tradicional permiten crear etiquetas y añadir estructura en documentos y páginas web, aunque no indican nada sobre el significado de esa estructura. Esta capa incluida en la arquitectura de la Web Semántica asegura que se pueda integrar la definición de Web Semántica con los demás estándares basados en XML.

El lenguaje RDF (Resource Description Framework) [96] permite descri-

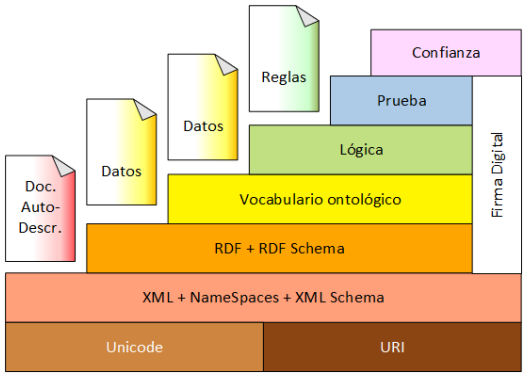

Figura 3.1: Arquitectura de la Web Semántica

bir los recursos de la Web Semántica. Un recurso se define como cualquier cosa que puede tener datos asociados, ya sea del mundo real o una entidad abstracta. En RDF el significado se expresa en tripletas, donde cada tripleta se compone de sujeto, predicado y objeto. Las sentencias definidas en RDF mediante tripletas expresan una relación entre dos objetos, siendo estos dos objetos el sujeto y el objeto. El predicado representa la naturaleza de la relación entre ambos. Los tres componentes de la tripleta se identifican inequívocamente con una URI. Un modelo RDF se define como un grafo dirigido etiquetado formado por el conjunto de todas sus tripletas. El sujeto y objeto son nodos del grafo mientras que el predicado representa el arco que conecta ambos nodos. RDF Schema (RDFS) [97] es una extensión del vocabulario de RDF que permite definir los recursos como clases, organizar estas clases en jerarquías, definir las relaciones entre clases como propiedades y definir sus dominios y rangos.

A pesar de la extensión RDFS para RDF, este lenguaje sigue siendo insuficiente, por ello la siguiente capa, vocabulario ontológico, mejora la funcionalidad y expresividad de la capa anterior. Proporciona nuevos conceptos, relaciones y propiedades.

La capa lógica está basada en la posibilidad de definir reglas de inferencia que son usadas por los razonadores como método para inferir nuevo conocimiento sobre el ya disponible.

Las últimas capas, prueba y confianza, apenas han sido desarrolladas o estandarizadas. En general estos últimos niveles establecen una capa de seguridad que evalúa los recursos y junto a la firma digital garantiza la fiabilidad de los mismos.

3.1.1 Resource Description Framework (RDF)

RDF define un modelo estándar para el intercambio de recursos en Internet. Con recursos nos referimos a cualquier cosa, documentos, personas, objetos físicos, conceptos abstractos, etc. RDF permite expresar la información sobre los recursos de manera que estos pueden ser compartidos entre distintas aplicaciones sin que haya pérdida de información.

En RDF los recursos se describen por medio de sentencias simples en forma de tripletas. Cada tripleta está formada por un sujeto, un predicado y un objeto. El sujeto de una tripleta es una URI que identifica a un recurso. El objeto puede ser o bien un valor literal, como una cadena de caracteres, un número o fecha; o bien la URI de otro recurso que está relacionado con el sujeto. El predicado indica qué tipo de relación existe entre el sujeto y el objeto, y también se identifica con una URI. Cuando el objeto es un valor literal, la tripleta está describiendo una propiedad del recurso sujeto. Si el objeto es otro recurso, la tripleta expresa la relación entre dos recursos.

Así pues, RDF utiliza URI para identificar recursos y sus relaciones y literales para identificar valores básicos que no son recursos. Mientras que las URI se utilizan en todas las posiciones de la tripleta (sujeto, predicado, objeto), los literales sólo pueden ser utilizados como objeto. Un literal puede tener opcionalmente un etiqueta de lenguaje y normalmente se asocian con un tipo de datos (una URI de tipo de datos). Los tipos de datos utilizados se definen en la especificación de tipos de datos de XML Schema.

Hay ciertos casos en los que puede ser útil referirse a un recurso sin necesidad de utilizar un identificador global. Para estos casos, RDF proporciona nodos en blanco (blank nodes), que solo tienen un identificador local, y no pueden ser identificados más allá del grafo en el que se encuentran.

3.1.2 Ontologías

La tecnología semántica base para la Web Semántica es la ontología, que representa una visión común, compartible y reutilizable del conocimiento de un dominio de aplicación [98]. Este concepto se ha tomado prestado de la filosofía, donde ontología es una teoría sobre la naturaleza de la existencia. En ciencias de la información se han dado varias definiciones formales de ontología, siendo la más famosa la de Gruber, que define una ontología como la representación de una vista común, compartible y reutilizable de un dominio de aplicación [18] o la de Guarino, que define una ontología como un conjunto de axiomas lógicos que explican el significado de un vocabulario formal [99]. En general, en las ciencias de la información podemos ver una ontología como un modelo que representa formalmente estructuras de cono-

cimiento, que incluye conceptos, sus propiedades, sus atributos, las relaciones con otros conceptos y los axiomas relacionados con estos.

En la Web Semántica existen varios lenguajes para representar ontologías, siendo RDFS y Web Ontology Language (OWL) [100] los más utilizados. RDFS, como se ha dicho anteriormente, presenta una extensión de RDF y proporciona un lenguaje mínimo, con clases, propiedades, jerarquías y restricciones de dominio y rango. OWL proporciona un lenguaje más complejo que veremos en detalle en la siguiente sección.

3.1.2.1 RDF Schema (RDFS)

RDFS es una extensión de RDF que permite expresar ontologías simples utilizando la sintaxis RDF. RDFS define los recursos rdfs:Class, rdfs:Resource y rdf:Property para definir clases (conceptos), recursos y propiedades, respectivamente.

RDFS soporta jerarquías de clases y propiedades, y restricciones de dominio y rango para las propiedades. Para ello aporta el siguiente conjunto de meta-propiedades:

- rdf:type, para definir la relación de "instancia de".
- rdfs:subClassOf, para modelar las jerarquías de clase.
- rdfs:subPropertyOf, para modelar las jerarquías de propiedades.
- rdfs:domain, para restringir las instancias sujeto de una propiedad como instancias de una clase concreta.
- rdfs:range, para restringir las instancias objeto de una propiedad como instancias de una clase concreta.

3.1.2.2 Web Ontology Language

OWL ha sido diseñado para que las aplicaciones puedan procesar y explotar el contenido de la información. Este lenguaje facilita una mayor interpretación del contenido web por parte del ordenador del que proporcionan XML, RDF y RDFS, pues proporciona vocabulario adicional para describir propiedades y clases junto con una semántica formal. Al igual que en RDFS, los elementos básicos de OWL son clases, propiedades e individuos, que son instancias de las clases. Sus propiedades son relaciones binarias que se distinguen entre owl:ObjectProperty y owl:DatatypeProperty. Las owl:ObjectProperty relacionan dos individuos mientras que las owl:DatatypeProperty relacionan

un individuo con un valor literal. También define jerarquías, restricciones de dominio y rango, así como restricciones de cuantificadores universales y existenciales entre otras. En OWL se agrupan varios sublenguajes que se diferencian en su nivel de expresividad y que fueron propuestos en la primera versión de OWL (ver figura 3.2 izquierda):

- OWL Lite: es el lenguaje menos expresivo. Fue diseñado para representar jerarquías de clasificación con restricciones simples. Por ejemplo, soporta restricciones de cardinalidad pero sólo con valores de 0 o 1. Respecto a RDFS, añade restricciones de rango local, restricciones existenciales, restricciones de cardinalidad simple y varios tipos de propiedades (inversa, transitiva y asimétrica).

- OWL Full: proporciona la máxima expresividad y la libertad sintáctica de RDF sin garantías computacionales. Incorpora los niveles Lite y DL, y permite mezclar libremente OWL y RDF. Por lo tanto, cualquier documento RDF es un documento válido en OWL Full.

- OWL DL: proporciona la mayor expresividad garantizando que todas las conclusiones son computables y finalizarán en un tiempo finito. Incluye todas los constructores del lenguaje OWL, pero sólo pueden ser usados bajo ciertas restricciones. Su nombre viene de su correspondencia con la lógica descriptiva [101], un campo de la investigación que ha estudiado las lógicas que forman la base formal de OWL.

  En la versión 2.0 de OWL [102] se introducen tres perfiles para OWL DL con distinta expresividad orientados a características prácticas para aplicaciones del mundo real (ver figura 3.2 derecha):

  - OWL 2 EL: perfil cercano a la lógica descriptiva EL++ [103] , asegura un tiempo polinomial para todas los problemas estándar de razonamiento. Es adecuado para aplicaciones que requieren ontologías muy largas y donde se puede sacrificar poder de expresividad con tal de obtener mejor rendimiento.
  - OWL 2 QL: destinado a aplicaciones que utilizan ontologías relativamente ligeras para organizar gran numero de individuos y donde es útil acceder a los datos directamente a través de consultas relacionales (SQL)). Permite responder a consultas de un modo formal y completo en un tiempo computacional razonable.
  - OWL 2 RL: permite la implementación de algoritmos con tiempo de razonamiento polinomial usando tecnologías de bases de datos basadas en reglas que operan directamente en tripletas RDF. Es

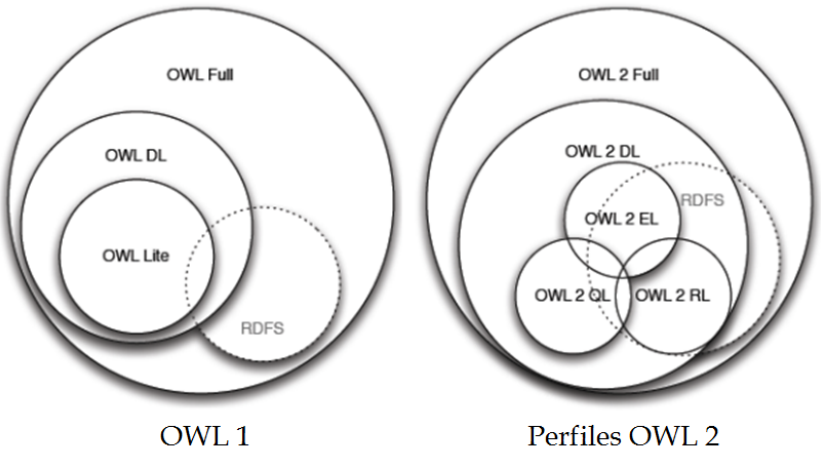

Figura 3.2: Relación entre los lenguajes y perfiles OWL

particularmente adecuado para aplicaciones con ontologías relativamente ligeras que organizan gran número de individuos y donde es útil manejar directamente datos en forma de tripletas RDF.

En este trabajo, cuando hablamos de OWL, nos referiremos a OWL DL que aporta una expresividad y un rendimiento razonable.

3.1.2.3 Razonamiento

En la Web Semántica se define inferencia como la capacidad de encontrar nuevas relaciones entre recursos a través de procedimientos automáticos de razonamiento basándose en los datos y en información adicional en forma de un conjunto de reglas [104].

Nos centraremos en las ontologías representadas con el lenguaje OWL DL que, como ya mencionamos, proporciona la máxima expresividad garantizando completitud computacional y por lo tanto garantiza que cualquier razonamiento se hará en un tiempo finito.

En la comunidad de la lógica descriptiva se considera que cualquier sistema de representación de conocimiento basado en lógica descriptiva es capaz de proporcionar un conjunto específico de razonamientos como mínimo [101]. OWL DL es una variante sintáctica de la lógica descriptiva $\{\mathcal{SHOIN}(\mathcal{D})\}$ [105] por lo que un razonador OWL debe proporcionar al menos el siguiente conjunto de servicios de inferencia de lógica descriptiva:

- Chequeo de consistencia: se asegura de que una ontología no contiene ninguna definición contradictoria.

- Satisfacibilidad de clase: comprueba si es posible que una clase tenga instancias. Si una clase no puede tener instancias, definir instancias para esa clase causa que la ontología sea inconsistente.
- Clasificación: computa las relaciones de subclase entre todas las clases para crear la jerarquía de clases completa.
- Realización: encuentra la clase más específica a la que un individuo pertenece. Realización sólo se puede realizar después de la clasificación ya que el tipo de un individuo se define respecto a la jerarquía de clases.

Existen varios axiomas en OWL-DL que son relevantes para el razonamiento. El axioma `subClassOf` se conoce como la condición necesaria. Una condición necesaria es una condición que debe cumplirse para que un individuo pertenezca a una clase, pero no es suficiente por sí sola. El axioma `equivalentTo` se conoce como condición necesaria y suficiente. Una condición necesaria y suficiente es una condición que si se cumple, basta para garantizar que un individuo pertenece a una clase. Otros axiomas de interés son aquellos que expresan identidad en las entidades en OWL, afirmando que una entidad dada es diferente a otras. Los axiomas `sameAs` y `differentFrom` son usados en individuos, mientras que los axiomas `disjointFrom` y `equivalentTo` se utilizan en clases.

Existen varios herramientas de razonamiento disponibles para OWL como son Pellet [106], FaCT++ [107] y Hermit [108].

3.1.3 Lenguaje de consulta

3.1.3.1 SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) [109] es un lenguaje de consulta para RDF que permite acceder a varios repositorios de datos RDF o repositorios que ofrecen vistas RDF de acceso para obtener tripletas de datos.

Una consulta SPARQL contiene un conjunto de patrones de tripletas que forman un patrón de grafo básico. Estos patrones son tripletas RDF en los que el sujeto, predicado y objeto puede ser una variable. La consulta devuelve un subgrafo del grafo RDF consultado que es equivalente al patrón de grafo básico con las variables reemplazadas por términos RDF del subgrafo de los datos. Por ejemplo, dados los datos RDF:

```
@prefix dc:   <http://purl.org/dc/elements/1.1/> .
@prefix :     <http://example.org/book/> .
```

```
:book1  dc:title  "SPARQL Tutorial" .
```

Una consulta SPARQL para recuperar el título de un libro consulta sería:

```
PREFIX  dc:<http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE
{
  <http://example.org/book/book1> dc:title ?title .
}
```

En la consulta, el objeto de la tripleta es una variable (`?title`) y devuelve como resultado ‘‘`SPARQL Tutorial`’’. Las variables se identifican con “?”, mientras que las URI se representan entre “<” y “>”.

SPARQL ha sido diseñado para un uso a escala de la Web, por lo que permite hacer consultas sobre orígenes de datos distribuidos, independientemente del formato.

3.1.4 Linked Data

Linked Data se refiere a un conjunto de buenas prácticas para publicar y enlazar entre sí datos estructurados en la web. Estas buenas prácticas fueron propuestas por Tim Berners-Lee [110] y se conocen como principios de Linked Data:

- Usar URI para identificar recursos.
- Usar HTTP URI (URI accesibles a través del protocolo HTTP), para que las personas pueden buscar los recursos por su identificador.
- Cuando se accede a un recurso por su URI, proporcionar información útil usando estándares como RDF o su lenguaje de consulta más extendido, SPARQL [109].
- Incluir enlaces a otras URI para que desde un recurso se puedan descubrir otros.

Para permitir operar correctamente a las distintas aplicaciones que requieren procesar contenido web, es importante llegar a un acuerdo sobre el formato estándar de este contenido. Al publicar datos siguiendo los principios de Linked Data, los datos se representan usando RDF, que proporciona un modelo muy simple pero muy apropiado a la arquitectura de la Web. Utilizar RDF en el contexto de Linked Data tiene una serie de beneficios aunque

también es necesario evitar algunas de sus características [111]. Entre los beneficios encontramos:

- Utiliza HTTP URI como identificadores únicos globales para elementos de datos y términos de vocabularios, por lo que está diseñado para ser usado a escala global y permite a cualquiera hacer referencia a cualquier recurso.

- Cualquier URI en un grafo RDF en la Web puede ser usada para recuperar información adicional, es decir, cualquier tripleta RDF sirve como punto de partida para explorar los datos.

- El modelo de datos de RDF permite establecer enlaces entre datos de distintas fuentes.

- La información de fuentes diferentes puede combinarse fácilmente integrando los conjuntos de tripletas de las distintas fuentes en un único grafo.

- RDF permite representar información que usa una representación distinta en un único grafo, lo que significa que se pueden mezclar términos de distintos vocabularios para representar los datos.

- Si se combina RDF con RDFS y OWL, el modelo de datos permite estructurar los datos tanto como se quiera.

Es recomendable utilizar las características de RDF mencionadas arriba y evitar las siguientes para facilitar el uso de Linked Data:

- Reificación de RDF: la reificación consiste en asignar una URI a una tripleta y utilizar esta como sujeto u objeto en otra tripleta. Se suele utilizar para proporcionar más información a una sentencia RDF, pero este tipo de tripletas son complejas de consultar con SPARQL.

- Colecciones y contenedores RDF: Se utilizan cuando el orden relativo de los elementos de un conjunto es significativo, pero al igual que en el anterior caso, son problemáticos a la hora de ser consultados con SPARQL.

- Nodos en blanco (blank nodes): Son nodos cuyo identificador es local y no pueden ser identificados más allá del grafo al que pertenecen, por lo que no pueden ser enlazados a recursos externos, yendo en contra de los principios de Linked Data.

La adopción de los principios de Linked Data ha llevado a la extensión de la Web a un espacio de datos global en el que datos de distintas fuentes y dominios están conectados entre sí. A este espacio global se le conoce como Web de Datos y contiene datos de dominios tan variados como personas, empresas, libros, publicaciones científicas, música, televisión y programas de radio, cine, genes, proteínas, ensayos clínicos, datos científicos, etc. Esto lleva a la aparición de nuevas aplicaciones que pueden explotar esos datos, trabajando en un espacio global y permitiendo obtener respuestas más elaboradas según nuevas fuentes de datos aparecen en la web [19].

El ejemplo más notable de uso de los principios de Linked Data es el proyecto Linked Open Data (LOD) de W3C [20], que tiene como objetivo impulsar la Web de Datos identificando conjuntos de datos existentes publicados bajo licencias abiertas, convertirlos a RDF siguiendo los principios de Linked Data y publicarlos en la Web. Es un proyecto abierto a cualquier que quiera publicar datos según Linked Data, lo que ha favorecido su éxito, el proyecto comenzó en 2007 con 13 conjuntos de datos y en abril de 2014 se contabilizaron un total de 1014 conjuntos. En [112] se puede ver el diagrama que representa los conjuntos de datos publicados y su interrelación en la actualidad. Berners-Lee propuso un esquema de despliegue de Linked Open Data con cinco niveles puntuados con estrellas [110]:

- $\star$ Los datos están disponibles en la web (en cualquier formato) con una licencia abierta. Se clasifican como Open Data.
- $\star\star$ Los datos están disponibles de forma estructurada por lo que pueden ser procesados por un ordenador.
- $\star\star\star$ Cumple el nivel 2($\star\star$) y además los datos están en un formato no propietario.
- $\star\star\star\star$ Cumple todos los niveles anteriores pero además usa estándares abiertos de la W3C (RDF y SPARQL) para identificar los recursos de manera que estos puedan ser enlazados.
- $\star\star\star\star\star$ Cumple todos los puntos anteriores y además los datos están relacionados con otros conjuntos de datos que les proporcionan más contexto.

Es decir, con las tecnologías de la Web Semántica se consiguen los niveles más altos de este esquema de estrellas. Utilizando RDF y con un uso apropiado de URI se alcanza el nivel de cuatro estrellas y enlazando los conjuntos de datos con datos externos se llega al nivel cinco.

3.2 Ingeniería ontológica

Ingeniería ontológica es la disciplina que investiga los principios, métodos y herramientas para el diseño, desarrollo y mantenimiento de ontologías. En la literatura existen diversas metodologías para construir ontologías que se pueden clasificar según distintos parámetros. En las primeras metodologías propuestas todo el proceso de desarrollo se realiza a mano, construyendo las ontologías desde cero. Algunas propuestas proponen un diseño colaborativo de las ontologías, mientras que en el aprendizaje ontológico se utilizan diferentes recursos como base para la creación automática de ontologías.

3.2.1 Creación manual de ontologías

En 1990 se presentó la metodología Cyc [113] de diseño de ontologías compuesta de tres fases. La primera fase consiste en la extracción manual de conocimiento común implícito en distintas fuentes. Una vez que tenemos suficiente conocimiento en nuestra ontología, la segunda y tercera fase consisten en la adquisición de nuevo conocimiento común utilizando herramientas de lenguaje natural o aprendizaje computacional. En la segunda fase este proceso se realiza de forma manual con la asistencia de herramientas software, mientras que la tercera fase se basa en el uso de las herramientas software como única fuente de adquisición de nuevo conocimiento.

La ontología Enterprise se construyó siguiendo la metodología presentada por Uschold y King [114]. Esta metodología propone tres pasos principales para la construcción de una ontología: (1) identificar el propósito de la misma; (2) capturar los conceptos, las relaciones entre estos conceptos, y los términos usados para referirse a dichos conceptos y a las relaciones entre ellos; y (3) codificar la ontología. Los principios de esta metodología influyeron en muchas de las propuestas posteriores en la comunidad ontológica [115].

La metodología propuesta por Grüninger y Fox [116] se basa en la experiencia de desarrollo de la ontología del proyecto TOVE. Propone un método para formalizar la construcción de ontologías basada en los siguientes pasos: (1) identificar el escenario que motiva la necesidad de crear la ontología; (2) formular preguntas en lenguaje natural, llamadas cuestiones de competencia, para determinar el ámbito de la ontología; (3) especificar la terminología de la ontología en un lenguaje formal; (4) definir las cuestiones de competencia formalmente; (5) especificar los axiomas y definiciones para los términos de la ontología formalmente; (6) establecer las condiciones de completitud de la ontología.

En [117], dentro del proyecto Esprit KACTUS, se construye una ontología

sobre una base de conocimiento por medio de un proceso de abstracción. Se comienza construyendo una base de conocimiento para una aplicación específica. A continuación, cuando se necesita una nueva base de conocimiento en un dominio parecido, se generaliza la primera base de conocimiento en una ontología y se adapta para las dos aplicaciones, repitiendo el proceso para nuevas aplicaciones. Cada vez que se desarrolla una aplicación se siguen estos pasos: (1) especificación de la aplicación; (2) diseño preliminar basado en categorías ontológicas top-level relevantes; (3) refinamiento y estructuración de la ontología.

METHONTOLOGY [118; 119] es una metodología para construir ontologías tanto desde cero como reutilizando ontologías existentes de forma directa o aplicando re-ingeniería. Permite la construcción de ontologías a nivel de conocimiento. El framework consiste en: (1) identificación del proceso de desarrollo de la ontología, donde se incluyen las principales actividades (evaluación, configuración, gestión, conceptualización, integración, implementación, etc); (2) un ciclo de vida basado en prototipos evolucionados; (3) y la metodología en sí, que especifica los pasos a tomar para llevar a cabo cada actividad, las técnicas usadas, los resultados a obtener y como se han de evaluar. Propone un proceso de desarrollo compuesto por tres categorías, divididas a su vez en subcategorías como sigue:

- Actividades de gestión:
  - Planificación de las tareas a realizar, su gestión, el tiempo a dedicarles, y recursos necesarios.
  - Control de la completitud de las tareas.
  - Control de calidad de los resultados.
- Actividades de desarrollo:
  - Pre-desarrollo
    - Estudio del contexto
    - Estudio de viabilidad
  - Desarrollo
    - Especificación
    - Conceptualización
    - Formalización
    - Implementación
  - Postdesarrollo

    * Mantenimiento
    * Reutilización

- Actividades de soporte, las cuales se realizan al mismo tiempo que las actividades de desarrollo:
    - Adquisición de conocimiento
    - Integración
    - Fusión
    - Alineamiento
    - Evaluación
    - Documentación
    - Gestión de la configuración

La metodología basada en SENSUS [120] es un enfoque top-down para derivar ontologías específicas del dominio a partir de grandes ontologías. Los autores proponen identificar un conjunto de términos semilla que son relevantes en un dominio en particular. Tales términos se enlazan manualmente a una ontología de amplia cobertura. Los usuarios seleccionan automáticamente los términos relevantes para describir el dominio y acotar la ontología SENSUS. Consecuentemente, el algoritmo devuelve el conjunto de términos estructurados jerárquicamente para describir un dominio, que puede ser usado como esqueleto para la base de conocimiento. Esta metodología recomienda los siguientes pasos: (1) tomar una serie de términos como semillas; (2) enlazar los términos manualmente; (3) incluir todos los conceptos en el camino que va de la raíz de SENSUS a los conceptos semilla; (4) añadir nuevos términos relevantes al dominio; (5) opcionalmente, añadir para aquellos nodos por los que pasan más caminos su subárbol inferior.

3.2.2 Creación colaborativa descentralizada de ontologías

Las metodologías expuestas hasta ahora están pensadas para ser desarrolladas en un entorno centralizado, donde el equipo implicado en la ingeniería de la ontología se concentra en un lugar, y se producen reuniones presenciales, normalmente se trata de desarrollos dentro de una misma empresa. Frente a esto, en el desarrollo descentralizado de ontologías, el equipo de desarrollo está disperso geográficamente, mientras deben desarrollar y mantener una ontología compartida.

DILIGENT [121] es una metodología desarrollada para abordar la ingeniería ontológica descentralizada. Esta metodología pone especial énfasis en dar soporte al proceso de argumentación necesario para ponerse de acuerdo en cambios en la ontología. La metodología DILIGENT incluye cinco actividades principales: (1) construir; (2) adaptación local; (3) análisis; (4) revisión; (5) actualización local. El escenario que plantea es un pequeño grupo de usuarios, expertos del dominio, ingenieros de conocimiento, y de construcción de ontologías, construyen una ontología inicial que no necesariamente cubre todo el dominio. Una vez que la ontología inicial está disponible, los usuarios pueden utilizarla y adaptarla localmente a sus propósitos. Lo normal es que en este punto las ontologías locales evolucionen. En DILIGENT existe la ontología compartida y la ontología local. La compartida está disponible a todo el mundo y no puede ser modificada, mientras que los usuarios pueden realizar los cambios que quieran en las ontologías locales, que son una copia de la ontología compartida. Un panel de expertos analiza las ontologías locales y los usuarios tratan de identificar similitudes con sus ontologías, en este punto se decide qué cambios deberían incluirse en la ontología compartida. La ontología compartida debe ser revisada de forma regular. Una vez que una nueva versión de la ontología compartida se hace disponible, los usuarios pueden actualizar sus ontologías locales a la nueva versión.

Otras aproximaciones de desarrollo colaborativo de ontologías proponen el desarrollo de una misma ontología común a partir de ontologías individuales, es decir, utilizando integración de ontologías. La propuesta definida en [122] presenta un entorno cooperativo para la integración de ontologías. El entorno se divide en las fases de selección, instanciación y transformación. El proceso de selección se ocupa de decidir qué ontologías tomarán parte en el proceso de integración. Para ello, por cada ontología, se calcula su conjunto de ontologías compatibles. Cuando todos los conjuntos han sido calculados, se selecciona el mejor conjunto de ontologías. En el proceso de instanciación se armonizan las terminologías en todo el conjunto de ontologías a integrar. La elección de la terminología más apropiada para ser incluida en la ontología derivada de la integración es un proceso dinámico y se realiza con una función que depende de varios parámetros: (1) el agente que solicita la integración; (2) el hecho de que cada ontología presente en la ontología final debe ser consistente con el resto; (3) el hecho de que ninguna ontología presente en la ontología final puede ser redundante con el resto; y (4) la cantidad de conocimiento que una ontología contiene. En la última fase del proceso, todas las ontologías se unen y se crea la ontología final transformada.

3.2.3 Reutilización de ontologías

En los sistemas basados en conocimiento, la captura de conocimiento es una tarea compleja y que lleva tiempo, por lo que construir una base de conocimiento desde cero requiere un gran esfuerzo. Se convierte en necesaria la reutilización de conocimiento para poder construir sistemas de conocimiento de buena calidad, basados en el trabajo y experiencia de otros. Las ontologías, entendidas como medio para compartir y reutilizar el conocimiento, proporcionan las piezas reutilizables de conocimiento declarativo, que junto a métodos de resolución de problemas y servicios de razonamiento permiten construir sistemas basados en conocimiento de alta calidad [123].

La reutilización de contenido ontológico existente se considera una buena práctica en ingeniería ontológica. Cada vez más, el desarrollo de ontologías es un proceso centrado en la reutilización [124], donde se combinan partes de ontologías y recursos existentes.

El método expuesto en [125] describe un método de reutilización de ontologías basado en reingeniería, que se aplica al caso de uso de construcción de una ontología sobre contaminantes del medio ambiente. El proceso define los siguientes pasos: (1) selección de los candidatos a reutilizar; (2) reingeniería, donde por un proceso de ingeniería inversa se obtiene un modelo conceptual a partir de la ontología fuente, el resultado se revisa y reestructura, para finalmente reconstruir la ontología según el modelo conceptual obtenido; (3) unión de las ontologías seleccionadas en un producto final.

En el experimento llevado a cabo por Uschold y Healy en [126], se reutiliza una ontología de ingeniería matemática para detallar la especificación de una aplicación software. En el proceso detallan los pasos más importantes (1) comprender la ontología y encontrar la parte reutilizable; (2) transformar la ontología (a una representación a nivel de conocimiento); (3) especificar la tarea y refinar la ontología a código ejecutable; (4) verificar que el código ejecutable corresponde con la especificación de la ontología base; (5) integrar la ontología en la aplicación.

En [127] se describe un caso de uso para construir una ontología en el domino de las campañas aéreas. La ontología se construye utilizando reutilización, aunque no sigue ninguna metodología específica, ejecuta los siguientes pasos: (1) selección de ontologías candidatas, en este caso, una ontología general sobre el tiempo y dos ontologías en el dominio; (2) transformación de formato de las ontologías; (3) unión de las ontologías de dominio; y (4) integración con la ontologías de tiempo.

En [128] se propone el desarrollo de ontologías como módulos de forma análoga a las metodologías de ingeniería del software, para fomentar la reuti-

lización, el desarrollo colaborativo y el diseño limpio. La metodología explota el mecanismo de importación de OWL y asume que cada módulo se mantiene como un fichero separado. La ontología se construye utilizando interfaces y enlazando módulos.

En [124] se realiza una revisión sobre estos y otros casos de uso [117; 129; 130; 131]. En ellos los autores coinciden en la complejidad de la labor de reutilización de las ontologías, directamente relacionada con la complejidad de las ontologías a reutilizar, y destacan que las distintas etapas en la reutilización de una ontologías (selección, transformación, unión e integración) no son triviales. Además, apuntan a la necesidad de propuestas de reutilización orientadas a la tarea que se ha de realizar y de definir una metodología bien detallada de reutilización. También destacan la necesidad de herramientas maduras que permiten automatizar parte del proceso de reutilización. Simperl [124] propone, basándose en el estudio realizado y sus propios casos de uso, los requisitos que debe cumplir una metodología y su nivel de detalle, que pasa por los siguientes pasos: (1) descubrimiento de ontologías, (2) selección de aquellas a reutilizar, (3) personalización de las ontologías relevantes, (4) integración en una ontología de aplicación. Además, la metodología debería seguir un ciclo de vida incremental que permitiera monitorizar y mejorar los procesos intermedios.

Por lo tanto, vemos que en el desarrollo de ontologías existe una carencia de metodologías de reutilización. Esto dificulta la reutilización de ontologías existentes para el desarrollo de nuevas, así como la complejidad de las ontologías a reutilizar. Bajo el supuesto de que existen clases de problemas que pueden ser resueltos aplicando soluciones comunes, se sugiere dar soporte a la reutilización en el ámbito del diseño de ontologías por medio de la definición y uso de patrones [92].

3.2.3.1 Patrones de diseño ontológico

Los patrones de diseño ontológicos (PDO) expresan soluciones comunes de modelado y buenas prácticas para un problema de diseño ontológico recurrente, y ayudan a los creadores de ontologías a utilizar mejor la expresividad y rigor del lenguaje de representación del conocimiento [92; 132].

Los PDO se agrupan en categorías según tipo de solución de modelado. Se dividen en léxico sintácticos, de presentación, de razonamiento, de correspondencia, estructurales, y de contenido [133].

- PDO léxico sintácticos: son patrones que conectan constructores de lenguaje, en lenguaje natural, con constructores ontológicos.
- PDO de presentación: incluyen convenciones de nombres y esquemas

de anotación. Definen buenas prácticas para presentar y documentar las ontologías y sus elementos.

- PDO de razonamiento: definen formas comunes de aplicar razonamiento en una ontología, como procedimientos de clasificación o normalización.

- PDO de correspondencia: incluyen PDO de re-ingeniería y PDO de alineamiento. PDO de reingeniería proporcionan a los diseñadores soluciones al problema de transformar un modelo conceptual, que puede ser una ontología o un recurso no ontológico, a otra ontología. PDO de alineamiento son patrones para crear asociaciones semánticas entre dos ontologías existentes.

- PDO estructurales: incluyen PDO lógicos y de arquitectura. Se encargan de la estructura lógica de la ontología, la expresividad del lenguaje de modelado y problemas relacionados, tanto a nivel local (PDO lógicos), como a nivel global (PDO de arquitectura).

- PDO de contenido (o de dominio): son patrones estructurales, que también restringen la estructura del modelo, pero además proporcionan soluciones para modelar conceptos específicos. Tienen un vocabulario explícito no-lógico para un domino específico de interés y dependen del contenido (el dominio), aunque el dominio puede ser muy general.

En [92] y [134] proponen los PDO de contenido, y su uso en métodos de construcción de ontologías basados en reutilización.

Los PDO de contenido contienen conceptos y relaciones que pueden ser especializados cuando el patrón es utilizado. Para utilizarlos se debe hacer una selección de los patrones más adecuados al dominio del problema de modelado, y aplicarlos a la ontología final por medio de especialización, importación, composición o expansión. Los PDO de contenido se caracterizan por [134]:

- Ser componentes computacionales, pues se representan y codifican en algún lenguaje lógico, como OWL, para que puedan ser procesados y reutilizados como componentes básicos en la construcción de ontologías.

- Ser pequeños y autónomos, facilitando el diseño de redes de ontologías mediante la modularización de las mismas.

- Ser componentes que permiten la inferencia, por lo que un sólo elemento, o una clase sin axiomas asociados, no puede ser un patrón.

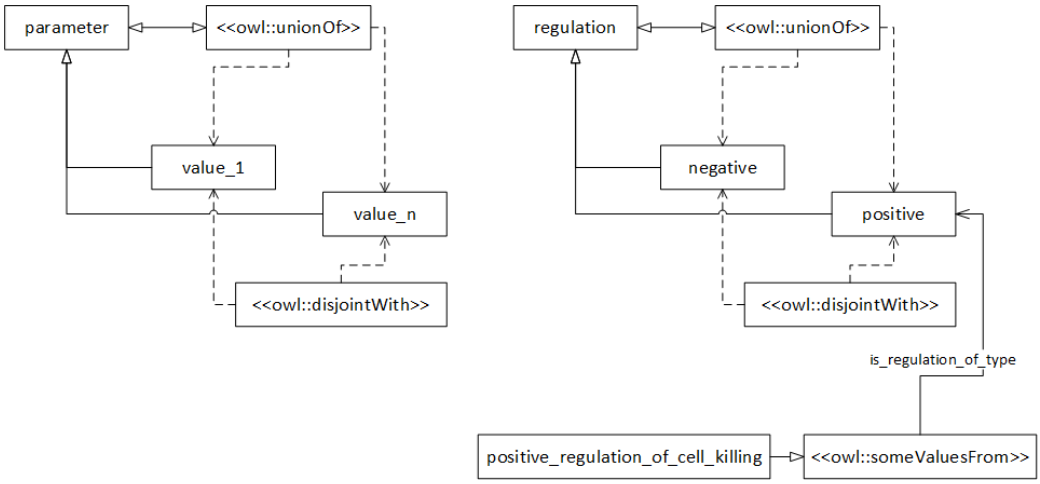

Figura 3.3: (izq.) Patrón de diseño ontológico de contenido para un "Value Partition", (dcha.) Aplicación del patrón "Value Partition" para modelar una regulación biológica, con sólo puede ser positiva o negativa

- Ser componentes jerárquicos, todos participan en un orden parcial, donde la relación de orden se llama especialización. La especialización requiere que al menos una entidad del patrón más específico sea incluida por al menos una entidad del patrón más general.
- Ser componentes relevantes cognitivos. Su visualización debe ser intuitiva y compacta, recogiendo las nociones relevantes del dominio.

La mayoría de bio-ontologías no utilizan toda la expresividad que permiten los lenguajes de representación de conocimiento como OWL, como resultado, carecen de la riqueza axiomática y rigor de dichos lenguajes [135]. Como los PDO de contenido guían a los creadores de ontologías en la explotación de las capacidades de los lenguajes de representación de conocimiento para resolver problemas concretos, pueden ser utilizados por biólogos para crear bio-ontologías axiomáticamente ricas y rigurosas [136]. La figura 3.3 izquierda muestra el diagrama de un patrón de diseño ontológico de contenido modelando una estructura "value partition"(particiones de valor), que representa un conjunto cerrado de valores para un parámetro en particular por medio de los axiomas "unionOf" y "disjointWith". La parte derecha de la figura 3.3 muestra la aplicación del patrón para modelar una regulación biológica, que solo puede tomar uno valor de un conjunto de dos, positivo o negativo.

Portales como Ontology Design Patterns [137] recopilan colecciones de

PDO para facilitar su aceptación en la comunidad. En [138] encuentran una carencia de herramientas que especifiquen la utilización de un patrón por parte de una ontología, por ello, proporcionan un lenguaje y un framework para definir de forma explícita los patrones y su uso dentro de ontologías OWL. El lenguaje Ontology Pre-Procesing Language [138] es un lenguaje declarativo para la consulta y modificación de ontologías expresadas en OWL. En [139] se propone un sublenguaje de OPPL para la definición de patrones. Los patrones OPPL encapsulan estructuras recurrentes de conocimiento expresadas en OWL, de manera que pueden ser utilizados sobre las ontologías OWL de forma inmediata. Los patrones OPPL están formados por variables y acciones, donde las variables pueden ser OWL Class, OWL objectProperty, OWL dataProperty, OWL Individual y constant. Las acciones pueden ser añadir (ADD) o eliminar (REMOVE) un axioma, que pueden ser cualquier axioma OWL-DL construido como combinación de variables y entidades de la ontología.

3.2.3.2 Patrones SHN

Como he comentado en el capítulo anterior, la red SemanticHealthNet (SHN) [90], que busca la consecución de la interoperabilidad semántica de la información clínica, propone una infraestructura semántica basada en un framework ontológico y un conjunto de patrones ontológicos de contenido como solución complementaria a los diferentes estándares de HCE existentes. El framework ontológico está formado por tres ontologías, cuya relación entre sí podemos ver en la figura 3.4.

- Una ontología de alto nivel, BioTopLite (prefijo btl) que proporciona un conjunto básico de clases y relaciones fundamentales.
- Una ontología de dominio, SNOMED CT (prefijo sct). Es una terminología clínica parcialmente construida con principios de ontología formal. De ella, conceptos seleccionados se colocaran por debajo de las clases de alto nivel proporcionadas por BioTopLite.
- Una ontología de entidades de información de HCE (prefijo shn:) para representar información como diagnósticos o resultados de actividades clínicas, como observaciones, investigaciones o evaluaciones. Todas las clases de esta ontología se representan como subclases de la clase de alto nivel btl:InformationObject.

Los patrones ontológicos de contenido crean una vista parcial sobre el framework ontológico subyacente, de manera que ayudan a modelar la información clínica haciendo uso del framework pero evitando que el usuario

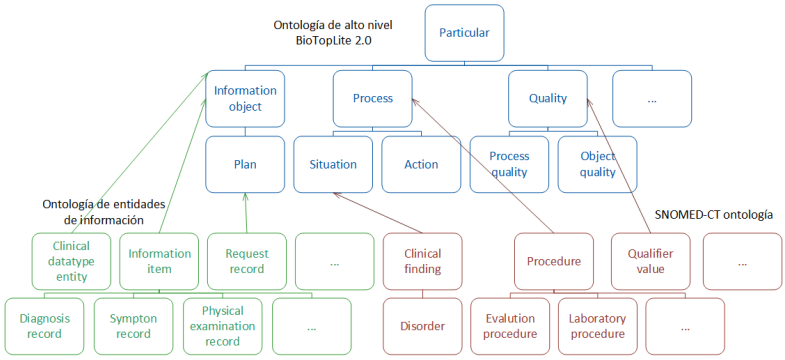

Figura 3.4: Fragmento del framework ontológico SHN

necesite entender por completo las complejas expresiones ontológicas subyacentes. SHN propone una representación formal de los patrones basada en lógica descriptiva, lo que permite utilizar razonamiento para detectar equivalencias entre información clínica representada utilizando diferentes modelos de información y terminologías [140].

La tabla 3.1 muestra las tripletas que representan el patrón que define la prescripción de un medicamento.

Tabla 3.1: Representación basada en tripletas del patrón Administración de Medicación

Sujeto	**Predicado**	**Objeto**
btl:Plan	isRealizedBy	sct:MedicationAdministration
sct:MedicationAdministration	hasFocusOn	sct:PharmaceuticalProduct
sct:MedicationAdministration	hasRoute	sct:RouteOfAdministration
sct:MedicationAdministration	hasStartTime	btl:PointInTime
sct:MedicationAdministration	hasEndTime	btl:PointInTime
sct:MedicationAdministration	hasDuration	btl:Duration
sct:MedicationAdministration	hasFrequency	shn:Frequency
sct:PharmaceuticalProduct	hasComponent	sct:Substance
sct:PharmaceuticalProduct	hasDose	shn:PhysicalQuantity
sct:PharmaceuticalProduct	hasForm	shn:DrugDosForm
sct:Substance	hasStrength	shn:PhysicalQuantity
sct:Substance	hasForm	shn:DrugDoseForm
shn:PhysicalQuantity	hasValue	xml:double
shn:PhysicalQuantity	hasUnits	shn:MeasurementUnits

Este patrón permite definir la prescripción por producto y por principio activo. Permite codificar el producto administrado, los ingredientes activos que lo componen, la dosis y forma del producto, la potencia y forma de cada ingrediente activo, el día de comienzo y fin del tratamiento, la duración del tratamiento, la vía de administración, y la frecuencia de administración.

La representación OWL DL de este patrón siguiendo el framework ontológico de SHN se muestra en la tabla 3.2. El sujeto (SUB) y el objeto (OBJ) corresponden con clases de la ontología y el predicado a una expresión OWL DL.

Tabla 3.2: Representación OWL DL del patrón Administración de Medicación

Predicado	**expresión OWL DL**
isRealizedBy	SUBJ subClassOf **btl:hasRealization** only OBJ
hasFocusOn	SUBJ subClassOf **btl:hasPatient** some OBJ
hasRoute	SUBJ subClassOf **btl:includes** some OBJ
hasStartTime	SUBJ subClassOf **btl:projectsOnto** some OBJ
hasEndTime	SUBJ subClassOf **btl:projectsOnto** some OBJ
hasDuration	SUBJ subClassOf **btl:projectsOnto** some OBJ
hasFrequency	SUBJ subClassOf **btl:isBearerOf** some (shn:Frequency and **btl:projectsOnto** only OBJ)
hasComponent	SUBJ subClassOf **btl:hasComponentPart** some OBJ
hasDose	SUBJ subClassOf **btl:isBearerOf** some (shn:DrugDoseForm and **btl:projectsOnto** only OBJ)
hasForm	SUBJ subClassOf **blt:isBearerOf** some (shn:DrugDoseForm and **btl:projectsOnto** only OBJ)
hasStrength	SUBJ **subClassOf btl:isBearerOf** some (shn:Strength and **btl:isRepresentedBy** only OBJ)
hasForm	SUBJ subClassOf **blt:isBearerOf** some (shn:DrugDoseForm and **btl:projectsOnto** only OBJ)
hasValue	SUBJ **shn:hasValue** some OBJ
hasUnits	SUBJ **shn:hasInformationObjectAttribute** some OBJ

3.3 Actividades semánticas

3.3.1 Anotación semántica

La anotación semántica es el proceso de añadir metadatos semánticos a contenidos. Se trata de enlazar las entidades de los contenidos con sus descripcio-

nes semánticas [141]. Los procesos de anotación [142] pueden ser manuales, semi-automaticos, y automáticos.

- Los métodos manuales permiten al usuario acceder a un repositorio de recursos semánticos, proporcionando métodos de búsqueda, y seleccionar los términos de anotación. Estos métodos manuales necesitan la participación de expertos del dominio, que sean capaces de realizar anotaciones precisas.

- Los métodos automáticos buscan términos para ser usados en la anotación y sugieren los más adecuados. Estos métodos no requieren la interacción de los usuarios, sino la existencia de reglas de anotación, de cuya calidad dependerá la calidad de la anotación.

- Los métodos semi-automáticos realizan una búsqueda de términos relacionados para sugerir anotaciones al usuario. El usuario confirma la validez de las recomendaciones, y normalmente, el método aprende de estas validaciones para mejorar el proceso de recomendación.

3.3.2 Similitud Semántica

La adopción de ontologías para la anotación de entidades proporciona métodos para comparar entidades en aspectos que de otra manera no serían comparables [143]. El uso de ontologías y métodos de similitud semántica proporciona un mecanismo para comparar entidades provenientes de distintas fuentes de información, ayudando a su comunicación, integración y recuperación [144].

Un método de similitud semántica es una función que, dadas dos entidades, devuelve un valor numérico que refleja cómo de parecidos en significado son dichas entidades. Estos métodos utilizan una estructura semántica como contexto para la estimación de las similitudes entre entidades del dominio. Para relacionar entidades y conceptos entre fuentes diferentes, los conceptos extraídos deben ser comparados en términos de su significado. La similitud semántica ofrece los medios mediante los que este objetivo puede conseguirse [144].

Existen dos propuestas principales para el cálculo de la similitud semántica [143; 145; 146]:

- Los métodos basados en aristas [145] se basan en contar el número de aristas que hay en el camino del grafo entre los dos términos comparados. La técnica más común es la de distancia, que selecciona el camino más corto o la media de todos los caminos disponibles. Otra técnica es

la del camino común, que calcula la similitud directamente por la longitud del camino entre el ancestro común más cercano a los dos términos y el nodo raíz.

Para que esta propuesta funcione correctamente, deben cumplirse dos suposiciones: (1) los nodos y las aristas de la ontología se distribuyen de forma uniforme; y (2) las aristas al mismo nivel en la ontología corresponden a la misma distancia semántica entre términos. Esto se cumple raramente en ontologías biomédicas [143].

- Los métodos basados en nodos [147] utilizan como fuentes de datos principales a los nodos y sus propiedades. Se basan en comparar las propiedades relacionadas con los términos, ya sea directamente, con sus ancestros, o con sus descendientes. Un concepto que suele ser utilizado en estos métodos es el de "Contenido de Información", que da una medida de cómo de específico e informativo es un término. El concepto de "Contenido de Información" puede ser aplicado a los ancestros comunes de los dos términos, para cuantificar la información que comparten y así medir su similitud semántica. Esto se puede hacer de dos maneras: la técnica del ancestro común más informativo, en el que solamente se tiene en cuenta el ancestro común como mayor contenido de información; y la técnica de los ancestros comunes disjuntos, en la que se consideran todos los ancestros comunes disjuntos (los ancestros comunes que no integran a ningún otro ancestro común).

Los métodos basados en aristas y en nodos trabajan explotando información de la estructura y contenido de la información de los términos en una jerarquía, y por lo tanto, son adecuados para comparar términos de la misma ontología. Como la estructura y contenido de la información de diferentes ontologías no es directamente comparable, existen otros métodos para estos casos [144]:

- Los métodos basados en características miden la similitud entre dos términos como una función de sus propiedades o basándose en sus relaciones con otros términos similares en la taxonomía.

- Los métodos híbridos son una combinación de todas las propuestas anteriores [148]. La similitud de términos se calcula comparando sinónimos, términos vecinos y características de los términos. Las características de los términos se distinguen en partes, funciones y atributos.

Una observación importante y una propiedad deseable en la mayoría de métodos de similitud semántica es que asignen similitudes mayores a términos

que estén más cerca (en términos de longitud de camino) y más abajo en la jerarquía (términos más específicos), que a términos que estén igualmente de cercanos pero más altos en la jerarquía (términos más generales) [144].

La similitud semántica se ha convertido en una herramienta muy útil para validar los resultados provenientes de estudios biomédicos como clustering de genes, análisis de datos en expresión de genes, predicción y validación de interacciones moleculares, y priorización de genes patológicos. La Gene Ontology es la ontología más investigada en el estudio de la similitud semántica en biología molecular, no solo porque es la ontología más ampliamente adoptada por la comunidad científica en ciencias de la vida, sino también porque comparar productos de genes a nivel funcional es crucial para una variedad de aplicaciones [143].

3.4 Web Semántica Biomédica

La Web Semántica se ha propuesto como espacio tecnológico en el cual la información biomédica pueda ser integrada y explotada [17]. La Web de Datos surgida a partir de la aplicación de los principios de Linked Data permite publicar y compartir conjuntos de datos biomédicos, mientras que las ontologías permiten crear completos modelos de conocimiento. Todos los posibles usos de las ontologías en el dominio de la biomedicina están directamente relacionados con el hecho de que las ontologías son descripciones de las entidades de un dominio [51]. Entre estas aplicaciones destaca su uso como vocabulario controlado, como modelo de la estructura de esquemas de representación y para facilitar la consulta de datos usando los modelos de conocimiento que representan[51; 149].

Las ontologías pueden ser usadas como referencia en el estudio de un dominio, pues aportan una definición de entidades y clases que, incluso cuando no hay consenso en la comunidad para la definición de términos, sirve como base para la discusión sobre el dominio.

En el dominio biológico, dada una secuencia, se asocia a la misma anotaciones que describen características fisicoquímicas, comentarios como función, enfermedad, expresión, especie, nombres, etc. Estas anotaciones, tan necesarias en las bases de datos bioinformáticas, han impulsado el uso de vocabularios controlados que unifiquen términos, descripciones y conocimiento. El auge de las bio-ontologías como vocabulario controlado para anotar instancias de un dominio comenzó a finales de los 90, cuando la comunidad investigadora de tres especies, *Drosophila melanogaster* [71], *Mus musculus* [55] y *Saccharomyces cerevisae* [150], decidieron crear de forma conjunta tres catálogos: funciones moleculares, procesos biológicos y componentes ce-

lulares, que juntos se convirtieron en la Gene Ontology (GO) [21]. Desde entonces, GO es la ontología de referencia para la anotación y soporte de datos y resultados en la investigación biomédica. Las anotaciones enriquecen semánticamente las instancias de datos, lo que permite realizar un procesado más a fondo de estas. Por ejemplo, permiten la agrupación y clasificación de entidades, atendiendo a las anotaciones compartidas entre las mismas. Además, estas anotaciones también pueden ser utilizadas como índices, que permiten enlazar y acceder directamente a las entidades anotadas.

Las ontologías también se utilizan como modelos de representación de datos para dar una solución a la heterogeneidad de los esquemas de representación de los recursos biomédicos. Esta heterogeneidad, existente incluso cuando se trata de esquemas representando entidades semánticamente equivalentes, se debe a la existencia de diferentes formas de describir un dominio, e impide comparar e integrar esquemas sin una reconciliación previa a nivel de datos y esquema. Las ontologías pueden ser utilizadas para proporcionar un modelo de organización de los datos. La reconciliación de esquemas se basa en la idea de remodelar los esquemas siguiendo la representación definida por la ontología. Utilizar un modelo común para la representación de conceptos del dominio junto a un vocabulario controlado para el etiquetado facilita la consulta precisa y el análisis de los datos. Además, si los datos se transforman a una representación ontológica en un lenguaje que permite razonamiento, esta nueva representación puede ser usada para inferir conocimiento adicional. La capacidad de razonamiento también puede ser usada para identificar correspondencias entre ontologías y utilizarlos para la recuperación de información anotada con distintos recursos terminológicos.

Las ontologías capturan conocimiento sobre un dominio e incluyen restricciones a cumplir para pertenecer a una clase. Esto permite reconocer una instancia del dominio como miembro de una clase y por lo tanto, dado un conjunto de instancias, permite clasificarlas. Además, las ontologías pueden ser usadas como guía en el dominio y soportar procesos de ayuda a la decisión, pues las restricciones en una ontología reducen el espacio de posibilidades, lo que resulta útil en dominios grandes y complejos como el biomédico.

El uso de ontologías para el intercambio y la interoperabilidad semántica de la información biomédica está bastante extendido. El informe final del proyecto SemanticHEALTH [10] identificó a los estándares de Historia Clínica Electrónica (HCE), las ontologías y las terminologías como piezas clave para conseguir la interoperabilidad semántica. Así, a las diferentes propuestas de estándares de HCE basados en arquitectura dual y el uso de terminologías, se les unen el potencial de las tecnologías de la web semántica.

3.4.1 Recursos semánticos biomédicos

Las ventajas del uso de ontologías en la integración, consulta y explotación de la información biomédica ha provocado la multiplicación del número de ontologías disponibles. Si tenemos disponibles numerosas ontologías para representar los mismos dominios, si cada institución usa ontologías diferentes, o construye sus propias ontologías para el manejo de su información y la creación de nuevas aplicaciones, su uso pierde efectividad como herramienta de representación e integración de información biomédica. El consorcio Open Biomedical Ontologies trata de dar solución a este problema, creando la OBO Foundry [22], un experimento colaborativo formado por desarrolladores de ontologías sobre las ciencias de la vida que establecen una serie de principios a seguir en el desarrollo de ontologías con el objetivo de crear un conjunto de ontologías de referencia en el dominio biomédico. Entre los principios que deben cumplir, las ontologías (i) deben ser abiertas, es decir, disponibles para ser usadas sin ninguna restricción, (ii) definir un contenido claro y específico y ser ortogonales respecto al resto de ontologías que siguen los principios, de manera que se permita que dos ontologías diferentes sean combinadas a través de relaciones adicionales complementarias y se evite la creación de varias ontologías para un mismo dominio, (iii) deben estar expresadas en una sintaxis bien especificada, (iv) y contener un identificador único. Otros principios establecen que las ontologías tengan un desarrollo colaborativo, usen relaciones comunes definidas de forma no ambigua, proporcionen procedimientos que permitan la retroalimentación del usuario e identificación de versiones.

Una de las ontología perteneciente a la OBO Foundry, es la ya mencionada GO. Esta ontología se utiliza para describir el rol de genes y sus productos en un organismo. La secuenciación completa de genomas de organismos eucariotas puso en evidencia que un gran porcentaje de genes involucrados en las principales funciones biológicas son compartidos entre organismos. De esta manera la información disponible sobre estos genes y sus productos aporta conocimiento sobre todos los organismos en los que se encuentran y puede permitir utilizar las anotaciones biológicas en modelos experimentales manejables a otros organismos menos manejables basándose en la similitud de genes y proteínas. La constatación de las similitudes entre genes de distintas especies, llevó a la necesidad de crear un vocabulario controlado dinámico para unificar nomenclaturas entre recursos, que permitiera representar el conocimiento sobre el rol de los genes y sus productos en las células, teniendo en cuenta que el conocimiento que se tiene va incrementando con el tiempo y está sujeto a cambios. Con GO se trató de representar la diversidad de funcionalidad que caracteriza a las células eucariotas y evitar la definición poco precisa de términos. Para ello se crearon tres ontologías independientes:

(1) procesos biológicos, (2) funciones moleculares y (3) componentes celulares. La ontología *procesos biológicos* registra los procesos en los que el gen o sus productos están involucrados, siendo un proceso biológico la ejecución ordenada de una serie de funciones moleculares que llevan normalmente a la transformación de un producto. En *función molecular* se definen las actividades bioquímicas de los productos de los genes mientras que *componente celular* identifica los lugares en la célula donde se activa el producto de un gen.

Dadas las posibilidades de uso de ontologías en biomedicina, es importante proporcionar servicios que permitan acceder y explotar los recursos ontológicos disponibles. BioPortal [23] es un portal web creado por The National Center for Biomedical Ontology (NCBO) [151] que proporciona acceso a un repositorio de ontologías y terminologías biomédicas y pretende proporcionar un conjunto de servicios para un acceso común a las bio-ontologías disponibles y facilitar el desarrollo de nuevas ontologías mediante la reutilización de las ya existentes.

BioPortal proporciona un conjunto de servicios web [152] que permiten el acceso a las ontologías y su explotación desde aplicaciones software, así como la participación de la comunidad en la evaluación y evolución de las ontologías. Los servicios web que BioPortal proporciona permiten: obtener metadatos de una ontología; obtener términos ontológicos individuales; descargar ontologías completas; crear y obtener vistas ontológicas, las cuales son subconjuntos de una o más ontologías, lo que resulta útil para trabajar con secciones más pequeñas de las ontologías; obtener todos los detalles de los términos de una ontología; obtener instancias de una ontología; obtener la información esencial sobre ontologías y sus términos en RDF; añadir y obtener correspondencias entre términos de las ontologías; y añadir y obtener comentarios de términos de las ontologías. Además, proporciona servicios adicionales que hacen uso del repositorio de ontologías. Un anotador utiliza los términos de las bio-ontologías para etiquetar datos textuales; otro servicio proporciona un índice basado en una ontología que da acceso a datos disponibles de forma pública; por último, un servicio de recomendación proporciona el conjunto de ontologías que se adecúan a una serie de palabras clave dadas.

El Instituto Europeo de Bioinformática, parte del Laboratorio Europeo de Biología Molecular (EMBL-EBI) [153] ofrece acceso gratuito a datos provenientes de investigación en las ciencias de la vida. Desde el año 2013 tienen disponible una plataforma RDF (EBI RDF Platform [24; 154]) para el acceso a sus bases de datos por medio de tecnologías de la Web Semántica. La plataforma permite identificar los conjuntos de datos que están disponibles como Linked Data, explorar la estructura de dichos datos, descargar los

conjuntos de datos en formato RDF y consultar los datos por medio del lenguaje SPARQL. Actualmente tiene 6 recursos RDF disponibles, BioModels [155], que proporciona un repositorio de modelos computacionales de procesos biológicos; BioSamples [156], que engloba información de muestra de datos existentes en una de las bases de datos de ensayos del EBI, a las que normalmente hacen referencia otros datos en distintos repositorios; ChEMBL [157], un recurso de datos de pequeñas moléculas bioactivas con propiedades similares a los fármacos, cuyos datos están curados manualmente; Expression Atlas [158], recurso sobre patrones de expresión génica bajo diferentes condiciones biológicas; Reactome [159], recurso de datos curados manualmente sobre rutas biológicas; y UniProt [160], recurso integral y completo sobre secuencias de proteínas y sus anotaciones.

Las terminologías clínicas también tienen versión OWL. La Organización Mundial de la Salud (OMS) está utilizando tecnologías de la Web Semántica en el desarrollo de la versión 11 de la International Classification of Diseases (ICD) [84], una terminología para la clasificación de diagnósticos clínicos que es utilizada principalmente en epidemiología, gestión de la salud y otros propósitos clínicos. Las tecnologías de la Web Semántica se utilizan en el modelado de ICD, en la infraestructura software y para el soporte de procesos colaborativos. La herramienta iCAT [161] permite el desarrollo colaborativo de ICD-11. Se utiliza OWL para representar ICD-11, lo que trae diversas ventajas, como utilizar semántica bien definida, que garantiza que sea interpretada de forma uniforme o la posibilidad de utilizar razonamiento para comprobar inconsistencias en la ontología durante su desarrollo. La utilización de OWL permite la reutilización de ICD en otras ontologías biomédicas, o para otros propósitos en herramientas que requieren información procesable por máquinas. La versión ICD de OWL contiene de forma natural identificadores únicos para todas las entidades, lo que es crítico en su reutilización. Hacer ICD, siendo un estándar de clasificación utilizado a través de todo el mundo, más amigable para su procesado por ordenador y para la web permite reforzar su asimilación en sistemas de codificación. La utilización del mismo estándar que otras grandes terminologías médicas que utilizan OWL, hará más fácil a ICD crear enlaces o correspondencias entre ontologías, fomentando la interoperabilidad de las herramientas biomédicas. Como ontología OWL, ICD-11 puede ser cargada en el repositorio de BioPortal. Además, el uso de las tecnologías de la Web Semántica ha demostrado ser de utilidad en la herramienta de desarrollo colaborativo, pues proporcionan la flexibilidad para ajustarse a los cambios de requisitos y para soportar los procesos cambiantes de forma ágil.

En [162] se afirma que usando un lenguaje más expresivo como OWL,

muchas de las dificultades identificadas en SNOMED-CT podrían superarse. La representación semántica de las terminologías clínicas permite aportar significado semántico a la información contenida en la HCE y explotar dichos significado a través de mecanismos de razonamiento. Actualmente, la distribución de SNOMED-CT incluye un script para obtener su representación en OWL y son numerosos los trabajos que estudian su representación en dicho lenguaje y tratan de explotar semánticamente su contenido.

3.4.2 Linked Data Biomédico

Existen numerosos conjuntos de datos publicados en la llamada Web de Datos, siguiendo los principios de Linked Data. El proyecto LOD cuenta con conjuntos de datos de diversos dominios, entre los que actualmente contiene 83 conjuntos de datos en el dominio de las ciencias de la vida. La figura 3.5 muestra parte de la nube de LOD para los conjuntos de datos pertenecientes a las ciencias de la vida (en rosa) y sus enlaces a conjuntos de datos de otros dominios. Entre otros, en ella aparecen los conjuntos de datos de la plataforma EBI RDF, del proyecto Bio2RDF o Linked Life Data (LLD).

Bio2RDF [25] es un proyecto par la publicación de datos provenientes de distintos repositorios bioinformáticos de forma integrada siguiendo los principios de Linked Data. El objetivo de Bio2RDF es proporcionar la red más grande de datos Linked Data sobre las ciencias de la vida. Para ello, define un conjunto de directrices para crear datos RDF entrelazados entre sí con URI normalizadas basándose en los principios de Linked Data. Bio2RDF se caracteriza por [163]:

- Ser de código abierto y estar disponible gratuitamente para uso, modificación o redistribución.
- Actuar como un conjunto de directrices básicas para producir datos enlazados sintácticamente interoperables a través de todos los conjuntos de datos.
- No trata de reunir todos los datos en un único esquema global.
- Proporciona un red federada de puntos de acceso.

Actualmente, Bio2RDF está formado por 35 conjuntos de datos [164], creados a partir de fuentes en formato que van desde texto plano, a XML, SQL u otros formatos propios. Para cada una de las fuentes de datos Bio2RDF crea un conversor a RDF disponible para la comunidad.

Linked Life Data (LLD) proporciona una plataforma de datos como servicio que proporciona acceso a 25 bases de datos biomédicas públicas a través

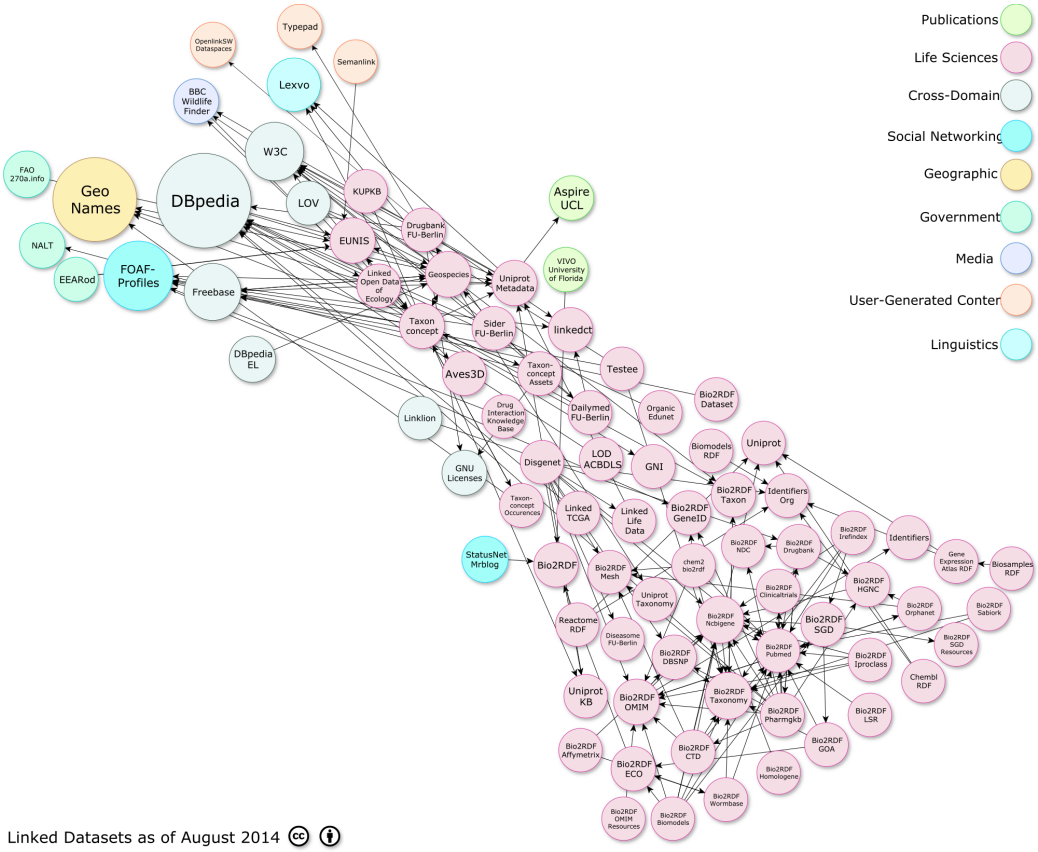

Figura 3.5: Sección de la nube LOD de conjunto de datos para las ciencias de la vida

de un único punto de acceso. Su conjunto de datos contiene datos sobre genes, proteínas, interacciones moleculares, rutas, dianas, fármacos, enfermedades y ensayos clínicos provenientes de bases de datos estructuradas como NCBI Gene, Uniprot, DrugBank, BioPAX, entre otras, repositorios de terminologías y de documentos como Pubmed.

OGOLOD [165] es un conjunto de datos basado en Linked Open Data (LOD) que representa información combinada sobre genes ortólogos y enfermedades genéticas del ser humano. Este conjunto de datos se ha construido a partir de las bases de datos de genes ortólogos: KOG [166], Inparanoid [63], HomoloGene [167] y OrthoMCL [168] junto a la base de datos de enfermedades genéticas humanas OMIM [169]. Este conjunto de datos se representa siguiendo la ontología OGO, que define todo el conocimiento del dominio de los genes ortólogos y las enfermedades genética del ser humano. Esta on-

tología utiliza bio-ontologías del dominio de las ciencias de la vida, Gene Ontology (GO) [21]; ECO [170], describe los tipos de evidencia científica en el ámbito de la investigación biológica, surgió para dar soporte a la anotación de productos de genes con términos de la GO; la taxonomía NCBI [79], una clasificación de organismos; RO [171], un conjunto de relaciones de alto nivel, como *part of* y específicas de biología, como *develops from*; y HPO [172], un vocabulario de anormalidades fenotípicas humanas. La inclusión de estos vocabularios comunes facilita la integración de OGOLOD con el resto de conjuntos presentes en la Web de Datos. Entre otros, contiene enlaces a conjuntos de datos de Bio2RDF.

3.5 Modelos clínicos en OWL

La existencia de varios estándares de HCE ha provocado que los sistema sanitarios de distintas instituciones utilicen diferentes estándares, perpetuando el problema de heterogeneidad de los mismos y dificultando la interoperabilidad semántica. El informe final del proyecto SemanticHEALTH [10] identificaba a los estándares de HCE, ontologías y terminologías como piezas clave para conseguir la interoperabilidad semántica. En los últimos años han aparecido varias propuestas de representación semántica de estándares de HCE para la gestión de la información y el conocimiento que contienen. Esto se debe al potencial de tecnologías como OWL, que permiten una representación formal de las entidades del dominio de información y del conocimiento que puede ser explotada de forma automática. La comunidad investigadora ha desarrollado ontologías para representar modelos de información y modelos clínicos y se han propuesto arquitecturas ontológicas para la interoperabilidad de los datos clínicos, modelos y aplicaciones.

3.5.1 Representación OWL de ISO 13606 y openEHR

3.5.1.1 Representación basada en individuos OWL

En [87] se propone una representación OWL para arquetipos de los estándares ISO 13606 y openEHR. Esta representación se obtiene analizando los modelos de referencia y de arquetipos de ambos estándares. Como resultado se obtienen tres ontologías, una para el modelo de referencia de ISO 13606, otra para el modelo de referencia de openEHR y otra para el modelo de arquetipos, común a ambos estándares. Las ontologías para los modelos de referencia definen las estructuras y tipos de datos a través de los cuales se estructura y representa la información clínica del arquetipo.

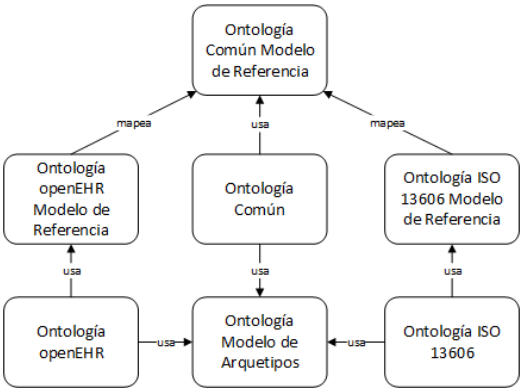

Figura 3.6: Relaciones entre las ontologías de representación de openEHR e ISO 13606

La ontología final para cada estándar importa la ontología de su modelo de referencia y la ontología del modelo de arquetipos, combinando ambos modelos. En esta propuesta se plantea una arquitectura para facilitar la interoperabilidad entre arquetipos de ambos estándares, definiendo una ontología común. Para definir esta ontología, primero se define una ontología común del modelo de referencia, que representa el conocimiento común y disjunto de cada una de las ontologías de los modelos de referencia de cada estándar. La ontología común final importa la ontología del modelo de arquetipos ya existente y la nueva ontología común del modelo de referencia. La figura 3.6 muestra las relaciones que hay entre todas las ontologías.

Los arquetipos ADL ISO 13606 y openEHR representados en OWL siguiendo esta arquitectura ontológica se definen como un conjunto de individuos pertenecientes a las clases de la ontología de ISO 13606 y openEHR respectivamente. Esta representación OWL basada en individuos ha sido utilizada para transformar arquetipos openEHR a ISO 13606 y viceversa [26]. Para realizar esta transformación, se identifican las similitudes y diferencias ente estándares para definir las correspondencias entre ambos. En general, estas correspondencias se definen a nivel de concepto, es decir, entre estructuras y tipos de datos, o propiedad, es decir, entre las propiedades de los conceptos. Se distinguen varios casos:

- Correspondencia entre conceptos:
  - Si existe un concepto con igual o parecido significado en el otro estándar se realiza la transformación de forma directa.
  - Si no existe ningún concepto con igual o parecido significado en

el otro estándar, el concepto se transforma a un concepto más general.

- Correspondencia entre propiedades:
  - Si ambas propiedades definen lo mismo y tienen el mismo tipo de datos, la transformación se hace de forma directa.
  - Si ambas propiedades definen lo mismo pero poseen diferente tipo de datos, se define la transformación entre tipos, cuando sea posible.
  - Si no existe una propiedad con el mismo o parecido significado en el otro estándar, no se transforma la propiedad.

La metodología de esta propuesta utiliza tecnologías de la Ingeniería de Modelos para realizar las transformaciones, tanto de arquetipos ADL a arquetipos OWL, como de arquetipos ADL ISO 13606 a arquetipos ADL openEHR y viceversa.

Por último, esta propuesta de transformación entre arquetipos ha sido aplicada para la transformación de datos clínicos basados en dichos estándares [27]. La metodología de transformación entre un arquetipo ISO 13606 a otro arquetipo openEHR (o viceversa) genera una conjunto de correspondencias conceptuales entre ambos arquetipos, que pueden ser reutilizadas en la transformación de extractos clínicos de ambos arquetipos.

La herramienta Poseacle Converter [173] implementa la transformación de arquetipos ADL a OWL y la transformación de arquetipos ISO 13606 a openEHR y viceversa. Ofrece una interfaz gráfica para la carga de modelos a transformar y la descarga de las transformaciones y un servicio RESTful para invocar la transformación a través de un API.

3.5.1.2 Representación basada en clases OWL

En [28] se propone una nueva ontología OWL para una representación de arquetipos openEHR basada en clases. Esta ontología se ha construido realizando una representación del modelo de referencia de openEHR usando la especificación del Ontology Definition Model (ODM) [174].

- El modelo de referencia en openEHR se define con un diagrama de clases UML, por lo que cada clase UML pasa a ser una `OWL:Class`.
- Los atributos de la clase UML pasan a ser propiedades de la clase OWL. Si el atributo es otra clase UML, pasa a ser una `OWL:ObjectProperty`, si no, pasa a ser una `OWL:DataProperty`.

- La cardinalidad del atributo se utiliza para declarar la propiedad funcional. Si el atributo no es multivaluado, la propiedad se declara como funcional.
- Relaciones de herencia entre las clases UML se transforman a axiomas `rdfs:subClassOf`.
- Las clases que comparten superclase son disjuntas, por lo que al transformarlas se añade el axioma `OWL:disjointWith`.
- Las relaciones de herencia entre atributos se transforman como axiomas `rdfs:subPropertyOf`.
- Cuando una clase tiene un atributo obligatorio, se utiliza el cualificador existencia en su restricción `rdfs:subClassOf` asociada.

Un arquetipo restringe las entidades del modelo de referencia, en concreto, restringe los atributos de cada entidad: rango, cardinalidad, etc. La representación de los arquetipos en OWL se hace definiendo una clase OWL con una restricción añadida por cada entidad restringida por el arquetipo.

Esta representación OWL de openEHR basada en clases se utiliza para validar la consistencia de los arquetipos respecto a su arquetipo padre y al modelo de referencia. La herramienta Archeck [175] implementa este método de validación. La herramienta transforma un arquetipo de entrada a su representación OWL basada en clases de la ontología comentada en esta sección y comprueba su consistencia con respecto a sus arquetipos padre, que también son transformador por la herramienta. Archeck también dispone de un servicio web para facilitar su integración con otras aplicaciones.

En [176] se utiliza otra representación OWL de arquetipos openEHR basada en clases. Esta propuesta define una ontología del modelo de referencia que se divide en cinco ontologías: tipos de datos, estructuras de datos, modelo demográfico, común y modelo de información. La ontología que define un arquetipo incorpora todas las ontologías anteriores, y reutiliza las clases de estas como clases padre de los nuevos conceptos creados para el arquetipo.

3.5.2 Representación OWL de CEM

El trabajo presentado en [177] propone una representación OWL para CEM. En esta propuesta, cada modelo CEM específico se convierte a clases de una ontología OWL definida sobre una meta-ontología base. Esta meta-ontología representa el modelo abstracto de instancias de CEM así como las restricciones para definir a cada una de las categorías estructurales entre las que se puede clasificar un modelo CEM.

En la meta-ontología se crea una clase OWL por cada categoría estructural, es decir, contiene las clases *Statement*, *Component*, *Modifier*, *Attribution*, *Panel*, *Collection*, *SemanticLink* y *Annotation*, y se utilizan axiomas OWL DL para definir formalmente la semántica, restricciones y relaciones para estas categorías. Los atributos *item* y *data* se transforman a OWL:ObjectProperty para indicar las asociaciones entre modelos clínicos o el valor de su *value choice*. Así mismo, las propiedades *qual, mod* y *att* se utilizan para señalar la asociación de un CEM con sus *qualifiers, modifiers* y *attributions*.

En esta propuesta, cada modelo clínico CEM tiene una representación OWL basada en clases que especializa la meta-ontología. El objetivo de esta propuesta es almacenar los datos clínicos recogidos mediante dichos modelos clínicos en un repositorio RDF, modelados según la representación OWL de los modelos clínicos. Esto proporciona una base para la interoperabilidad entre aplicaciones que intercambian datos de pacientes, utilizando un formato procesable por un ordenador. Además, esta representación permite usar las tecnologías de la web semántica sobre los datos y los modelos clínicos, para comprobar su consistencia semántica, y aplicar razonamiento para obtener conocimiento adicional.

3.5.3 Representación OWL de HL7

En [178] se propone una representación ontológica del modelo de información utilizado en HL7 v3, RIM. La ontología se construye en OWL-DL. Las clases de HL7-RIM, los tipos de datos y los vocabularios de HL7 se modelan como clases top-level en la ontología. Las asociaciones entre clases se modelan de la siguiente forma:

- Las asociaciones uno a muchos implican la creación de una `OWL:ObjectProperty`, cuyo dominio es la clase fuente, y rango la clase destino de la asociación. La propiedad se define como funcional. Su propiedad inversa también se crea.

- Las asociaciones (0..1) a muchos generan una `OWL:ObjectProperty` cuyo dominio es la clase fuente, el rango la clase destino, y la cardinalidad máxima de la propiedad es igual a 1. Su propiedad inversa también se crea.

- Las asociaciones (0..1) a uno se modelan como en el caso anterior añadiendo a la propiedad la característica de ser funcional inversa, es decir, la propiedad inversa es funcional y sólo puede tomar un único valor por cada instancia de la clase destino.

- Las asociaciones (1..*) a muchos generan una `OWL:ObjectProperty` con la cardinalidad mínima igual a 1. Su propiedad inversa también se crea.

Los documentos HL7 CDA son creados como instancias de esta ontología, es decir, como individuos de las clases de la ontología.

Capítulo 4

Métodos para llevar la información a la Web Semántica

La información biomédica se caracteriza por estar distribuida en distintos sistemas, y hacer uso de distintos formatos de representación y terminologías. La investigación y el uso eficiente de esta información requiere el acceso a la misma de forma integrada, es decir, que haya una comunicación entre sistemas sin fisuras, que permita compartir y comprender la información con independencia de su origen. Para conseguirlo, es necesario abordar la heterogeneidad de la información a todos los niveles. Los recursos de información biomédica son heterogéneos a nivel estructural, de nomenclatura, semántico y de contenido [179]. Están formados por grande conjuntos de datos que tienen su propia complejidad de esquema (heterogeneidad estructural); cada recurso puede referirse al mismo concepto semántico o campo con sus propios términos o identificadores, lo que puede llevar a discrepancias semánticas entre muchas fuentes y también puede ocurrir justo lo contrario, ya que muchas fuentes pueden usar el mismo término para referirse a objetos semánticamente distintos (heterogeneidad semántica y de nomenclatura); por último hay que tener en cuenta que un mismo objeto semántico puede tener datos diferentes dependiendo del recurso de origen (heterogeneidad de contenido). Hoy en día, está aceptado que sin el uso de las tecnologías de la Web Semántica en general, y las ontologías en particular, es imposible solucionar el problema de la heterogeneidad semántica, y por consiguiente, alcanzar la interoperabilidad semántica de la información [180]. En el caso de la información biomédica, las ontologías juegan un rol fundamental en su interoperabilidad semántica y como espacio para la integración y explotación de la información [17].

Es en este escenario donde surge la necesidad de encontrar soluciones

para obtener una representación basada en tecnologías de la Web Semántica de contenidos almacenados y publicados en formatos tradicionales, como pueden ser las bases de datos relacionales o ficheros XML. Existen varios sistemas y herramientas que hacen uso de estas tecnologías para resolver problemas asociados con el acceso, gestión y manipulación de la información biomédica. Entre estos, existen varias herramientas y metodologías para la representación de información de forma semántica, las cuales persiguen distintos propósitos [181], como facilitar la integración de información heterogénea, dar acceso a información haciendo uso de una ontología, facilitar consultas sobre la información, hacer pública la información en la Web de Datos y conectarla con otros recursos Linked Data, o generar ontologías a partir de información (normalmente con el propósito de realizar integración o acceso a datos basado en ontologías).

Las metodologías de transformación de información a representación semántica siguen un esquema común. Parten de un repositorio de datos fuente, del que extraen los datos que serán transformados siguiendo un modelo semántico destino. El proceso de extracción y transformación de datos se basa en la definición de correspondencias entre el modelo que siguen los datos de origen y un modelo de salida que puede ser generado a partir de la fuente o existir previamente.

La mayoría de las metodologías expuestas en este capítulo no han sido desarrolladas para su uso específico en el dominio de la biomedicina, sin embargo, están orientadas a proporcionar un acceso a la información usando un formalismo común semántico, y orientadas a dominios en los que la interoperabilidad semántica, integración de información y el acceso unificado su explotación posterior para uso secundario y la necesidad de proporcionar fácil acceso a la información son elementos clave, como es el caso del dominio biomédico.

4.1 Definición del modelo semántico

El modelo semántico definido debe reflejar correctamente la semántica del dominio y permitir la representación consistente de los recursos fuente. En la definición de este modelo se distinguen dos casos: (1) el modelo de salida se crea desde cero partiendo del repositorio de entrada, en este caso las metodologías explotan la estructura de representación de entrada y el contenido del repositorio, (2) el modelo de salida se define en el dominio de la información utilizando técnicas de ingeniería ontológica.

Varias metodologías de creación de ontologías se mencionan en el capítulo anterior (ver sección 3.2). Dentro de la ingeniería ontológica, otra propuesta

es el aprendizaje de ontologías, que incorpora métodos de ingeniería ontológica, aprendizaje automático, representación de conocimiento, extracción de información y computación lingüística para crear ontologías a partir de datos disponibles en diversos formatos [182], reduciendo el esfuerzo de creación manual. En [183] se clasifican los datos relevantes en el aprendizaje de ontologías en: (1) datos en forma de ontologías, es decir, reutilizar ontologías existentes en la construcción de nuevas; (2) datos según esquemas, incluyendo esquemas de bases de datos, por ejemplo relacionales, y esquemas comunes en la web, como esquemas XML; (3) datos como instancias, es decir, datos almacenados en bases de conocimiento; (4) datos semi-estructurados, es decir, datos con cierta estructura pero con ausencia de un esquema fijo o rígido; (5) datos en lenguaje natural, que puede estar enriquecido con información semi-estructurada.

En este capítulo nos centramos en datos basados en esquemas y semi-estructurados, poniendo atención en aquellos más comunes en la representación de información biomédica (ver secciones 2.1.2 y 2.2.2).

Tim Berners-Lee dio una primera aproximación [184] a la relación entre el modelo de la web semántica y el modelo de bases de datos relacionales, proponiendo una correspondencia directa:

- Un registro o fila de una tabla es un nodo RDF.
- Cada atributo de la tabla (columna) es una RDF propertyType.
- Cada valor de un registro para un atributo es un valor de la propertyType.

En [181] se propone una ampliación de este mapeo básico para generar un esquema RDFS a partir de una base de datos relacional, quedando:

1. Cada tabla (relación) R se mapea a una clase RDFS C.
2. Por cada entrada en la tabla R se crea un nodo RDF I cuyo tipo es C, es decir, una instancia de la clase C.
3. Por cada atributo att en la tabla R se crea una propiedad RDF P.
4. Para cada entrada en la tabla R, el valor de el atributo att se asocia al valor de la propiedad P para el nodo I.

Esta aproximación se conoce como aproximación básica, y es ampliamente usada por muchas herramientas de transformación que buscan publicar información proveniente de bases de datos relacionales en la Web Semántica.

La W3C creó un grupo de trabajo, RDB2RDF Working Group [185], para la estandarización en la definición de correspondencias entre bases de datos relacionales y esquemas RDF y OWL. Uno de los resultados fue Direct Mapping [186], una estrategia para asociar datos relacionales a RDF.

Direct Mapping sigue la aproximación básica propuesta por Berners-Lee, pero es más específico en la definición de URI y la conversión de atributos en propiedades. Dada un base de datos de entrada siguiendo un esquema relacional, su grafo RDF directo se define como la unión de los llamados "grafos de tabla" de cada una de las tablas del esquema de base de datos, donde:

- El grafo de tabla es la unión de los grafos de fila de cada fila en la tabla.
- Un grafo de fila es un grafo RDF compuestos por: (1) la tripleta de tipo de fila, (2) una tripleta referencia por cada columna/s de la tabla que sea una clave ajena y cuyo valor no sea nulo, y (3) una tripleta literal por cada columna de la tabla cuyo valor no sea nulo.
- Una tripleta de tipo de fila es una tripleta cuyo sujeto es el nodo RDF de la fila, el predicado es rdf:type y el objeto es la URI de la tabla. El nodo de la fila se obtiene a partir de la clave primaria y la URI de la tabla se obtiene a partir del nombre de la tabla.
- Una tripleta literal es una tripleta cuyo sujeto es el nodo RDF de la fila, el predicado es la URI de la propiedad literal de la columna (construida con el nombre de la tabla y el nombre de la columna) y el objeto es un literal RDF obtenido a partir del valor de la columna.
- Una tripleta referencia es una tripleta cuyo sujeto es el nodo RDF de la fila, la URI es la propiedad referencia de la columna (construida con el nombre de la tabla, la palabra "#ref" y el nombre de cada columna que forma la clave ajena) y el objeto es el nodo RDF de la columna referenciada.

4.2 Definición de correspondencias

El primer paso para la transformación es la definición de asociaciones o correspondencias entre el modelo de entrada y el modelo de salida. Las asociaciones identifican los elementos del modelo de entrada que se corresponden con elementos del modelo de salida. Exceptuando aquellos métodos totalmente automáticos en los que la transformación sea directa y no dependa de

ninguna parametrización de entrada o que no permiten la reutilización de las asociaciones definidas, las asociaciones se representan en algún lenguaje específico, que normalmente es un lenguaje propio.

La definición de correspondencias puede ser un proceso manual, con intervención del usuario, o puede ser resultado de un proceso de alineamiento de esquemas. Los procesos de alineamiento de esquemas pretenden identificar las correspondencias semánticas entre estructuras de metadatos o modelos, como esquemas de bases de datos, XML u ontologías. Los procesos de alineamiento tienen como objetivo dar soporte al intercambio de datos, evolución de esquemas e integración de datos [187].

4.2.1 R2RML: RDB to RDF Mapping Language

El grupo de trabajo RDB2RDF Working Group creó el lenguaje R2RML [188] para expresar correspondencias entre bases de datos relacionales y conjuntos de datos RDF. Las correspondencias en este lenguaje están diseñadas para: (1) construir repositorios RDF a partir de las bases de datos relacionales, (2) acceder a la base de datos relacional a través de un punto de acceso virtual SPARQL o (3) crear una interfaz Linked Data.

Las correspondencia R2RML se expresa como un grafo RDF definido en sintaxis Turtle [189]. Las correspondencias R2RML asocian filas de una tabla lógica con tripletas RDF. Las tablas lógicas pueden ser tablas de una base de datos relacional, vistas sobre la base de datos, o una consulta SQL. La correspondencia se especifica con un mapa de tripletas, formado por:

- Una regla de sujeto (subject map), que define el sujeto de todas las tripletas RDF generadas por el mapa de tripletas para una fila de la tabla lógica. El sujeto se define a partir de una URI definida a partir de un campo de la fila, normalmente la clave primaria. Esta regla genera la tripleta que asocia el sujeto con una `rdfs:Class` a través del predicado `rdf:type`.

- Por cada tripleta en la que esté involucrado el sujeto, se define una regla predicado-objeto (predicate-object map). La regla utiliza información de la fila de la tabla lógica para definir las URIs del predicado y objeto de cada tripleta.

R2RML es un lenguaje independiente de la implementación, permite especificar con sus reglas la asociación de una base de datos relacional con un grafo RDF existente, o crearlo desde cero. Direct Mapping fue pensado como base para este lenguaje, cualquier implementación del lenguaje debería in-

cluir la creación de las correspondencias directas de manera que la definición con R2RML no tenga que partir de cero.

4.2.2 D2RQ Mapping Language

D2RQ mapping language [190] es un lenguaje de correspondencias declarativo para describir la relación entre un esquema de base de datos relacional y un vocabulario RDFS u ontología OWL.

Las correspondencias en este lenguaje se representan como un documento RDF en sintaxis Turtle. Una correspondencia D2RQ está formada por un ClassMap, que asocia una clase de la ontología con una URI construida con valores de la base de datos, como una clave primaria. Cada ClassMap contiene un conjunto de PropertyBridge, que especifican como se definen las propiedades de las instancias creadas por un ClassMap. Hay dos tipos de PropertyBridge, DatatypePropertyBridge, para especificar valores literales de una propiedad y ObjectPropertyBridgese para especificar como valor de la propiedad, URIs referidas a otros ClassMap.

4.3 Transformación de datos

La transformación de datos ofrece una representación semántica, guiada por las correspondencias definidas, de información representada según un modelo de entrada. En estos métodos la representación de salida normalmente se basa en RDF, RDFS u OWL.

Las distintas clasificaciones de estas metodologías realizadas en la literatura utilizan distintos criterios [181], atendiendo a la automatización del proceso de transformación distingue entre (1) procesos totalmente automáticos, (2) procesos semi-automáticos y (3) procesos manuales. Los procesos automáticos no requieren la intervención del usuario en ningún punto del proceso de transformación. La automatización total es más común en aquellos métodos que generan el modelo de salida a partir del recurso de entrada, por lo que sólo dependen del modelo de entrada. La existencia de un modelo de salida hace el proceso de transformación dependiente del modelo de entrada y del modelo de salida y, por lo tanto, la automatización total resulta más compleja. Los procesos semi-automáticos y manuales permiten y requieren respectivamente a un usuario personalizar el proceso de transformación. Atendiendo al tipo de acceso a los datos resultantes, distinguimos entre (1) los métodos que hacen una transformación completa de los datos, obteniendo un nuevo repositorio semántico con toda la información proveniente de las fuentes de datos, y (2) los métodos que crean vistas virtuales y

puntos de acceso a la información basados en tecnologías de la Web Semántica, de manera que se hace un acceso a los datos utilizando consultas, por ejemplo SPARQL, y estas se traducen a consultas sobre los repositorios de información originales.

4.3.1 Herramientas de transformación

En esta sección describiremos algunas de las herramientas existentes que permiten definir un asociación entre recursos de entrada y un modelo semántico de salida y que permiten el posterior acceso a la información.

4.3.1.1 D2RQ

D2RQ [191] es una plataforma que permite acceder a bases de datos relacionales a través de grafos RDF virtuales. Es decir, crea un acceso basado en RDF a la base de datos relacional sin necesidad de que la información de la base de datos se replique en un repositorio semántico. La plataforma define las correspondencias entre un esquema de base de datos relacional y una ontología utilizando D2RQ Mapping Language. Soporta tanto generación automática de correspondencias como semi-automática. En el modo automático la creación de la correspondencia con el grafo RDF virtual se realiza aplicando la aproximación básica explicada más arriba. El usuario tiene la posibilidad de modificar y personalizar estas correspondencias generadas automáticamente.

La plataforma dispone de un motor de acceso a los datos y un servidor D2R [192] que proporciona acceso a los datos a través de SPARQL, Linked Data y HTML.

4.3.1.2 Triplify

Triplify es una herramienta para hacer visible en la Web Semántica información utilizada por aplicaciones web y almacenada en bases de datos relacionales [193]. Para ello transforma datos relacionales a RDF y los publica como datos enlazados (Linked Data).

Esta herramienta convierte resultados de consultas SQL a un modelo RDF. Define un enlace entre una URI y una vista SQL sobre la base de datos relacional. Por lo tanto, utiliza SQL como lenguaje para definir las correspondencias. En la transformación se tiene en cuenta:

- La primera columna de la vista contiene el identificador que es utilizado para crear las URI de cada instancia generada con esta vista. Definida

la instancia, esta hará de sujeto del resto de tripletas definidas con esta vista.

- Los nombres de las columnas generan las URI de las propiedades (predicado de las tripletas).

- Los valores de las celdas de la vista contienen valores para propiedades de datos o referencias a otras instancias. Es decir, son los objetos de las tripletas.

Por defecto, Triplify considera que todas las propiedades creadas a partir de una vista toman como valor un literal. Esto se puede configurar con una anotación en las columnas, para indicar que se transformarán a una propiedad referenciando a la URI de otro objeto. También permite definir una anotación (alias) para los nombres de las columnas, de manera que se puede indicar que una columna se transforme a una propiedad existente en un vocabulario RDF externo.

En resumen, Triplify crea un modelo RDF a partir de la base de datos relacional, estando la transformación dirigida por ésta. El uso de vistas sobre la base de datos relacional hace la transformación más flexible y la posibilidad de anotar las columnas la enriquece con vocabularios existentes.

4.3.1.3 Linked Data Views de Virtuoso

OpenLink Virtuoso [194] es un sistema gestor de bases de datos y servidor de aplicaciones web que combina funcionalidad de bases de datos relacionales tradicionales, bases de datos objeto-relacionales, bases de datos virtuales, manejo de datos RDF y XML en un solo proceso multihilo. Entre sus funcionalidades, ofrece una vista RDF sobre una base de datos relacional a través de su herramienta Linked Data Views [195], que permite consultar datos relacionales con SPARQL a través de la definición de correspondencias.

En Virtuoso, el almacenamiento de RDF se realiza en cuádruplas, que contienen las tripletas (sujeto, objeto, predicado) y el identificador del grafo al que pertenecen. Las correspondencias de Linked Data Views se almacenan en patrones de cuádruplas, que explican como se puede construir una cuádrupla a partir de los datos relacionales. Además, pueden incluir búsquedas SQL adicionales para restringir el ámbito. Virtuoso ofrece un traductor SPARQL a SQL, que cuando se realiza una consulta SPARQL, se compila a un patrón de tripletas, busca los patrones de cuádruplas asociado a dichas tripletas y lo utiliza para recuperar los datos de la base de datos relacional. Virtuoso permite combinar el mapeo de datos relacionales con datos RDF

almacenados de forma nativa, pues puede procesar un consulta SPARQL para la que algunas tripletas procederán de repositorios relacionales y otras de repositorios RDF locales.

Virtuoso realiza una correspondencia guiada por la base de datos y sigue la aproximación "tabla relacional - clase RDF", "columna relacional a predicado RDF" a la hora de realizar las correspondencias, aunque permite restringir las consultas SQL generadas de forma manual.

4.3.1.4 XS2OWL

El modelo de transformación XS2OWL [196] permite la representación de esquemas XML en sintaxis OWL. El modelo toma como entrada un esquema XML y genera una ontología OWL principal que captura la semántica del esquema XML, y una ontología de correspondencias que contiene las correspondencias entre el esquema XML de entrada y la ontología principal de salida. La tabla 4.1 muestras las equivalencias entre los constructores de un esquema XML y los constructores OWL que se utilizan en esta transformación. La información de los elementos que no han sido transformados se utiliza para realizar una transformación inversa, y obtener el XML esquema a partir de la ontología OWL. Por lo tanto, utiliza OWL como lenguaje para definir las correspondencias.

XS2OWL 2.0 forma parte del framework SPARQL2XQuery 2.0 [197], que permite la consulta de datos XML según una ontología OWL. SPARQL2-XQuery utiliza la ontología de correspondencias para transformar consultas SPARQL sobre la ontología en expresiones XQuery (lenguaje de consulta para XML).

4.3.1.5 RDB2OWL

RDB2OWL [198] es una propuesta para transformar información almacenada en bases de datos relacionales a una representación basada en un esquema RDF o una ontología OWL preexistente. Las correspondencias entre el esquema relacional de entrada y la ontología OWL o esquema RDF de salida se almacenan en una base de datos relacional, cuyo esquema contiene las siguientes tablas:

- En la tabla class_map cada registro asocia una tabla t de la base de datos de entrada con una clase de la ontología o recurso RDF, es decir, genera una tripleta $\langle\ s_t, rdf:type, r\ \rangle$, donde r es la URI de la clase o recurso y s_t se genera a través de la información de los registros de la tabla t.

Tabla 4.1: Correspondencias entre los constructores XML Schema y sintaxis OWL en XS2OWL

Constructor XML Schema	**Constructor OWL**
Complex Type	Class
Simple Datatype	Datatype Definition
Element	(Datatype or Object) Property
Attribute	Datatype Property
Sequence	Unnamed Class - Intersection
Choice	Unnamed Class - Union
Annotation	Comment
Extension, Restriction	subClassOf axiom
Unique (Identity Constraint)	HashKey axiom
Key (Identity Constraint)	HasKey axiom - ExactCardinality axiom
Keyref (Identity Constraint)	Object Property Range
Substitution Group	SubPropertyOf axioms
Alternative	En la ontología de correspondencias
Assert	En la ontología de correspondencias
Override, Redefine	En la ontología de correspondencias
Errors	Datatype

- Tabla datatype_property_map genera la tripleta $\langle\ s,p,l\ \rangle$ donde p es la URI de un recurso o propiedad de tipo de datos, s se obtiene a través un registro de la tabla class_map y denota la URI de un recurso o clase y l es una expresión literal, cuyo valor se obtiene de la misma tabla y registro del que se obtuvo la información para construir l.

- Tabla object_property_map, incluye la información necesaria para crear una tripleta $\langle\ s,p,o\ \rangle$, dónde p es la URI de un recurso o propiedad objeto, y s y o se obtienen a partir de dos correspondencias de la tabla class_map.

La información sobre la base de datos relacional origen y la ontologías destino también se encuentra en varias tablas de la base de datos de correspondencias.

Las correspondencias son usadas para construir un script SQL que extrae datos de la base de datos relacional fuente y los almacena como tripletas en el esquema RDF o como individuos en la ontología OWL. Tener las correspondencia almacenadas en una base de datos relacional permite ejecutar validación SQL. Concretamente, se puede chequear si se ha omitido algún campo necesario para crear el script y realizar la extracción correctamente,

o comprobar si la correspondencia es consistente, por ejemplo, al asociar un propiedad, comprobar que el dominio y rango corresponde con el definido en la ontología.

En RDB2OWL la transformación está dirigida por el dominio de la ontología o grafo RDF destino. La formulación de las correspondencias es totalmente manual y en el caso de ontologías destino muy grandes, puede resultar compleja.

4.3.1.6 Karma

Karma [199] es una herramienta que permite asociar fuentes estructuradas como bases de datos a ontologías ya existentes para construir un modelo fuente. Este modelo fuente es una descripción semántica de la asociación entre los datos fuentes y la ontología destino, y puedes ser utilizado para crear tripletas RDF a partir de los datos de entrada o construir un punto de acceso SPARQL para consultar los datos. Es decir, Karma es una herramienta para la generación semi-automática de correspondencias entre una base de datos relacional y una ontología destino.

El modelo fuente es un conjunto de asociaciones que definen como los datos de entrada se definen según los términos de la ontología destino. El proceso de generación semi-automática de las asociaciones tiene tres pasos:

- (1) Asignación de tipos semánticos: este proceso consiste en asignar a cada columna fuente un nodo de la ontología. Esta asignación se hace automáticamente basándose en los valores de la columna y en modelos probabilísticos construidos en asignaciones anteriores. Si la asignación realizada es incorrecta, el usuario la puede corregir y el sistema aprende de esta corrección. Estos tipos semánticos son clases OWL o pares de una clase OWL y una propiedad OWL.

- (2) Construcción de un grafo de asociaciones: este grafo define todas las posibles asociaciones entre la fuente y el destino. Se construye utilizando la asignación de tipos semánticos realizada en el paso anterior.

- (3) Refinamiento del grafo de asociaciones: el grafo construido es refinado por el usuario. Este grafo es construido de manera que se puede resolver con un algoritmo de árbol de Steiner. El árbol de Steiner es un problema de optimización combinatoria que trata de buscar la interconexión más corta para un conjunto de elementos.

- (4) Generación de una especificación formal de las asociaciones a partir del grafo. Esta especificación formal define el modelo fuente resultado, el cual puede ser usado para construir tripletas RDF.

Al ser un proceso semi-automático, Karma aligera el proceso de creación del modelo fuente. Sin embargo, la función de asignación automática de tipos semánticos depende de la existencia de una base de conocimiento de asignaciones de tipos semánticos previas.

4.3.1.7 Populous

Populous [200] es una herramienta que asiste a expertos en el proceso de creación de una ontología. La herramienta hace uso de patrones para guiar el proceso de recogida del conocimiento y su posterior inclusión en la ontología en desarrollo. Estos patrones de diseño (ver sección 3.2.3.1) proporcionan plantillas que dan las pautas para facilitar el desarrollo de la ontología.

A partir de los patrones, Populous crea una interfaz para recogida de datos basada en una hoja de cálculo, de manera que proporciona un formulario basado en tablas de columnas para la recogida de datos. Las columnas de el formulario están asociadas a variables del patrón, de manera que asisten al usuario en la recogida de conocimiento y automáticamente pueblan las plantillas que se transformarán en instancias del patrón para la inclusión en la ontología. Populous utiliza OPPL para formalizar los patrones y una vez instanciados estos, modificar la ontología final con el nuevo conocimiento obtenido.

4.3.1.8 Sistema OGO

En [201] se propone una metodología para la integración de varios repositorios relacionales en un repositorio semántico basado en un modelo ontológico. Esta metodología de integración requiere la transformación de las bases de datos relacionales de entrada a una representación basada en una ontología global. Esta ontología global conceptualiza el conocimiento del dominio de los repositorios a integrar.

La transformación se lleva a cabo por medio de la definición de correspondencias entre el esquema relacional de los recursos de entrada y la ontología global. Estas correspondencias definen cómo las instancias de datos almacenadas en una base de datos relacional se representan como individuos de la ontología global. Las correspondencias se definen como un conjunto de reglas expresadas en un lenguaje propio basado en XML. La metodología distingue tres tipos de reglas: (1) las reglas de clase se utilizan para crear una instancia en la ontología y asociar su tipo (clase) adecuado, por lo tanto enlazan una clase de la ontología con la entidad equivalente del esquema relacional de entrada; (2) las reglas de propiedad se utilizan para asignar un valor a una propiedad de un individuo; por último, (3) las reglas de relación se utilizan

para asociar dos individuos distintos.

Como el objetivo final de la metodología es la integración de varios repositorios heterogéneos, existe la probabilidad de que entidades equivalentes estén representadas de forma diferente en más de un repositorio. La metodología tiene que detectar estas equivalencias para garantizar la correcta integración de los datos y evitar la redundancia en el repositorio final. Para conseguirlo se crean las reglas de identidad. Estas reglas se definen sobre las entidades de la ontología global y expresan los criterios de identidad para un individuo de cada clase de la ontología. Es decir, dados dos individuos que pertenecen a la misma clase, si los valores asociados a sus criterios de identidad son iguales, la regla concluye que se trata del mismo individuo.

Esta metodología se ha aplicado a una serie de repositorios biomédicos para dar soporte a la investigación traslacional. La metodología se utilizó para transformar un conjunto de repositorios con información de genes y proteínas ortólogos con repositorios de enfermedades genéticas. Para ello se definió una ontología global que representaba el dominio de los recursos de entrada y que incluía bio-ontologías externas.

4.4 Discusión

Muchas de las herramientas aquí descritas facilitan la generación de contenido semántico, sin embargo, muchas de ellas solo realizan una transformación sintáctica del contenido, guiada por el esquema lógico de la representación origen del contenido. Las herramientas en las que la transformación está guiada por el dominio, utilizando una ontología existente en el proceso de definición de correspondencias, consiguen una representación semántica del contenido origen, sin embargo, tienen problemas de complejidad en la definición de las correspondencias. Una buena aproximación a este problema es la estrategia que sigue Populous, haciendo uso de patrones para asistir en la recogida de conocimiento y modificación de una ontología, aunque en este caso, tiene el problema de ser poco flexible respecto a las fuentes de entrada que acepta, que deben tener un estructura tubular, lo que limita la posible complejidad de las fuentes de entrada.

Existe por tanto, una necesidad de métodos de transformación guiados por el conocimiento del dominio y que sean flexibles y genéricos en su aplicación a diferentes formatos y contenidos de entrada.

Capítulo 5

Integración de información biomédica

La investigación en biomedicina evoluciona rápidamente y genera de forma intensiva datos que requieren ser revisados, anotados y explotados. A su vez, estas investigaciones dependen de la disponibilidad y uso eficiente de la información. Sin embargo, mientras que la información se distribuye entre distintos sistemas, las distintas actividades de investigación y estudio pueden requerir el acceso unificado a distintos repositorios lo que lleva a la necesidad de creación de metodologías de integración [202; 203]. Por ejemplo, una metodología de integración puede estar motivada por la necesidad de integrar información heterogénea semánticamente equivalente, como crear un repositorio virtual que contenga todos los datos sobre el cáncer de mama recogidos por varias instituciones (integración vertical), o integrar información heterogénea semánticamente complementaria, como permitir la consulta de datos de información genómica y clínica para la medicina genómica (integración horizontal).

Se define integración de datos como el proceso de combinar datos provenientes de diferentes fuentes, autónomas y heterogéneas, y proporcionar al cliente un modelo global, unificado y reconciliado de los datos [204].

Las metodologías de integración de datos pertenecientes a información biomédica deben enfrentarse con varios problemas [179; 205; 206; 207]: los repositorios biomédicos pueden tener un ámbito muy variado, que abarca un dominio muy amplio; los distintos recursos suelen operar de forma autónoma, por lo que cualquier modificación o eliminación de datos se hace sin previo aviso; siguen sus propios esquemas de representación y sus propias terminologías, es común la utilización de distintos términos para denominar a entidades semánticamente equivalentes, mientras que en otros casos se uti-

liza un mismo término para denominar a entidades diferentes. Para realizar una integración completa y estricta, una metodología de integración debe proporcionar un modelo de datos integrado, que represente el dominio de las fuentes de datos a integrar y proporcione un acceso unificado a los datos, poniendo atención a la consistencia de los mismos. Así, en [207] se resumen los pasos de una metodología de integración en: definición de un modelo común lo bastante expresivo para representar los modelos de datos de cada fuente; alinear los esquemas semánticamente, de manera que se resuelvan los conflictos de nombrado antes de realizar la integración; integrar los esquemas en el esquema común; transformar los datos a la representación definida por el modelo común y definir las correspondencias semánticas entre los datos para garantizar la consistencia. Dependiendo de la metodología final de integración, alguno de estos pasos puede no ser necesario o modificarse, algunas metodologías realizan una integración virtual en lugar de instanciación, con lo que realizan transformación de consultas en lugar de transformación de datos, y otras metodologías se basan en sistemas muy poco federados, donde el modelo común es una simple unión de todos los esquemas fuente.

5.1 Propuestas de integración

Las propuestas de integración utilizadas habitualmente en sistemas existentes pueden ser clasificadas en términos del modelo de datos que utilizan: texto, datos estructurados o registros enlazados. Los sistemas de integración que proporcionan datos textuales incluyen buscadores de texto y palabras clave sobre los repositorios fuente. Si el sistema maneja datos estructurados, las propuestas se dividen entre las que almacenan los datos en un repositorio único y las que recuperan los datos de los repositorios fuente bajo demanda. La última propuesta considera los datos como registros enlazables y navegables, por lo que implican dar soporte a la navegación entre las fuentes. En el ámbito biomédico, la mayoría de sistemas utilizan aquellas propuestas basadas en datos estructurados o en registros enlazados [179; 206].

5.1.1 Integración basada en almacén de datos

La integración basada en almacén de datos [202] guarda la información proveniente de distintas fuentes en un único repositorio que sigue un esquema global, unificado, que modela toda la información de origen. Las acciones de acceso y consulta se hacen en el almacén de datos en lugar de sobre las fuentes de origen de la información.

Esta metodología extrae los datos de las fuentes de origen, los transforma

y los carga en el almacén de datos. Estas acciones requieren la definición de correspondencias entre las fuentes de datos y el esquema global.

Integrar toda la información en un único almacén tiene la ventaja de mejorar la eficiencia de las consultas, pues ya no es necesario acceder a varios sistemas para recuperar toda la información. Además, el preprocesado que se hace a la información antes de importarla al almacén permite filtrar, validar, modificar y anotar los datos obtenidos desde las fuentes, lo que puede ser útil para una explotación posterior [179; 208]. Por el contrario, estos sistemas tienen problemas de mantenimiento, volúmenes de datos demasiado grandes pueden ser inmanejables para el almacén, y resulta costoso mantenerlos actualizados, pues deben comprobar las fuentes de datos periódicamente en busca de modificaciones y aplicar los cambios en el almacén principal.

En el domino biomédico se están generando datos continuamente, por lo que la acción de importación de datos al almacén debe hacerse de forma continua, lo que hace muy costoso el mantener el almacén actualizado [206]. Por otro lado, el dominio de la biomedicina es muy cambiante, continuamente se están añadiendo campos y cambiando nomenclaturas. Estos cambios repercuten en el modelo global unificado en el que se basa el almacén de datos ya que este se crea en base a las fuentes de origen y cambiar el modelo global unificado implica cambiar los procesos que recuperan los datos de las fuentes de información, que los transforman y que los insertan en el almacén de datos, lo que aporta complejidad al proceso.

Debido a los problemas que conllevan, en dominios biomédicos los almacenes de datos son más adecuados para la creación de bases de datos altamente curadas en un área de investigación específica [203]. DataFoundry Project [209] y The Enterprise Data Trust en Mayo Clinic [210] son ejemplos de almacenes de datos en dominios biomédicos.

5.1.2 Integración basada en mediadores

Los sistemas basados en mediadores, también conocidos como sistemas basados en vistas, se caracterizan por mantener la información en los repositorios fuente [206]. Un esquema global proporciona una vista integrada, unificada y virtual de las fuentes de datos. Estos sistemas contienen un elemento mediador que se ocupa de reformular las consultas realizadas al esquema global a consultas sobre las fuentes de datos subyacentes.

Al contrario que en el caso de el almacén de datos, la información no es transformada e importada, sino que permanece en sus repositorios de origen y lo que se transforma son las consultas. Para la transformación, estos sistemas requieren definir las correspondencias entre las fuentes y el esquema global,

para lo que se proponen dos enfoques [211]. El primer enfoque, global-as-view (GAV), requiere que el esquema global se exprese en términos de las fuentes de datos. El segundo enfoque, local-as-view (LAV), requiere que el sistema global se especifique de forma independiente a las fuentes, y las correspondencias entre el esquema global y las fuentes se establecen definiendo cada fuente como una vista sobre el esquema global.

Ambos enfoques tiene sus ventajas y desventajas. El enfoque LAV favorece la extensibilidad del sistema, añadir una nueva fuente únicamente implica definir nuevas correspondencias entre dicha fuente y el esquema global, sin necesidad de otros cambios. Por otro lado, el enfoque GAV solo resulta efectivo en sistemas con un conjunto de fuentes estable, pues añadir una nueva fuente puede provocar cambios en la definición del esquema global. Si nos fijamos en el procesado de las consultas, este es más complejo en el enfoque LAV, pues el único conocimiento que se tiene sobre los datos es a través de las vistas representando las fuentes, y dichas vistas proporcionan solo información parcial sobre los datos. Como las correspondencias asocian a cada fuente una vista sobre el esquema global, no se puede inferir inmediatamente cómo usar las fuentes para responder consultas expresadas sobre el esquema global. Sin embargo, en el enfoque GAV, procesar las consultas es más fácil, las correspondencias especifican directamente que consultas en la fuente corresponden directamente con elementos del esquema global.

Muchos de los sistemas de integración bioinformática fueron desarrollados antes de la aparición de los sistemas basados en mediadores, y en su lugar siguieron un modelo de base de datos federado [179]. Un sistema de integración federado consiste en fuentes subyacentes que son componentes autónomos pero que cooperan para permitir acceso controlado a todos sus datos. Los datos permanecen en las fuentes de origen y el sistema da acceso a ellos desde una interfaz común, pero no existe un esquema global que describa todos los datos, sino que los esquemas de las distintas fuentes se unen [212]. Desde este punto de vista, un sistema basado en mediadores podría ser visto como una versión poco acoplada de sistema federado.

El sistema de integración de datos BIRN (Biomedical Informatics Research Network) [213] se basa en un mediador GAV para proporcionar un framework de integración de información biomédica.

5.1.3 Integración basada en enlaces

Los sistemas de integración basados en enlaces se aprovechan del hecho de que muchos recursos están disponibles en la web y un usuario debe navegar manualmente a través de varias páginas para resolver sus consultas [179]. Esta

metodología forma un grafo en el que las entidades de las distintas fuentes se conectan mediante rutas, formadas por un conjunto de enlaces entre las fuentes, de manera que la salida de una fuente redirige a otra hasta alcanzar la información requerida. Son los usuarios los responsables de navegar entre las fuentes siguiendo los distintos enlaces. Estos sistemas de integración son muy vulnerables a las ambigüedades de nombrado (como el uso de un mismo término para entidades diferentes) y a las actualizaciones en los repositorios, que pueden llevar a que un enlace en un recurso deje de ser válido [206].

SRS [214] es un sistema de integración basado en enlaces, que proporciona una interfaz para navegar un conjunto de fuentes de datos. SRS reconoce campos estructurados en los repositorios fuente y permite que los administradores enlacen campos entre diferentes bases de datos. Los usuarios pueden realizar consultas que les llevan a fuentes de datos enlazadas con otras a través de enlaces web.

5.2 Tecnologías de Web Semántica en integración

Las tecnologías de la Web Semántica ofrecen un espacio tecnológico apropiado para la integración y explotación de información biomédica [215]. En este espacio tecnológico, las ontologías son una pieza clave para dar soporte a la integración permitiendo eliminar la heterogeneidad de la información biomédica a varios niveles. Por un lado, su uso como vocabulario controlado en un dominio permite unificar términos en las distintas fuentes, proporcionando la estandarización y el vocabulario común necesario para integrarlas. Por otro lado, es muy común su utilización como esquema global y para definir las correspondencias entre este esquema global y los esquemas locales de las fuentes que permitan el acceso a las mismas a través de la transformación de datos o de consultas. En las arquitecturas de integración que utilizan las ontologías como esquema integrador, éstas se utilizan comúnmente de tres formas distintas [216]:

- El enfoque de ontología única usa una ontología global para especificar la semántica del dominio utilizando un único vocabulario común. En este enfoque, se definen correspondencias entre los esquemas de representación propios de los recursos de información fuente y la ontología global. La desventaja de este enfoque es la de no tolerar bien los cambios en las fuentes de información, que pueden implicar cambios complejos en la ontología global y, por lo tanto, en la correspondencias definidas.

- El enfoque de ontología múltiple define una ontología propia para cada fuente de información. Al no haber una ontología común, para poder tener un acceso uniforme a las fuentes, se requiere la existencia de correspondencias entre las distintas ontologías, que definen los términos de las mismas que son semánticamente equivalentes o iguales. Aunque este enfoque simplifica los cambios necesarios al añadir o cambiar una de las fuentes de información, la definición de las correspondencias entre ontologías puede ser muy compleja para fuentes muy heterogéneas.

- En el enfoque híbrido se define una ontología para definir la semántica de cada una de las fuentes de información, pero estas se construyen según una ontología común que proporciona un vocabulario compartido. El vocabulario compartido contiene términos básicos del dominio, que se combinan con ciertos operadores para definir términos más complejos. Como cada término en las ontologías individuales se define según el vocabulario común, es mucho más fácil comparar las ontologías entre sí, y añadir nuevas fuentes de información no requiere complejas modificaciones de la ontología común o en las correspondencias definidas. Como desventaja, las ontologías existentes no puede ser reutilizadas fácilmente, pues tiene que ser definidas según la ontología común.

El enfoque de ontología única se adecúa a sistemas de almacenamiento basados en almacén de datos y sistemas basados en mediador con enfoque GAV, mientras que el enfoque híbrido es apropiado para sistemas basados en mediador con un enfoque LAV.

5.2.1 Alineamiento de ontologías

El uso de ontologías en propuestas de integración de datos, requiere en muchos casos la aplicación de técnicas de alineamiento ontológico, por ejemplo, en los métodos que utilizan el enfoque híbrido o el enfoque de ontología múltiple. Se define el alineamiento ontológico como la tarea de establecer una colección de relaciones binarias entre los vocabularios de dos ontologías [217]. En esencia, un alineamiento es un conjunto de correspondencias, donde cada correspondencia define la relación entre dos entidades de dos ontologías diferentes. Con el amplio uso de las ontologías en el ámbito biomédico, el alineamiento de ontologías no solo se presenta como una solución a la integración de información, que incluye la integración de datos, esquemas y catálogos, sino como necesaria en la interoperabilidad semántica, unión de ontologías y resolución de consultas. En [218] presentan una clasificación de las técnicas de alineamiento atendiendo al nivel en el que se realiza el alineamiento:

- Técnicas a nivel de elemento: incluyen diferentes técnicas que realizan alineamientos atendiendo a atributos y propiedades de los elementos de las ontologías. Incluyen: técnicas basadas en texto, que alinean elementos utilizando sus descripciones textuales asociadas, que son tratadas como secuencias alfabéticas de palabras; técnicas de procesamiento de lenguaje natural, que estudian las propiedades morfológicas del texto asociado a los elementos; técnicas que estudian las restricciones asociados a los elementos, como la cardinalidad; técnicas que estudian los recursos externos asociados con los elementos de la ontología, bajo la premisa de que dos elementos asociados al mismo recurso pueden tener correspondencias; y técnicas que hacen uso de recursos externos formales, como ontologías externas, de alto nivel o específicas del dominio, que aportan una semántica compartida que facilita el alineamiento de los elementos.

- Técnicas a nivel de estructura: comparan las entidades o instancias de la ontología utilizando sus relaciones con otros entidades o instancias. Entre las propuestas aquí clasificadas se incluyen las técnicas que consideran las ontologías a ser alineadas como grafos etiquetados; técnicas que solo toman en consideración la relación de especialización; técnicas que utilizan la interpretación semántica de las ontologías a alinear; técnicas que comparan los conjuntos de instancias de dos clases, para comprobar si dichas clases pueden corresponder.

Los lenguajes para expresar alineamientos son variados, es común la utilización de lenguajes propios de la Web Semántica, como OWL, que puede ser utilizado para expresar equivalencia entre dos conceptos, o lenguajes de reglas, como SWRL (SemanticWeb Rule Language) [219]. También se utilizan lenguajes propios, como EDOAL [220], utilizados en la Alignment API [221] en sus últimas versiones, que surgió por la necesidad de tener un lenguaje más expresivo para definir correspondencias más precisas. EDOAL incluye restricciones y transformaciones, lo que permite generar alineamientos más expresivos y hacer una gestión más complejas de las entidades a alinear [220].

Los alineamientos en EDOAL están formados por un conjuntos de celdas, donde cada celda define una asociación entre dos entidades. Las entidades que una celda asocia pueden ser descripciones de entidades compuestas y se categorizan en clases, “Class”, instancias “Instance”, relaciones (owl:objectProperty) “Relation” y propiedades (owl:datatypeProperty) “Property”. Las entidades pueden ser restringidas, de manera que una celda asocie una clase con unas cardinalidades o valores específicos para sus propiedades y relaciones, y se permite aplicar transformaciones a las entidades de la catego-

ría instancias, como por ejemplo transformaciones de unidad o concatenación de dos cadenas de texto.

5.2.2 Ejemplos de integración

Antes de la aparición de Linked Open Data (LOD), el uso más común de las tecnologías de la Web Semántica para la integración de datos era la construcción de almacenes de datos semánticos y sistemas mediadores utilizando una ontología como esquema global [222]. Un ejemplo es YeastHub [223], un almacén de datos RDF que permite la integración de diferentes tipos de datos genómicos de la levadura, proporcionados por distintos recursos en distintos formatos. El proceso de integración de este almacén de datos sigue los siguientes pasos: (1) descargar los contenidos de la web de cada recurso; (2) convertir los contenidos a RDF, aquellos en formato delimitado por tabulaciones, con una transformación guiada paso a paso; aquellos en bases de datos relacionales, utilizando D2RQ; (3) almacenar los datos transformados en un almacén de datos RDF. Cuando los datos se cargan en el almacén de datos, las consultas basadas en RDF pueden realizarse para recuperar y consultar los datos de forma integrada.

Con la aparición de los principios de LOD, comenzó la publicación de conjuntos de datos biológicos abiertos en la web siguiendo los principios de Linked Data. Bio2RDF [25] es un ejemplo de conjunto de recursos abiertos publicados según Linked Data, que forman una red federada de puntos de acceso. Bio2RDF usa documentos RDF y una lista de reglas para crear URI que crearán datos enlazados. Convierte los documentos de la web a formato RDF, para ello crea una descripción OWL para cada página HTML y genera la ontología global uniendo todas las ontologías generadas. Creada la ontología, Bio2RDF crea las herramientas que transformarán cada uno de los recursos a RDF, las cuales definirán las correspondencias entre los datos del documento original y los elementos RDF, y normalizarán las URI de cada recurso siguiendo la sintaxis Bio2RDF. Bio2RDF normaliza las URI utilizando el mismo patrón, para que los mismos elementos provenientes de distintas fuentes, se generen con la misma URI y, por lo tanto, los enlaces RDF entre entidades queden generados automáticamente.

El sistema OGO [224] es un ejemplo de repositorio integrado de información biomédica de genes y proteínas ortólogos y sus enfermedades relacionadas. La información proviene de distintos recursos heterogéneos (KOG, Inparanoid, OrthoMCL, Homologene y OMIM) y se almacena en una base de conocimiento que toma como modelo una ontología global que define todo el conocimiento de ortólogos y enfermedades genéticas. La integración de la in-

formación siguiendo la semántica de la ontología permite relacionar los genes implicados en una enfermedad genética con sus grupos de ortólogos. La metodología de integración sigue la transformación de las fuentes heterogéneas siguiendo el método de transformación del sistema OGO [201] comentado en el capítulo anterior, donde las reglas de correspondencia definen cómo se transformar los recursos relacionales de entrada a la representación guiada por una ontología global y las reglas de identidad se utilizan para detectar los individuos que tienen distinta URI pero representan la misma entidad.

En [205] se propone una metodología de integración de datos biológicos haciendo uso de tecnologías de la Web Semántica. En su sistema utilizan la propuesta de almacén de datos para construir un repositorio central con todos los datos agregados. La información a integrar está formada por datos sobre genes y sus productos, originalmente en formatos RDF, OWL, tabular y bases de datos relacionales. Las nuevas descripciones RDF utilizadas en el proceso de transformación toman su vocabulario de una nueva ontología definida, llamada Biowl. Además de esta ontología propia, se utiliza GO y la ontología core.owl definida por UniProt. Estas tres ontologías se unifican en una, por medio de la definición de equivalencias entre sus clases y propiedades de forma manual y con herramientas externas. Para identificar los distintos recursos relacionados entre sí, se utiliza la información de relaciones entre recursos disponible en bases de datos como Ensembl, KEGG y NCBI y se definen correspondencias a mano. El resultado final es una base de conocimiento que puede ser consultada con SPARQL.

El proyecto Semantic Enrichment of the Scientific Literature (SESL) [225] se enfoca en la integración y compartición de información sobre diabetes mellitus tipo 2 (T2DM) en adultos. En el proyecto se integra literatura científica con los recursos biomédicos UniProt Knowledgebase (UniProtKB) [76], Gene Expression Atlas [226] y OMIM [169]. El proyecto utilizó un total de 20.168 publicaciones sobre T2DM, a los cuales se les aplican procesos de minería de texto para identificar oraciones y bloques de texto y se anotan con terminologías estandar sobre enfermedades y proteínas. En concreto, los nombres de genes y proteínas se identifican utilizando LexEBI [227], mientras que la identificación de enfermedades se realiza utilizando terminologías de UMLS [82]. Todas las oraciones que contienen un par que incluye un gen y una enfermedad se identifican e integran en un repositorio de tripletas de SESL. El repositorio UniProtKB, cuyo contenido está disponible como repositorio de tripletas, se procesa y reduce para obtener sólo el contenido sobre proteínas humanas. Los datos provenientes de Gene Expressión Atlas y las enfermedades importadas desde OMIM se normalizan utilizando UMLS, Gene Expressión Atlas está anotado con Experimental Factor Ontology [228], por lo

que las anotaciones se normalizan con anotaciones de Disease Ontology [229] para usar las correspondencias de UMLS.

Los sistemas de integración de recursos heterogéneos deben ser capaces de identificar las instancias que representan a una misma entidad y conectarlas entre sí, así como normalizar los vocabularios utilizados en el nombrado de las entidades. Los sistemas ejemplo aquí presentados identifican unívocamente a los individuos a través de la normalización de las URI y realizan una identificación de las equivalencias de forma manual. El sistema OGO se basa en la definición de reglas sobre la ontología global que modela el dominio para definir los requisitos de identidad. Para la normalización de vocabularios, los sistemas seleccionan un vocabulario común dado por una terminología seleccionada, y realizan la normalización de los datos de entrada a dicha terminología.

5.3 Discusión

Las arquitecturas de integración más comunes se pueden dividir entre aquellas que realizan una integración física de los repositorios fuente, mediante extracción de datos e integración en un repositorio común (almacenes de datos), y aquellas que realizan una integración virtual, dónde las consultas se realizan sobre un modelo común pero los datos se mantienen en sus repositorios de origen (sistemas basados en mediadores). El tercer tipo de sistemas más común aprovecha la existencia de recursos disponibles en la web para definir una integración basada en enlaces entre los datos.

Con la irrupción de las tecnologías de la Web Semántica se ha generalizado el uso de las ontologías en los sistemas de integración, siendo muy común su utilización para modelar el esquema global de integración de distintos recursos. Estos hace que técnicas como el alineamiento de ontologías también cobren importancia en los sistemas de integración.

La aparición de los principios de Linked Open Data (LOD) ha fomentado la publicación de recursos en formato semántico que siguen los principios de LOD. En general, para la publicación de los recursos se crean soluciones propias a los recursos a integrar. Los procesos de transformación, normalización e integración son propios y dedicados a cada fuente de datos y dependen en gran parte de la intervención manual para resolver los conflictos de modelado. Existe pues, una carencia de soluciones genéricas de integración, aplicables a distintas fuentes de datos y distintos formatos de representación e independientes del dominio de aplicación.

Capítulo 6

Objetivos

6.1 Motivación

La medicina traslacional requiere la explotación integrada de información biomédica para dar soporte a la investigación, sin embargo, la generación continuada de datos biomédicos por distintas instituciones, y su representación y gestión utilizando sistemas propios, lleva a la situación de tener el conocimiento distribuido y representado de forma heterogénea.

Varias propuestas han surgido para mejorar esta situación. Los estándares de HCE y las terminologías clínicas surgen con el objetivo de normalizar la información clínica de forma que pueda ser intercambiada y entendida por los distintos sistemas. Estándares y especificaciones de HCE como ISO 13606, openEHR, HL7 o CEM, basados en una arquitectura dual, surgen con el propósito de facilitar la interoperabilidad semántica de la información clínica. Además, la información biomédica es anotada y codificada utilizando conceptos procedentes de terminologías y ontologías estandarizadas. Sin embargo, la variedad de estándares y terminologías utilizados por las distintas instituciones perjudica la consecución de la interoperabilidad semántica, por lo que se hace necesario la utilización de técnicas adicionales para lograr la explotación conjunta de la información.

Entre las propuestas para dar solución a estos problemas, el uso de las tecnologías de la Web Semántica para la representación, gestión y compartición de la información biomédica es muy común. Las ontologías biomédicas, como Gene Ontology, son ampliamente utilizadas en la anotación de información biológica, y han aparecido iniciativas como OBO Foundry, para la estandarización del diseño de ontologías biomédicas, y BioPortal, como repositorio de gestión de las mismas. En el ámbito clínico, algunas iniciativas para la representación de modelos clínicos utilizando OWL han demostrado la uti-

lidad de este lenguaje para la gestión y explotación del conocimiento, pues permite realizar actividades semánticas como el uso de razonamiento para la obtención de conocimiento. Por otro lado, iniciativas como Linked Open Data proponen la publicación de datos en la web bajo unas condiciones que facilitan su consulta, explotación y combinación con otras fuentes.

Muchas propuestas ofrecen métodos para la transformación de información a una representación semántica basada en lenguajes como OWL o RDFS, al igual que existen muchos proyectos de creación de sistemas integrados de información biomédica para dar soporte a estudios e investigaciones científicas. Sin embargo, estas soluciones presentan algunas limitaciones. Se trata de metodologías propias, orientadas a formatos de representación y recursos concretos, y por lo tanto, poco flexibles a la hora de adaptarse a distintos tipos de recursos y problemas. Por otro lado, las herramientas de gestión de modelos clínicos existentes no aprovechan las ventajas de las tecnologías semánticas y no permiten hacer una gestión integrada de modelos clínicos, terminologías y datos.

En este trabajo se proponen soluciones para la explotación integrada de información biomédica haciendo uso de tecnologías de la Web Semántica. La solución propuesta se basa en (1) un modelo genérico de transformación, basado en la definición de reglas de transformación entre esquemas de representación de contenido, (2) un modelo de integración basado en el modelo de transformación y una arquitectura ontológica basada en ontologías y patrones de diseño ontológico de contenido, (3) una plataforma de integración y explotación de información biomédica, que permita realizar una gestión controlada basada en explotación de la representación semántica de datos y modelos clínicos.

6.2 Objetivos

El objetivo principal de esta tesis es la investigación y desarrollo de soluciones basadas en las tecnologías de la Web Semántica para la integración y estandarización de conocimiento biomédico utilizado en medicina traslacional. Para conseguir este objetivo se definen las siguientes tareas:

- Diseño e implementación de un modelo de transformación genérica de datos entre esquemas de representación estructurados.
- Diseño e implementación de un modelo de integración de información biomédica heterogénea.
- Diseño e implementación de una plataforma de integración, gestión y

explotación de información biomédica, que permita el acceso integrado a información biomédica procedente de las HCEs y recursos externos, y que haga uso de tecnologías de la Web Semántica para la explotación del conocimiento.

- Aplicación y validación de los resultados obtenidos por medio de la transformación e integración de recursos biomédicos heterogéneos, su integración en la plataforma de gestión de información biomédica y su explotación en actividades de uso secundario de información biomédica.

6.3 Hipótesis

La hipótesis principal de esta tesis es que mediante el uso de tecnologías de la Web Semántica se puede generalizar la integración de información biomédica proveniente de recursos heterogéneos y facilitar la gestión de modelos y datos clínicos. Esta hipótesis se divide en las siguientes sub-hipótesis:

- **Es posible la definición de un método de transformación de información biomédica guiado por el dominio de salida a través del uso de reglas de transformación y patrones de diseño.** Comprobar esta hipótesis requiere contestar a las siguiente preguntas:

  1. ¿Qué representaciones son las más comunes en los sistemas de información biomédica?
  2. ¿Cuáles son los métodos de transformación de recursos a representación semántica y qué problemas tienen asociados?
  3. ¿Qué componentes definen un modelo de transformación genérico?
  4. ¿Qué ventajas y facilidades traen el uso de reglas de transformación y patrones de diseño en el modelo de transformación?

- **La aplicación del modelo de transformación genérico para la transformación a una representación OWL permite definir un proceso de integración genérico para información proveniente de fuentes heterogéneas.** Comprobar esta hipótesis requiere contestar a las siguiente preguntas:

  5. ¿Qué técnicas de integración de recursos heterogéneos existen y cuáles son sus problemas asociados?
  6. ¿Cómo se pueden generalizar los procesos de integración para que sean aplicados a cualquier recurso de información?

7. ¿Cómo mejoran las tecnologías de la Web Semántica la integración de recursos heterogéneos?

- **Mediante la aplicación del proceso de integración definido y métodos basados en tecnologías de la Web Semántica se facilita la explotación integrada del conocimiento incluido en los recursos biomédicos y el uso secundario de la información.** Comprobar esta hipótesis requiere contestar a las siguiente preguntas:

8. ¿Qué tareas son clave en la gestión, explotación y uso secundario de la información biomédica?
9. ¿Cómo el uso de representación semántica basada en ontologías OWL facilita las tareas de gestión de información biomédica y su uso secundario?

6.4 Metodología

La metodología a seguir se basa en el estudio del estado del arte, la formalización de los métodos propuestos en esta tesis, su implementación y su validación en un dominio de aplicación.

- Estudio del estado del arte:
  - Información biomédica: estudio de los formatos de representación más comunes de la información biomédica. Esto engloba estándares y especificaciones para la HCE, poniendo mayor interés en aquellos basados en arquitectura de dos niveles, es decir, ISO 13606, openEHR, HL7 y CEM; representación y gestión de modelos clínicos (arquetipos); representación más común de bases de datos biológicas; y terminologías biomédicas.
  - Web Semántica: estudio de las tecnologías de la Web Semántica, RDF, lenguaje OWL, propuestas de Linked Data para la publicación de datos y técnicas de ingeniería ontológica para la creación reutilización de ontologías. Además del estudio de las ontologías biomédicas disponibles, su utilización más común y las propuestas existentes para su creación y gestión, así como estudio de las propuestas existentes de representación ontológica de estándares de la HCE y herramientas relacionadas.
  - Transformación de contenidos a representación semántica: análisis de las propuestas existentes para obtención de representaciones RDF/OWL de datos almacenados en repositorios no semánticos.

  - Integración de información: estudio de las propuestas existentes para la integración de repositorios heterogéneos, así como la aplicación de ontologías en este ámbito.

- Formalización de la propuesta:

  - Desarrollo de una metodología de transformación genérica. La transformación está guiada por la definición de reglas de transformación entre esquemas de entrada y salida y la utilización de patrones de diseño ontológico de contenido.
  - Desarrollo de un proceso de integración de recursos heterogéneos. Por medio de la instanciación de la metodología de transformación, se integran distintos recursos heterogéneos seleccionados para un dominio biomédico. El modelo de salida se define por un arquitectura ontológica formada por una ontología global y patrones de diseño ontológico de contenido.
  - La selección de representaciones semánticas adecuadas de modelos clínicos pertenecientes a estándares de HCE y de métodos de anotación, comparación y validación adecuados.

- Implementación de la propuesta por medio de herramientas para la transformación, integración, gestión y explotación de información biomédica. Las herramientas implementadas agrupan los métodos formalizados y crean una plataforma de integración, gestión y explotación de información biomédica que contiene un repositorio integrado de modelos y datos clínicos.

- Validación de la propuesta a través de la definición de varios escenarios de validación en dominios biomédicos. En concreto, se aplicarán los métodos y herramientas diseñadas a: estudio de los datos clínicos de pacientes del programa de cribado de cáncer de colon y recto para la clasificación automática de pacientes; transformación entre modelos clínicos CEM y arquetipos openEHR; creación de un repositorio integrado sobre genes ortólogos, enfermedades genéticas e información sobre anotación de secuencias genómicas; transformación de bases de datos de componentes químicos a representación OWL.

Bloque II

Resultados

Capítulo 7

Modelo de transformación

La Web Semántica proporciona un entorno tecnológico en el que información proveniente de fuentes diferentes, con distintas representaciones y uso de distintos vocabularios puede ser armonizada. Representar información por medio de tecnologías de la Web Semántica facilita el uso y acceso a la misma. La representación siguiendo los principios de Linked Data permite publicar la información y enlazarla con información relacionada, mientras que la representación en lenguajes como OWL permite utilizar las características de razonamiento para obtener conocimiento adicional y potencia el uso secundario de la información.

En dominios como el biomédico, donde la facilidad de acceso a la información y la necesidad de su explotación en investigación son claves, la representación de la información biomédica utilizando tecnologías de la Web Semántica es cada vez más común. La información biomédica tiene la característica de ser heterogénea, pues la información está generada por distintas instituciones, utiliza distintos estándares y diferentes herramientas para generar los datos, por lo que se hace importante encontrar un formalismo común de representación para la correcta explotación de la misma.

En este capítulo se presenta un modelo de transformación que recibe un recurso de información conforme a un modelo de entrada y realiza la transformación del contenido a una representación conforme a un modelo de salida.

El modelo de transformación es un modelo genérico, aplicable a cualquier repositorio de información cuyo modelo de representación de origen y cuya representación de salida buscada cumplan con las características del metamodelo de entrada y salida definido. Sin embargo, el modelo ha sido diseñado atendiendo a las características de los repositorios de información biomédica. Por norma general, este tipo de repositorios son de gran tamaño, están suje-

tos a modificaciones continuas, y son heterogéneos entre los distintos sistemas y organizaciones. El uso eficiente de la información biomédica requiere de su integración e intercambio entre las diferentes organizaciones, para dar soporte a la bioinformática y la medicina traslacional, por lo tanto, se requiere un completo entendimiento de los mismos y de una representación que permita su consulta y recuperación de forma eficiente.

7.1 Definición del modelo de transformación

El modelo de transformación se formaliza como la tripleta $\langle\langle E,I_E\rangle, R, \langle S, I_S\rangle\rangle$ donde la primera tupla representa al recurso de entrada formado por un modelo de entrada E y las instancias de información I_E definidas según el modelo E, R corresponde al conjunto de reglas de transformación y por último el modelo de salida S que junto a las instancias de información I_S forman el recurso de salida resultante de la transformación. El proceso de transformación está guiado por las reglas de transformación, que se dividen en dos: (1) las reglas de correspondencia definen la asociación entre el modelo de entrada y el modelo de salida para la extracción de datos de la fuente; (2) las reglas de identidad comprueban que las nuevas instancias creadas son únicas.

En las siguientes secciones describo los metamodelos que definen la estructura de los modelos de entrada y salida aceptados por el modelo de transformación.

7.1.1 Metamodelo de entrada y salida

Un modelo de entrada E y un modelo de salida S, para poder ser utilizados en el modelo de transformación, deben seguir el metamodelo definido por la tupla $\langle$ *Entidad*, *Atributo* , *Relación*, *Asociación*$\rangle$, donde:

- *Entidad* se refiere al conjunto de conceptos del dominio. Un concepto del dominio pertenece al conjunto *Entidad* cuando es independiente y puede ser identificado inequívocamente. Es decir, existe una función *id* para la que se cumple:

$$\forall c_i, c_j \in C, id(c_i) = id(c_j) \Rightarrow c_i = c_j \tag{7.1}$$

  Donde C es el conjunto *Entidades* del metamodelo.

- *Atributo* representa al conjunto de propiedades que caracterizan a las entidades del metamodelo.

- *Relación* es el conjunto de propiedades que se utilizan para enlazar las distintas entidades del metamodelo.

- *Asociación* se refiere a la relación de *Entidad* con el conjunto *Atributo* y *Relación*. Por medio del conjunto *Asociación* se define la estructura del metamodelo.

7.1.1.1 Modelo de entrada

Existen varios tipos de recursos de información cuyo modelo de entrada se adecúa al metamodelo propuesto. Destaco los más comunes en la representación de información biomédica, (1) las bases de datos relacionales, (2) los repositorios XML y (3) los repositorios de extractos clínicos arquetipados. Los repositorios de extractos clínicos también son repositorios XML, sin embargo, realizo la distinción porque los extractos clínicos no utilizan un XML schema como modelo, sino un arquetipo, por lo que la aplicación del modelo de transformación difiere.

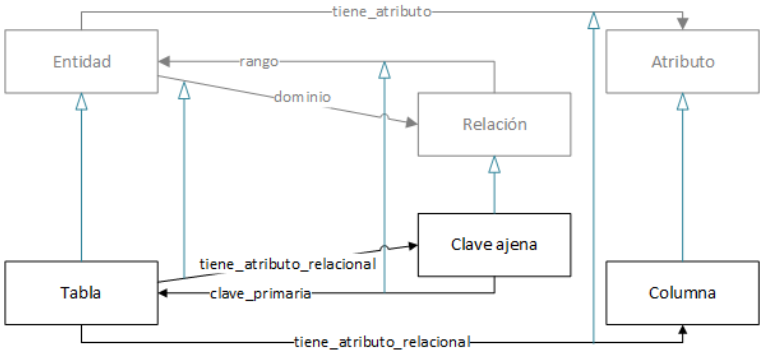

Figura 7.1: Esquema de base de datos relacional como modelo de entrada

La figura 7.1 muestra las equivalencias entre un esquema de base de datos relacional y el metamodelo de entrada. El conjunto *Entidad* viene definido por el conjunto de tablas (llamadas relaciones en el modelo entidad-relación) que define el esquema. Cada una de las columnas define un atributo para una entrada de la tabla y las relaciones entre tablas vienen definidas por las claves ajenas. Es decir, una tabla define varias columnas de propiedades (atributos) y enlaza con otras tablas a través de una propiedad de clave ajena (relaciones), que hace de clave primaria (identificador único) de otra tabla.

Los repositorios de datos XML también se adecúan a este metamodelo. Para este caso, el modelo de entrada es un XML schema. En la figura 7.2

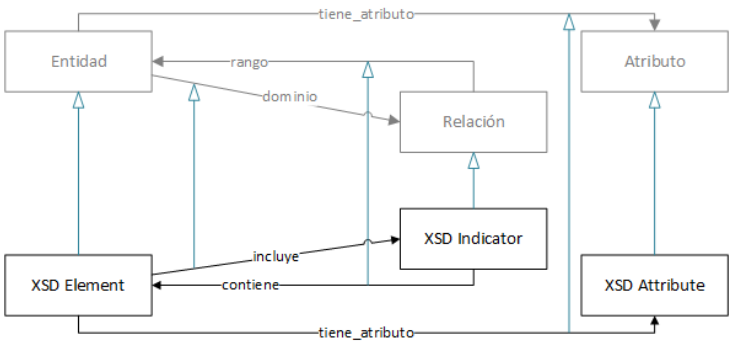

Figura 7.2: XSD schema como modelo de entrada para datos representados en XML

vemos que cada elemento *XSD Element* de un XML schema define una entidad en el modelo de entrada. Además, los elementos definen atributos por medio de *XSD Attributes*. Las relaciones entre elementos se dan a través de jerarquías y composiciones, un *XSD Element* puede ser un tipo compuesto formado por varios *XSD Element* y por medio de *XSD Indicators* se define cómo son usados esos elementos, definiendo orden y cardinalidad.

Los extractos de datos clínicos que forman parte de la HCE de un paciente se pueden representar por medio de ficheros XML, sin embargo, su estructura no se define en un XML schema, sino que se basa en la definida por un arquetipo que actúa como modelo de entrada. Los arquetipos, como parte de un estándar de modelo dual, se definen como restricciones sobre un modelo de referencia. La figura 7.3 muestra cómo las entidades en un arquetipo se definen por medio de las clases del modelo de referencia (*RM Class*) y sus atributos por medio de *Attributes* cuyo tipo de datos es simple. Las relaciones entre entidades se definen de forma jerárquica a través de atributos cuyo tipo de datos es una *RM Class*.

7.1.1.2 Modelo de salida

El modelo de transformación utiliza un modelo de salida cuyo dominio representado cumple la ecuación 7.2. D_E y D_S representan los dominios del modelo de entrada y el modelo de salida respectivamente. Ambos dominios tienen intersección no vacía, que representa el objetivo a transformar. Tanto el modelo de entrada como el de salida pueden representar dominios más amplios que no intervienen en la transformación.

$$\forall D_E, D_S, D_E \cap D_S \neq \emptyset \tag{7.2}$$

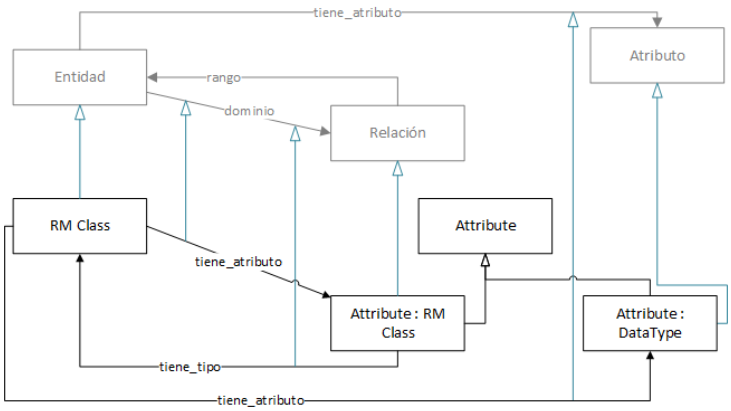

Figura 7.3: Arquetipo como modelo de entrada

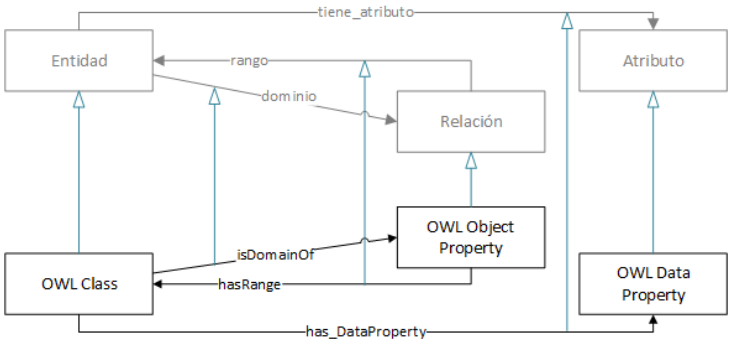

Figura 7.4: Ontología OWL como modelo de salida

Una ontología OWL sería un posible modelo de salida, que además permite obtener una representación semántica de la información. La figura 7.4 muestra un modelo de salida representado como una ontología OWL. Las entidades están formadas por un conjunto de clases `OWL Class`, los atributos se definen a través de propiedades `OWL DatatypeProperty` y las relaciones por medio de `OWL ObjectProperty`. Las definiciones de dominio y rango asocian las clases con las propiedades y relaciones.

7.1.2 Reglas de transformación

Las reglas de transformación tienen dos cometidos, controlar que el contenido de los recursos de entrada se transforme correctamente a una representación basada en el modelo de salida, y controlar la redundancia de datos en el

proceso de transformación. Se definen dos tipos de reglas, de correspondencia y de identidad.

7.1.2.1 Reglas de correspondencia

Las reglas de correspondencia se basan en la relación de congruencia. En el modelo de transformación, se define la relación de congruencia de una entidad a respecto a otra entidad b como la relación lógica que garantiza que se puede obtener a a partir de b y a será consistente respecto a su modelo de representación y respecto a b. Las reglas de correspondencia definen las relaciones de congruencia de las entidades del modelo de salida S respecto a las entidades del modelo de entrada E, es decir, una entidad C_1 del modelo de salida es congruente a una entidad C_2 del modelo de entrada, si cualquier instancia de C_1, creada a partir de información asociada a una instancia de C_2, es lógicamente consistente respecto al modelo de salida y su definición es consistente respecto a la instancia de C_2.

El conjunto de reglas de transformación R está formado por una serie de reglas que se definen como $q_E \mapsto q_S$, donde q_E es una consulta en base al modelo de entrada E y q_S una consulta en base al modelo de salida S, de manera que una regla expresa que las instancias que satisfacen la consulta q_E son congruentes con las instancias que satisfacen la consulta q_S. Si definimos q^{I_E} como el conjunto de instancias que cumplen q_E según el modelo E y q^{I_S} como el conjunto que cumplen q_S según el modelo S, se cumple:

$$q^{I_E} \subseteq I_E, q^{I_S} \subseteq I_S; \forall t_S \in q^{I_S}, \exists t_E \in q^{I_E} \mid t_S \cong t_E \tag{7.3}$$

Es decir, las reglas de transformación definen dos consultas que se traducen en la obtención de individuos de ambos modelos congruentes entre sí.

Las reglas de correspondencia son unidireccionales, es decir, dan la información suficiente para que el modelo de transformación obtenga las instancias del repositorio de entrada necesarias para crear las instancias congruentes en el modelo de salida. De esta forma, en la ecuación 7.3, $t_S \cong t_E$ significa que parte de los datos en t_E han sido utilizados para construir su congruente t_S en el modelo de salida S.

La creación en el modelo S de instancias que cumplan los criterios para pertenecer al conjunto I_S requiere definir consultas q_E y q_S de distinta complejidad para cubrir todos los componentes de una instancia. Atendiendo a la naturaleza de las consultas q_S, se definen tres tipos de reglas de correspondencia:

Reglas de clase. Son el conjunto de reglas que enlazan entidades del modelo de entrada con entidades del modelo de salida. Las consultas q_E, q_S en estas reglas crean la correspondencia entre una entidad del modelo de entrada y una entidad del modelo de salida. Una regla de clase entre una entidad C_1 del modelo de entrada y una entidad C_2 del modelo de salida se formaliza como:

$$Regla\_clase(C_1, C_2) : \forall i_1 \in C_1, \exists i_2 \in C_2 \mid i_1 \in I_E, i_2 \in I_S, i_2 \cong i_1 \quad (7.4)$$

En la ecuación 7.4 vemos que para todo individuo i_1 en el repositorio de entrada que sea instancia de la entidad C_1, se crea un nuevo individuo congruente i_2, que es instancia de la entidad C_2.

Reglas de propiedad. Las reglas de propiedad enlazan un atributo de una entidad del modelo de entrada con un atributo de una entidad del modelo de salida, de esta manera la regla permite dar valores a las propiedades de los nuevos individuos creados. Las consultas q_E, q_S en estas reglas se definen cada una a través de una entidad de los modelos de entrada y uno de sus atributos enlazados. Una regla de propiedad entre un atributo A_1 asociado a una entidad C_1 del modelo de entrada y un atributo A_2 asociado a una entidad C_2 del modelo de salida se formaliza como:

$$\begin{aligned} Regla\_propiedad((C_1, A_1), (C_2, A_2)) : \forall i_1 \in C_1, \exists i_2 \in C_2 \mid \\ value(C_1, A_1, i_1) = value(C_2, A_2, i_2) \end{aligned} \quad (7.5)$$

En la ecuación 7.5 vemos que para toda instancia de C_1 en el repositorio de entrada, que tenga un valor para el atributo A_1, existe una nueva instancia congruente i_2 de la clase C_2, con el mismo valor para una propiedad A_2 congruente con A_1.

Reglas de relación. Estas reglas enlazan una relación utilizada para asociar dos entidades del modelo de entrada con una relación que asocie dos entidades del modelo de salida, de esta manera la regla permite relacionar dos nuevos individuos creados. Las consultas q_E, q_S en estas reglas se definen cada una a través de dos entidades y una relación. Una regla de relación entre una regla R_1 que enlaza dos entidades C_{E1} y C_{E2} del modelo de entrada, y una relación R_2 que enlaza dos entidades C_{S1} y C_{S2} del modelo de salida, se formaliza como:

$$Regla\_relacion((C_{E1}, R_1, C_{E2}), (C_{S1}, R_2, C_{S2})) :$$
$$regla\_clase(C_{E1}, C_{S1}) \wedge regla\_clase(C_{E2}, C_{S2})$$
$$\wedge \{\forall i_{E1} \in C_{E1}, i_{E2} \in C_{E2}, \exists i_{S1} \in C_{S1}, i_{S2} \in C_{S2} \mid$$
$$relacion(i_{E1}, R_1, i_{E2}), relacion(i_{S1}, R_2, i_{S2})\}$$
$$\Rightarrow R_2 \cong R_1 \quad (7.6)$$

En la ecuación 7.6 vemos que si para dos entidades del modelo de entrada C_{E1} y C_{E2}, cuyos individuos se asocian a través de una relación R_1, se cumple la regla de clase, es decir, para cada individuo de C_{E1} y de C_{E2} existe un individuo congruente en el modelo de salida, entonces también existe una relación R_2 en el modelo de salida que asocie dichos individuos congruentes.

7.1.2.2 Patrones de transformación

Las reglas básicas de correspondencia anteriormente planteadas son, en algunos casos, insuficientes. Por un lado, transformar las instancias de datos a una representación de salida con una semántica precisa puede requerir información que no está explícita en los datos de entrada. Esto está relacionado con las limitaciones semánticas de algunos formalismos como XML o las bases de datos relacionales. Por ejemplo, puede ser necesario añadir una propiedad adicional a todas las instancias transformadas de una entidad del modelo de entrada, para así enriquecer su semántica. Por otro lado, puede ocurrir que una regla de correspondencia requiera crear varias instancias y sus relaciones entre sí, bajo condiciones específicas, lo cual no se puede hacer con las reglas de clase, propiedad y relación. Por ejemplo, crear una instancia a relacionada con una instancia b, sólo en el caso de que b esté relacionada con una instancia c.

Para dar una solución a este problema, se ha adaptado la aproximación de ingeniería del software e ingeniería ontológica, en las que se definen patrones para facilitar la construcción de software o de ontologías por medio de la modularización y reutilización de componentes (ver sección 3.2.3.1).

Los patrones en el modelo de transformación son plantillas sobre el modelo de salida que se definen para diferentes situaciones de modelado. Representan parte o la totalidad de la definición de una entidad en el modelo de salida, situándose entre el modelo de entrada y el modelo de salida a la hora de definir las correspondencias entre ambos y creando una vista que aísla al usuario de la complejidad del modelo de salida. Los patrones permiten definir correspondencias más complejas de manera más simple. Son plantillas cuya

aplicación a un conjunto de datos genera instancias en el modelo de salida. Se diseñan utilizando un conjunto prefijado de entidades, atributos, relaciones y asociaciones del modelo de salida, que junto a una serie de variables que parametrizan instancias de entidades y valores de atributos, crean plantillas de reglas.

Un patrón se formaliza como la tupla $\langle\ S', V, P_R\ \rangle$, donde se cumple:

- $S' \subset S$, es decir, todas las entidades, atributos, relaciones, instancias y asociaciones definidas en el patrón en S' son un subconjunto de las que componen el modelo de salida S.

- V es el conjunto de variables que parametrizan la instancia de una entidad o el valor de un atributo del conjunto S'.

- P_R es el conjunto de plantillas de reglas. Un patrón se define a partir de las asociaciones entre los elementos del conjunto S' y las variables del conjunto V. Aquellas asociaciones en las que estén implicadas las variables generan una serie de plantillas de reglas para definir reglas de correspondencia entre elementos del modelo de entrada y las variables del patrón. Las plantillas permiten que para cualquier tipo de regla sólo se tengan que especificar los elementos del modelo de entrada y las variables del patrón. El resto de información, como los elementos del modelo de salida involucrados, es aportado por la plantilla.

Las ecuaciones 7.7, 7.8 y 7.9 muestran las plantillas tipo para una regla de clase, una regla de propiedad y una regla de correspondencia que forman parte de la definición de un patrón.

Dada una entidad C_1 del modelo de entrada y una variable V_2 que parametriza las instancias de la entidad $C_2 \in S'$, congruente a C_1, se cumple:

$$Plantilla\_regla\_clase(C_1, V_2)\{ \\ V_2 \in C_2 \wedge \forall i_1 \in C_1 \exists i_2 \subseteq V_2 \mid i_2 \cong i_1\} \quad (7.7)$$

Es decir, la plantilla define una correspondencia de clase entre C_1 y C_2. Para ello sólo requiere la especificación de la entidad del modelo de entrada (C_1) y la variable del patrón (V_2). De la información del patrón obtiene que V_2 parametriza las instancias de C_2. Como resultado, por cada instancia i_1 de C_1 en el repositorio de entrada, se creará una instancia i_2 de C_2 en el repositorio de salida, que será congruente con i_2 y seguirá la definición de V en el patrón.

Dada una entidad C_1 y un atributo A_1 del modelo de entrada y una variable V que parametriza los valores de la entidad C_2 congruente a C_1 para el atributo A_2 congruente a A_1, se cumple:

$$\begin{aligned} &Plantilla\_regla\_propiedad((C_1, A_1), (V_2, V3))\{ \\ &\quad V_2 \in C_2 \wedge \forall i_1 \in C_1, \exists i_2 \subseteq V_2 \mid value(C_2, A_2, i_2) \subseteq V_3 \wedge \\ &\quad value(C_1, A_1, i_1) = value(C_2, A_2, i_2)\} \end{aligned} \quad (7.8)$$

Es decir, la plantilla define una correspondencia de propiedades entre (C_1, A_1) y (C_2, A_2). Para ello sólo requiere la especificación de la entidad y el atributo del modelo de entrada (C_1, A_1) y la variable del patrón (V_2). De la información del patrón se obtiene que V_2 parametriza los valores de las instancias de C_2 para el atributo A_2.

Dadas las entidades C_1, C_2 y su relación R_1 del modelo de entrada y las variables V_3 y V_4 que parametrizan las instancias de las entidades C_3 y C_4, pertenecientes a S' y congruentes a C_1 y C_2 respectivamente, se cumple:

$$\begin{aligned} &Plantilla\_regla\_relacion((C_1, R_1, C_2), (V_3, V_4))\{ \\ &\quad V_3 \in C_3 \wedge V_4 \in C_4 \wedge \\ &\quad Plantilla\_regla\_clase(C_1, V_3) \wedge \\ &\quad Plantilla\_regla\_clase(C_2, V_4) \wedge \\ &\quad \{\forall i_1 \in C_1, i_2 \in C_2, \exists i_3 \subseteq V_3, i_4 \subseteq V_4 \mid \\ &\quad relacion(i_1, R_1, i_2) \wedge relacion(i_3, R_2, i_4)\} \\ &\quad \Rightarrow R_2 \cong R_1\} \end{aligned} \quad (7.9)$$

Es decir, la plantilla define una correspondencia de relación entre (C_1, R_1, C_2) y (C_3, R_2, C_4). Para ello sólo requiere la especificación de las entidades y relación del modelo de entrada (C_1, R_1, C_2) y las variable del patrón (V_3, V_4). De la información del patrón obtiene que V_3 parametriza las instancias de C_3 y V_4 las instancias de C_4, además de que existe R_2 que asocia C_3 y C_4.

7.1.2.3 Reglas de identidad

Las reglas de identidad definen el conjunto de atributos y relaciones que permiten identificar de forma inequívoca cada instancia en el conjunto de datos de salida. Estas reglas se utilizan para evitar la redundancia en la creación de nuevos individuos. Esta característica da soporte a la transformación de distintos repositorios de entrada en un mismo repositorio de salida, ya que una

regla de identidad permite identificar qué entidades provenientes de distintos conjuntos de datos corresponden con la misma entidad del conjunto de datos de salida. Si consideramos PI como el conjunto de atributos y relaciones que proporcionan la identidad de una entidad C del modelo de salida, tal que dado $i_1 \in C$, la función $PI(i_1)$ obtiene los valores de identidad para la instancia i_1, la regla de identidad se formaliza en la ecuación 7.10. Es decir, si dos instancias tienen los mismos valores de identidad, se consideran la misma instancia.

$$Regla\_identidad(c) \ : \ \forall i_2 \ \in \ C, PI(i_1) \ = \ PI(i_2) \ \Rightarrow \ i_1 \ = \ i_2 \quad (7.10)$$

7.2 Aplicación a una representación semántica en OWL

El modelo de transformación definido presenta un enfoque genérico de transformación de información entre dos modelos distintos de información. Sin embargo, el interés de esta tesis es obtener una representación final de la información, basada en una arquitectura ontológica OWL, que facilite la integración, gestión y explotación de recursos biomédicos. Por lo tanto, en las siguientes secciones se considera la utilización de un modelo de salida basado en una arquitectura OWL y se muestran algunas consideraciones adicionales a tener en cuenta para este caso.

7.2.1 Definición de reglas de transformación

Tanto las reglas de correspondencia como las reglas de identidad se definen utilizando un lenguaje propio basado en XML. La tabla 7.1 presenta la gramática para las reglas de identidad, mientras que la tabla 7.2 presenta la gramática definida para el lenguaje de las reglas de correspondencia.

Tabla 7.1: Gramática para la definición de identidades

identidad	::="<condition>" **requisitoRaiz** "</condition>"
requisitoRaiz	::="<requirement>" **sujeto subRequisito** "</requirement>"
sujeto	::= "<class><id>" **IRI** "</id></class>"
subRequisito	::= **requisitoAND** | **requisitoOR**
requisitoAND	::="<and>" **expresion** "</and>"
requisitoOR	::="<or>" **expresion** "</or>"
expresion	::= **requisito** | **requisitoAND** | **requisitoOR**
requisito	::="<requirement>" **expresionObject** | **expresionData**"</requirement>"
expresionObject	::= **ambito propiedadObjeto valor clase**
expresionData	::= **ambito propiedadDatos valor**
ambito	::= "<scope>" **valorAmbito** "</scope>"
propiedadObjeto	::= "<objectproperty>" **IRI** "</objectproperty>"
propiedadDatos	::= "<dataproperty>" **IRI** "</dataproperty>"
valor	::= "<value>" **valorValue** "</value>"
clase	::= "<class>" **IRI** "</class>"
valorAmbito	::= "SOME" | "ALL"
valorValue	::= "EQUALS" | "EQUALS IGNORE CASE"

Tabla 7.2: Gramática para la definición de correspondencias

align	::= "<Alignment>" **tipo regla**+ "</Alignment>"
tipo	::= "<input>" **entrada** "</input><output>" **salida** "<output/>"
regla	::= **reglaClase** | **reglaPropiedad** | **reglaRelacion**
entrada	::= "RDB" | "ARCHETYPE" | "XML"
salida	::= "OWL"
reglaClase	::="<map><type>2Class</type>" **correspondClase** "</map>"
reglaPropiedad	::="<map><type>2Prop</type>" **correspondProp** "</map>"
reglaRelacion	::="<map><type>2Rel</type>" **correspondRel** "</map>"
correspondClase	::= **definicionClase definicionEntidad**
correspondProp	::= **correspondFuente predicado correspondDestinoProp**
correspondRel	::= **correspondFuente predicado correspondDestinoRel**
correspondFuente	::="<domain>" **correspondClase** "</domain>"
correspondDestinoProp	::="<range>"**definicionClase**? **definicionValue** "</range>"
correspondDestinoRel	::= "<range>" **correspondClase** "</range>"
predicado	::= "<predicate><id>" **caracter**+ "</id></predicate>"
definicionClase	::= "<class><id>" **caracter**+ "</id></class>"
definicionEntidad	::= "<entity>" **nodos**+ **infos*** **opcion*** "</entity>"
definicionValue	::= "<value>" **nodos**+ "</value>"
nodos	::= "<nodes>" **nodo**+ "</nodes>"
nodo	::= "<node id=" **digit**+ ">" **cuerpo** "</node>
infos	::= "<infos>" **info**+ "</infos>"
info	::= "<info>" **cuerpo** "</info>
cuerpo	::= **identificador**+ (**base** | **baseRef**)*
identificador	::= "<id>" **caracter**+ "<id>"
base	::= "<base>" **caracter**+ "<base>"
baseRef	::= "<base nodeRef=" **digit**+ "/>"
opcion	::= "<option>" **par** "</option>"
par	::= "<key>" **caracter**+ "</key><value>" **caracter**+ "</value>"
caracter	::= [^<>]
digit	::= ["0-"9"]

A continuación muestro la definición de reglas de transformación para un ejemplo concreto. La figura 7.5 muestra los esquemas de entrada (izquierda) y de salida (derecha) para la transformación de recursos de datos sobre componentes químicos. El modelo de entrada es un esquema XML, las entidades principales son ⟨ *data*, *molecule*, *atom*, *bond* ⟩. El esquema modela datos (*data*) de una molécula (*molecule*), que contiene átomos (*atom*) y enlaces entre átomos (*bond*). Los atributos del modelo vienen dados por la unión de los atributos de una molécula, ⟨ *coorddimension* ⟩; de un átomo, ⟨ *key*, *element*, *x*, *y*, *z* ⟩ y de un enlace ⟨ *atomref1*, *atomref2* ⟩. Las relaciones entre entidades vienen dadas por la jerarquía del esquema, que indica que una molécula puede tener varios átomos y varios enlaces. El resto de la información asociada a las moléculas, átomos y enlaces viene dada a través de la entidad *property*. Esta entidad se repite en el modelo y cambia el valor de atributo nombre

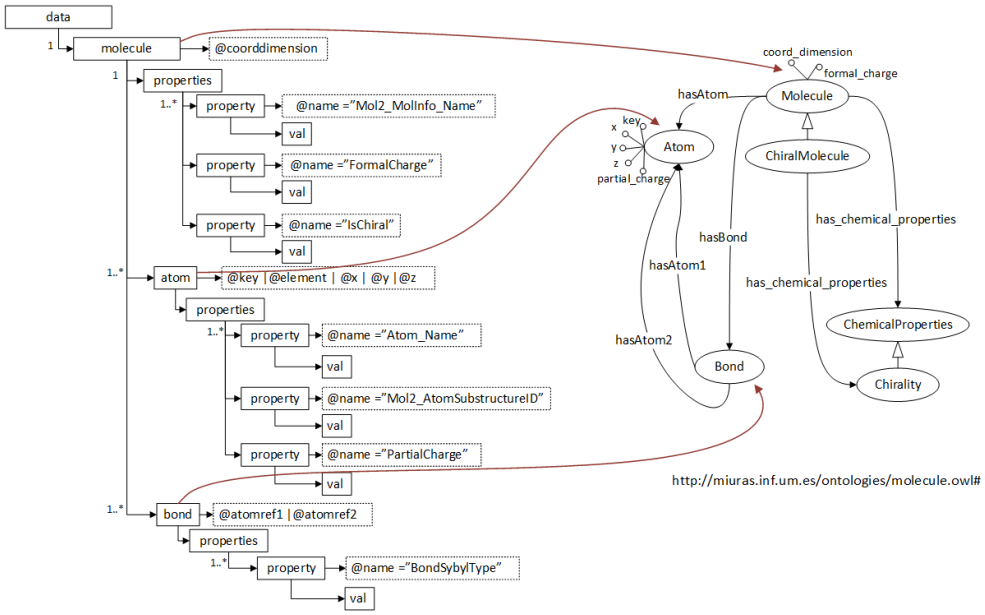

Figura 7.5: Esquema que sigue una base de datos sobre componentes químicos (izq) y ontología del dominio (dcha)

(*name*) dependiendo de la información que proporcione. Por simplicidad, la figura 7.5 muestra los tipos de propiedades más relevantes para cada una de las entidades principales. El modelo de salida es una ontología OWL, las entidades principales son las clases OWL ⟨ *Molecule*, *ChiralMolecule*, *Atom*, *Bond*, *ChemicalProperties*, *Chirality* ⟩, los atributos vienen dados por las `OWL DatatypeProperty` de *Molecule*, ⟨ *formal_ charge* ⟩; y de *Atom*, ⟨ *key, x, y, z, partial_ charge* ⟩. Las entidades se asocian entre sí a través de las relaciones ⟨ *hasAtom, has_ chemical_ properties, hasBond, hasAtom1, hasAtom2* ⟩. Vemos que las clases *Molecule*, *Atom* y *Bond* tienen una equivalencia directa con elementos del esquema XML. Veamos cómo se definen las reglas de transformación:

La entidad *molecule* del modelo de entrada corresponde con la clase *Molecule* del modelo de salida. La regla de transformación que define esta relación aparece en la figura 7.6. La primera parte de la regla (`<class>`), muestra el acceso a la clase *Molecule* a través de su identificador, en este caso la IRI. La segunda parte (`<entity>`), define la entidad en el esquema fuente. Contiene las etiquetas `<nodes>` e `<infos>`, dentro de `<nodes>` se define el acceso a la entidad a transformar. En este caso, al ser acceso a un XML, aparece el nombre del nodo XML (`<id>molecule</id>`), y la forma de acceso al mismo (`<base>`), que en este caso es una ruta de nodos desde el raíz. Por lo tanto,

```
<map>
    <type>2Class</type>
    <class>
        <id>http://miuras.inf.um.es/ontologies/molecule.owl#Molecule</id>
    </class>
    <entity>
        <nodes>
            <node id="1">
                <id>molecule</id><base>/datadoc/data</base>
            </node>
        </nodes>
        <infos>
            <info>
                <id>val</id>
                <base>
                    /datadoc/data/molecule/properties/property[@name="Mol2_MolInfo_Name"]
                </base>
                <base nodeRef="1"/>
            </info>
        </infos>
    </entity>
</map>
```

Figura 7.6: Regla de clase que define la correspondencia entre la entidad *molecule* y la clase *Molecule*

por cada nodo *molecule* en un XML de entrada, se creará un individuo de tipo *Molecule* en la ontología de salida. Cada `node` está identificado por su atributo `id`. Entre las etiquetas `<infos>` se proporcionan los datos que pueden ser usados para construir la IRI de los nuevos individuos. Estos datos se definen con respecto a la entidad que está siendo enlazada. En este caso, se utiliza el nombre de la molécula, que se encuentra en un nodo *val*, como se define respecto al nodo *molecule*, se indica por medio de `<base nodeRef=‘‘1’’>`, donde `nodeRef` hace referencia al atributo `id` del `node` referenciado, pero como el acceso no es directo, se indica también la ruta de nodos desde la raíz. La IRI se construye con la información obtenida de ese atributo, tomando como namespace por defecto el de la ontología, si no se especifica lo contrario.

Si se aplica la regla de la figura 7.6 a los datos XML mostrados en la figura 7.7, el resultado será el siguiente:

```
Prefix:
molecule:<http://miuras.inf.um.es/ontologies/molecule.owl#>
Individual: molecule:ZINC01532215_1
    Types:
        molecule:Molecule
```

```
<datadoc>
  <data>
    <molecule coorddimension="3">
      <properties>
        <property name="Mol2_MolInfo_Name">
          <val>ZINC01532215_1<val>
        <property>
        <property name="FormalCharge"><val>0</val></property>
        <property name="IsChiral"><val>1</val></property>
      </properties>
      <atom key="0" element="6" x="8.2965" y="42.3268" z="14.1097">
        <properties>
          <property name="AtomName"><val>C1</val></property>
          <property name="Mol2_AtomSubstructureID"><val>1</val></property>
          <property name="PartialCharge"><val>-0.0017</val></property>
        </properties>
      </atom>
      <atom key="1" element="6" x="9.1534" y="42.0172" z="13.0393">
        <properties>
          <property name="AtomName"><val>C2</val></property>
          <property name="Mol2_AtomSubstructureID"><val>1</val></property>
          <property name="PartialCharge"><val>-0.0750</val></property>
        </properties>
      </atom>
      <bond atomref1="0" atomref2="1">
        <properties>
          <property name="BondSybylType"><val>ar</val></property>
        </properties>
      </bond>
    </molecule>
  </data>
</datadoc>
```

Figura 7.7: Ejemplo XML para una molécula con dos átomos y un enlace

La molécula de la figura 7.7 tiene valor 3 para su atributo *coorddimension*. Este atributo se puede corresponder con la `OWL:datatypeProperty` *coord_ dimension* asociada con *Molecule* en el modelo de salida. La regla que define esta correspondencia se muestra en la figura 7.8. En esta regla, la primera parte (`<domain>`) corresponde a la regla de clase para *Molecule* (7.6), sin embargo, no es necesario especificar la información para la IRI, pues el individuo no se vuelve a crear de nuevo, solo es necesario especificar la entidad (nodo), del cual depende el atributo *coorddimension*. La parte central de la regla (`<predicate>`), especifica la nueva `OWL:datatypeProperty` que se va a añadir al individuo que fue creado con la regla de clase. La última parte de la regla (`<range>`), indica el acceso al valor de *coordimension* que será utilizado para dar valor a la `OWL:dataproperty`. Como vemos, el acceso es relativo al nodo *molecule* (`<base nodeRef=“1”/>`).

Si se aplica la regla de la figura 7.8 a los datos XML mostrados en la

```
<map>
    <type>2Prop</type>
    <domain>
        <class>
            <id>http://miuras.inf.um.es/ontologies/molecule.owl#Molecule</id>
        </class>
        <entity>
            <nodes>
                <node id="1">
                    <id>molecule</id><base>/datadoc/data</base>
                </node>
            </nodes>
        </entity>
    </domain>
    <predicate>
        <id>http://miuras.inf.um.es/ontologies/molecule.owl#coord_dimension</id>
    </predicate>
    <range>
        <value>
            <nodes>
                <node id="2">
                    <id>@coorddimension</id><base nodeRef="1"/>
                </node>
            </nodes>
        </value>
    </range>
</map>
```

Figura 7.8: Regla de propiedad que define la correspondencia entre el atributo *coorididimension* de *molecule* y la `OWL:dataproperty` *coord_dimension* de *Molecule*

figura 7.7, el resultado será el siguiente:

```
Prefix:
molecule:<http://miuras.inf.um.es/ontologies/molecule.owl#>
Individual: molecule:ZINC01532215_1
    Types:
        molecule:Molecule
    Facts:
     molecule:coord_dimension  3
```

La molécula de la figura 7.7 contiene dos átomos (nodos tipo *atom*). En la ontología del modelo de salida, esta asociación se modela a través de la asociación *Molecule hasAtom Atom*. La regla que define esta correspondencia se muestra en la figura 7.9. En esta regla, la primera parte (`<domain>`), al igual que en el caso anterior, corresponde con la regla de clase para *Molecule*. La parte central de la regla (`<predicate>`), especifica la nueva

```
<map>
    <type>2Rel</type>
    <domain>
        <class>
            <id>http://miuras.inf.um.es/ontologies/molecule.owl#Molecule</id>
        </class>
        <entity>
            <nodes><node id="1"><id>molecule</id><base>/datadoc/data</base></node></nodes>
        </entity>
    </domain>
    <predicate>
        <id>http://miuras.inf.um.es/ontologies/molecule.owl#has_atom</id>
    </predicate>
    <range>
        <class>
            <id>http://miuras.inf.um.es/ontologies/molecule.owl#Atom</id>
        </class>
        <entity>
            <nodes><node id="2"><id>atom</id><base nodeRef="1"/></node></nodes>
            <infos>
                <info><id>@key</id><base nodeRef="2"/></info>
                <info>
                    <id>val</id>
                    <base>
                        /datadoc/data/molecule/properties/property[@name="Mol2_MolInfo_Name"]
                    </base>
                    <base nodeRef="2"/>
                </info>
            </infos>
        </entity>
    </range>
</map>
```

Figura 7.9: Regla de relación que define la correspondencia entre la asociación entre *molecule* y *atom* en el modelo de entrada y la relación *hasAtom* entre *Molecule* y *Atom* en la ontología de salida

`OWL:ObjectProperty` que se va a añadir al individuo que fue creado con la regla de clase. La última parte de la regla (`<range>`), indica una regla de clase para *Atom*. En esta regla de clase, la primera parte (`<class>`) indica la IRI de la clase *Atom* en la ontología. En la segunda parte de la regla, aparecen los accesos a la información del nodo *atom* en el XML de entrada. En `<node>` localiza el nodo *atom* a partir del nodo *molecule* anterior (`<base nodeRef=‘‘1’’/>`. En `<infos>` aparece la localización de los datos que serán usados para construir la IRI del nuevo individuo creado.

Si se aplica la regla de la figura 7.9 a los datos XML mostrados en la figura 7.7, el resultado será el siguiente:

```
Prefix:
molecule:<http://miuras.inf.um.es/ontologies/molecule.owl#>
Individual: molecule:ZINC01532215_1
```

```
    Types:
        molecule:Molecule
    Facts:
     molecule:coord_dimension  3,
     molecule:has_atom  molecule:0_ZINC01532215_1,
molecule:has_atom  molecule:1_ZINC01532215_1
Individual: molecule:0_ZINC01532215_0
Types:
        molecule:Atom
Individual: molecule:0_ZINC01532215_1
Types:
        molecule:Atom
```

En la ontología, ChiralMolecule es una clase equivalente que se define como una molécula con la propiedad química de quiralidad, es decir, *Molecule has_ chemical_ property some Chirality*. En los recursos de datos de entrada, valores como el de quiralidad se modelan a través de la entidad *property* cuyo atributo *name* puede tener el valor *"isChiral"*, y cuyo elemento *val* tendrá valor "1" o "0" dependiendo de si la molécula es quiral o no. La ontología ya contiene un individuo de tipo *Chirality*, *"chirality"*, por lo que al crear individuos de tipo *Molecule*, si la propiedad *"isChiral"* tiene valor "1", el individuo resultado debe tener también la asociación *has_ chemical_ property chirality*. La forma más sencilla de añadir la propiedad de quiralidad a todas las moléculas con valor "1" en la propiedad *"isChiral"* es a través de un patrón.

```
?chiralMolecule:INDIVIDUAL
BEGIN
ADD ?chiralMolecule instanceOf Molecule,
ADD ?chiralMolecule has_chemical_properties chirality
END;
```

Figura 7.10: Patrón que define a una molécula quiral

La figura 7.10 muestra el patrón utilizado para siguiendo la gramática de OPPL2. Este patrón está formado por el conjunto S' compuesto por la entidad *Molecule*, la relación *has_ chemical_ properties* y la instancia *chirality*. El conjunto V de variables está formado por la variable *?chiralMolecule*, y el patrón define las asociaciones que indican que *?chiralMolecule* parametriza las instancias de *Molecule* y que dichas instancias se asocian con *chirality* a

través de la relación *has_chemical_properties*. Este patrón genera una plantilla de regla de clase para la variable *?chiralMolecule* que se muestra en la ecuación 7.11.

$$Plantilla\_regla\_clase(C_1, ?chiralMolecule)\{ \\ ?chiralMolecule \in Molecule \wedge \forall i_1 \in C_1, \exists i_2 \subseteq ?chiralMolecule \mid i_2 \cong i_1\} \tag{7.11}$$

La utilización del patrón instancia la plantilla y crea la regla de clase mostrada en la figura 7.11, para enlazar la variable a cualquier molécula con la propiedad *property "isChiral"* cuyo elemento *val* tenga valor 1.

```
<type>2Class</type>
<class>
    <id>http://www.coode.org/oppl/variablemansyntax#?chiralMolecule</id>
</class>
<entity>
    <nodes>
        <node id="1">
        <id>property[@name="IsChiral" and val="1"]</id>
        <base>/datadoc/data/molecule/properties/</base>
        <node>
    </nodes>
    <infos>
        <info>
            <id>val</id>
            <base>
                /datadoc/data/molecule/properties/property[@name ="Mol2_MolInfo_Name"]
            </base>
            <base nodeRef="1"/>
        </info>
    </infos>
</entity>
```

Figura 7.11: Instanciación de la plantilla de regla de clase para mapear moléculas quirales

Si a la molécula de la figura 7.7 le aplicamos este patrón, el resultado final del individuo creado es:

```
Prefix:
molecule: <http://miuras.inf.um.es/ontologies/molecule.owl#>
Individual: molecule:ZINC01532215_1
    Types:
        molecule:Molecule
    Facts:
```

```
     molecule:has_chemical_properties molecule:chirality,
     molecule:coord_dimension  3,
     molecule:has_atom  molecule:0_ZINC01532215_1,
molecule:has_atom  molecule:1_ZINC01532215_1
Individual: molecule:0_ZINC01532215_0
Types:
        molecule:Atom
Individual: molecule:0_ZINC01532215_1
Types:
        molecule:Atom
```

Las reglas de identidad se definen sobre el modelo de salida, en este caso la ontología en el dominio de los componentes químicos. Podemos definir una regla de identidad para la clase *Bond*. En el modelo de salida, un enlace se identifica por la molécula a la que pertenece y por los dos átomos que enlaza, por lo tanto, utilizando las `OWL ObjectProperties` *in_ molecule*, *has_ atom1* y *has_ atom2* se define la regla de identidad. La figura 7.12 muestra dicha regla de identidad. La primera parte de la regla (`<class>`) identifica la clase de la ontología sobre la que se define la regla, es decir, todas las instancias de *Bond* deberán cumplir la regla definida. En la siguiente parte de la regla se define un requisito (`<requirement>`). La etiqueta `<and>` indica que todos los requisitos incluidos deben cumplirse obligatoriamente (una etiqueta `<or>` indicaría que al menos uno debe cumplirse). El primer requisito define que para todas (`<scope>ALL</scope>`) las asociaciones *in_ molecule* (`<objectproperty>`) de dos individuos, si el objeto *Molecule* (`<class>`) es el mismo (`<value>EQUALS</value>`), ambas instancias cumplen dicho requisito para ser idénticas. El ámbito (`<scope>`) también puede tener el valor `SOME`, en cuyo caso la regla sólo requiere cumplirse para alguna de las asociaciones. Los requisitos pueden crearse con `OWL DatatypeProperties`, en cuyo caso, la etiqueta `<objectproperty>` cambia a `<datatypeproperty>` y desaparece la etiqueta `<class>`. El valor (`<value>`) también puede ser "`EQUALS IGNORE CASE`" en el caso de que se estén comparando cadenas de texto.

```
<condition>
    <class><id>http://miuras.inf.um.es/ontologies/molecule.owl#Bond</id></class>
    <requirement>
        <and>
            <requirement>
                <scope>ALL</scope>
                <objectproperty>
                    http://miuras.inf.um.es/ontologies/molecule.owl#in_molecule
                </objectproperty>
                <value>EQUALS</value>
                <class>
                    http://miuras.inf.um.es/ontologies/molecule.owl#Molecule
                </class>
            <requirement>
            <requirement>
                <scope>ALL</scope>
                <objectproperty>
                    http://miuras.inf.um.es/ontologies/molecule.owl#has_atom1
                </objectproperty>
                <value>EQUALS</value>
                <class>
                    http://miuras.inf.um.es/ontologies/molecule.owl#Atom
                </class>
            </requirement>
            <requirement>
                <scope>ALL</scope>
                <objectproperty>
                    http://miuras.inf.um.es/ontologies/molecule.owl#has_atom2
                </objectproperty>
                <value>EQUALS</value>
                <class>
                    http://miuras.inf.um.es/ontologies/molecule.owl#Atom
                </class>
            </requirement>
        </and>
    </requierement>
</condition>
```

Figura 7.12: Regla de identidad para la clase *Bond*

7.2.2 Arquitectura y algoritmo de transformación

La arquitectura del modelo de transformación se muestra en la figura 7.13. Las entradas al modelo son los recursos de datos junto al modelo de entrada que los definen, el modelo de salida y, opcionalmente, patrones de contenido. Las reglas de correspondencia se obtienen a partir de su definición entre el modelo de entrada y el modelo de salida, o entre el modelo de entrada y los patrones de contenido. Con el modelo de salida se definen las reglas de identidad y finalmente la ejecución de las reglas de correspondencia sobre los recursos de entrada extraen las instancias necesarias. Las reglas de identidad se utilizan para validar los datos extraídos, desechando aquellos datos que corresponden a entidades redundantes. Por último, los datos extraídos y validados se transforman siguiendo el modelo de salida.

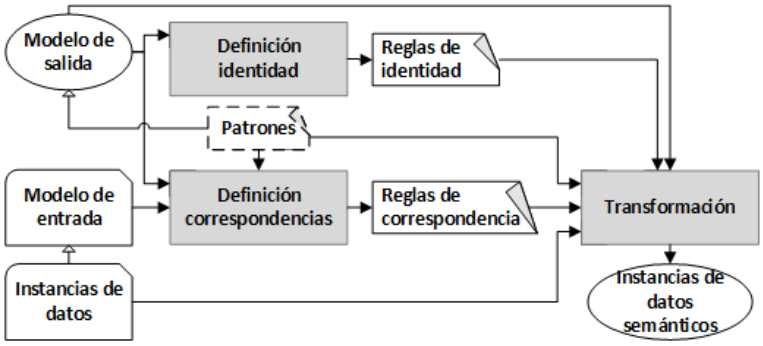

Figura 7.13: Arquitectura del modelo de transformación

La transformación de datos a representación semántica se lleva a cabo a través de la ejecución de las reglas de transformación e identidad. El algoritmo 1 muestra la sucesión de pasos que se llevan a cabo. Para cada clase c en el modelo de salida S, se seleccionan sus reglas de correspondencia asociadas. Las reglas de clase definen los datos para crear el individuo como instancia de una clase, mientras que las de propiedad y relación añaden valores a sus propiedades y lo enlazan con otros individuos. Si existen patrón y reglas asociadas a los mismos, estas reglas se ejecutan y los patrones se instancian. Un patrón instanciado puede ser utilizado para crear nuevos individuos o para añadir información adicional a individuos ya creados. Hecho esto, se comprueba con sus reglas de identidad que el individuo sea único. En ese caso, se añade el individuo al conjunto de individuos I del modelo de salida S.

La complejidad del algoritmo se calcula como la fórmula 7.12, dónde se cumple:

- c: máximo número de entidades.
- i: máximo número de instancias obtenidas a partir de la regla de clase.
- r_p: máximo número de reglas de patrón para cada entidad.
- ip: máximo número de elementos obtenidos a partir de la regla de patrón.
- r_a máximo número de reglas de propiedad para cada entidad.
- a: máximo número de valores de atributos obtenidos a partir de la regla de propiedad

- r_r: máximo número de reglas de relación para cada entidad
- r: máximo número de instancias relacionadas obtenidas a partir de la regla de relación.
- r_i: máximo número de reglas de identidad para cada entidad.
- i_t: máximo número de nuevas instancias totales creadas.

Algoritmo 1: Algoritmo de transformación

Data:
C : conjunto de entidades del modelo de salida
I_E: conjunto de instancias en el modelo de entrada
I_S : conjunto de instancias en el modelo de salida
R_C : conjunto de reglas de clase
R_P: conjunto de reglas asociadas a un patrón
P : conjunto de patrones
R_A : conjunto de reglas de propiedad
R_R : conjunto de reglas de relación
R_I : conjunto de reglas de identidad

```
1  for c ∈ C do
2      rc' ⟵ seleccionaReglaClase(R_C, c);
3      R'_P ⟵ seleccionaReglasPatron(R_P, c);
4      R'_A ⟵ seleccionalReglasPropiedad(R_A, c);
5      R'_R ⟵ seleccionaReglasRelacion(R_R, c);
6      R'_I ⟵ seleccionaReglasIdentidad(R_I, c);
7      NI ⟵ ejecutaReglaClase(rc', c, I_E);
8      for nuevaInstancia ∈ NI do
9          if R'_P ≠ ∅ then
10             for rp' ∈ R'_P do
11                 p' ⟵ seleccionaPatron(P, rp');
12                 addInfoPatron(p', c, nuevaInstancia, I_E);
13         for ra' ∈ R'_A do
14             addPropiedades(ra', c, nuevaInstancia, I_E);
15         for rr' ∈ R'_R do
16             addRelaciones(rr', c, nuevaInstancia, I_E);
17         for ri' ∈ R'_I do
18             if cumpleIdentidad(nuevaInstancia, ri', I_S) = true then
19                 I_S ⟵ I_S + nuevaInstancia;
```

$$
\begin{aligned}
complejidad\_algoritmica = O(c) * (\\
O(1) + t_1 + O(i) * (\\
O(r_p) * (O(1) + t_2 + O(ip) * O(1)) + \\
O(r_a) * (t_3 + O(a) * O(1)) + \\
O(r_r) * (t_4 + O(r) * O(1)) + \\
O(r_i) * (t_5 + O(1)))) + O(i_t)^2
\end{aligned} \quad (7.12)
$$

- $O(c)*$: por cada entidad (línea 1 en el algoritmo).
- $O(1) + t_1+$: $O(1)$ es el tiempo constante de acceder a las estructuras HashMap que contienen las reglas (líneas 2 a 6 en el algoritmo). t_1 se refiere al tiempo de ejecutar la regla de clase, que depende del tipo de fuente de entrada y la librería utilizada para manipular los recursos (línea 7).
- $O(i)*$: por cada instancia recuperada (línea 8 en el algoritmo).
- $O(r_p) * (O(1) + t_2 + O(ip) * O(1))+$: $O(r_p)*$, por cada regla de patrón (línea 10) se selecciona el patrón ($O(1)$, línea 11) y la siguiente instrucción (línea 12) se divide en t_2, que es el tiempo de ejecución de la regla de patrón, y $O(ip) * O(1)$, por cada instancia de patrón generada se añade la información a la instancia en un tiempo constante.
- $O(r_a) * (t_3 + O(a) * O(1))+$: $O(r_a)*$, por cada regla de propiedad (línea 13), se ejecuta la línea 14, donde se ejecuta la regla (t_3) y por cada valor de atributo recuperado se añade la información a la instancia ($O(a) * O(1)$).
- $O(r_r) * (t_4 + O(r) * O(1))+$: $O(r_r)*$, por cada regla de relación (línea 15), se ejecuta la línea 16, donde se ejecuta la regla (t_4) y por cada instancia relacionada recuperada (de la cual sólo se construye la URI) se añade la información a la instancia ($O(r) * O(1)$).
- $O(r_i) * (t_5 + O(1))))$: $O(r_i)*$, por cada regla de identidad (línea 17), se comprueba si ésta se cumple en la línea 18 (t_5, tiempo de ejecución de la regla de identidad), y en tal caso se añade la instancia al listado de instancias ($O(1)$, línea 19).
- $O(i_t)^2$: al tiempo de ejecución anterior se le suma el tiempo dedicado para hacer todas las nuevas instancias distintas entre sí, es decir añadir el axioma `owl:differentFrom` a cada nueva instancia respecto al resto.

Por un lado, los tiempos t_1, t_2, t_3, t_4, t_5 depende del tipo de recurso de entrada y de salida, pero no del tamaño de los mismos. Por otro lado, la complejidad del algoritmo se concentra en el número de iteraciones que tiene que realizar, por lo que se pueda simplificar la fórmula de la complejidad como se muestra a continuación:

$$complejidad\_algoritmica = O(c) * (O(i) * (\\ (O(r_p) * O(ip)) + (O(r_a) * O(a)) + (O(r_r) * O(r)) + O(r_i))) + O(i_t)^2 \quad (7.13)$$

Es decir, la complejidad depende del número de instancias encontradas para cada clase y del número de propiedades y relaciones que tengan dichas instancias.

7.2.3 Transformación basada en clases

La representación de información en bases de datos relacionales o XML difiere de su representación en una ontología. Tanto en bases de datos relacionales como en un XML hay una clara distinción entre el esquema y las instancias de datos, pero en la representación en OWL la diferencia no es tan clara. En este tipo de representación, ser una instancia o una clase es un rol que adquiere un concepto [230]. La representación del conocimiento, por lo tanto, puede estar basada en instancias o basada en clases. La decisión de una representación u otra depende del uso destinado a la misma.

La representación del dominio basada en clases permite crear ontologías siguiendo buenas prácticas como la reutilización. Un modelo de salida es reutilizado para crear un nuevo modelo de salida que permita la representación del dominio de los recursos de entrada. Es decir, parte del proceso de transformación permite crear nuevas clases (entidades) en el modelo de salida, a partir del dominio representado por los recursos de entrada, mientras que el resto de la transformación crea instancias en el nuevo modelo de salida, a partir de los datos contenidos en los recursos de entrada.

Para llevar a cabo la adaptación del modelo de entrada se definen dos nuevos tipos de reglas de correspondencia. Dadas dos entidades C_{e1} y C_{s1}, donde C_{s1} (perteneciente al modelo de salida) es congruente respecto a C_{e1} (perteneciente al modelo de entrada), con la característica adicional que se cumple que la definición de C_{s1} es una generalización de C_{e1}, la regla 7.14 define la regla que permite crear subclases de C_{s1}, a partir de instancias de C_{e1}. El resultado es que por cada instancia i_{e1} de C_{e1} se crea una nueva entidad C_{s2} en el modelo de salida que modela ($\models$) a la instancia i_{e1}.

$$Regla\_individuo\_clase(C_{e1}, C_{s1}) : \forall i_{e1} \in C_{e1}, \exists C_{s2} \sqsubseteq C_{s1} \mid C_{s2} \models i_{e1} \tag{7.14}$$

La nueva entidad resultado modela la información representada por la instancia i_{e1}, así, por cada atributo y relación asociado a i_{e1}, se creará una asociación congruente en C_{s2}, siguiendo las reglas 7.15 y 7.16.

$$Regla\_individuo\_propiedad((C_{e1}, A_1), C_{s2}) : \forall i_{e1} \in C_{e1}, \exists value(C_{e1}, A_1, i_{e1}) \mid \exists A_2 \cong A_1 \wedge \forall i_{s2} \in C_{s2}, \exists value(C_{s2}, A_2, i_{s2}) \tag{7.15}$$

$$Regla\_individuo\_relacion((C_{e1}, R_1, C_{e2}), C_{s2}) : \forall i_{e1} \in C_{e1}, i_{e2} \in C_{e2}, \exists relacion(i_{e1}, R_1, i_{e2}) \mid \exists R_2 \cong R_1 \wedge \exists C_{s3} \models i_{e2} \wedge \forall i_{s2} \in C_{s2}, i_{s3} \in C_{s3}, \exists relacion(i_{s2}, R_2, i_{s3}) \tag{7.16}$$

Una variación de la regla 7.14 es la creación de una nueva entidad en el modelo de salida a partir de las instancias de una entidad del modelo de entrada, sin que exista previamente una entidad congruente en el modelo de salida. Esta regla se define como:

$$Regla\_individuo\_clase2(C_{e1}) : \forall i_{e1} \in C_{e1}, \exists C_{s1} \models i_{e1} \tag{7.17}$$

Esta aproximación se puede extender a la transformación entre entidades. Dadas las entidades C_{e1} y C_{s1} del caso anterior (C_{s1} congruente respecto a C_{e1} pero con una definición más general), la ecuación 7.18 define la regla para transformar la entidad C_{e1} en una nueva entidad C_{s2} en el modelo de salida, subclase de C_{s1}, de manera que C_{s2} sea congruente respecto a C_{e1}. De la misma manera que en el caso anterior, los atributos y relaciones asociados a C_{e1} se transforman a su vez como nuevos atributos y relaciones asociados a C_{s2}. Esta nueva regla puede servir para aplicar la regla de clase 7.4 entre las entidades C_{e1} y C_{s2} y crear nuevas instancias de C_{s2} a partir de las instancias de C_{e1}.

$$Regla\_clase\_clase(C_{e1}, C_{s1}) : \exists C_{s2} \sqsubseteq C_{s1} \mid C_{s2} \cong C_{e1} \tag{7.18}$$

Y su variación en el caso de que no exista previamente una entidad congruente en el modelo de salida:

$$Regla\_clase\_clase2(C_{e1}) \quad : \quad \exists C_{s1} \quad \cong \quad C_{e1} \quad (7.19)$$

La aproximación de transformación a clases también permite la utilización de patrones. La única diferencia es que el conjunto de variables V que forma el patrón parametriza entidades del mismo en lugar de instancias. La plantilla de regla instancia clase que se define en para este patrón se muestra en la ecuación 7.20.

$$Plantilla\_regla\_instancia\_clase(C_{e1}, V_2)\{ \\ V_2 \in C_{s1} \wedge \forall i_{e1} \in C_{e1}, \exists C_{s2} \subseteq V_2 \mid C_{s2} \models i_{e1}\} \quad (7.20)$$

De la misma forma, se pueden utilizar patrones para crear nuevas entidades a partir de entidades existentes en el modelo de entrada. La plantilla de regla se define como:

$$Plantilla\_regla\_clase\_clase(C_{e1}, V_2)\{ \\ V_2 \in C_{s1} \wedge \exists C_{s2} \subseteq V_2 \mid C_{s2} \cong C_{e1}\} \quad (7.21)$$

7.2.4 Transformación de modelos

El modelo de transformación planteado obtiene la representación según un modelo de salida de contenidos de entrada representados de forma diferente en origen. Esta propuesta parte de un modelo de salida ya existente, sin embargo, la obtención de un modelo de salida a partir de un modelo de entrada tiene especial interés en el contexto de la HCE. Permite obtener un modelo clínico a partir de otro de entrada existente, de manera que facilita la reutilización de modelos clínicos en otros estándares, permitiendo poner a disposición de otras instituciones y comunidades científicas modelos existentes, compartiendo el conocimiento clínico y facilitando la interoperabilidad semántica de la información clínica.

El modelo de transformación propuesto en este capítulo puede ser aplicado a la transformación de un modelo de entrada en un modelo de salida, en lugar de la transformación de instancias. En este modelo, la creación de nuevas entidades se basa en añadir nuevos axiomas a un modelo ontológico base. En el ámbito clínico es especialmente aplicable esta metodología, pues en los últimos años se han desarrollado representaciones basada en ontologías para

estándares de HCE basados en el modelo dual (ver sección 3.5). Muchas de estas representaciones siguen la metodología de dos niveles del modelo dual (ver sección 2.1.1), los conceptos principales del modelo de información se representan como clases de una ontología base, mientras que los modelos clínicos, que en el modelo dual restringen las clases del modelo de información, se representan como subclases del modelo de información.

El resultado de esta adaptación es la obtención de un nuevo modelo de salida a partir de un modelo de entrada mediante la aplicación de las reglas definidas en la sección 7.2.3, que se añadiendo nuevos axiomas a un modelo base.

7.3 Semantic Web Integration Tool (SWIT)

La propuesta de modelo de transformación ha sido implementada en la herramienta SWIT (Semantic Web Integration Tool [231]). Esta herramienta proporciona una interfaz web que guía al usuario por las etapas del modelo. Primero, el usuario debe seleccionar los modelos de entrada y salida. Actualmente SWIT da soporte a bases de datos MySQL, esquemas XML y arquetipos como modelos de entrada, y ontologías OWL como modelos de salida. También se seleccionan los patrones de diseño en caso de que se vayan a utilizar. Los repositorios semánticos generados por SWIT pueden estar en formato OWL o RDF, que pueden ser descargados o almacenados en una base de conocimiento Jena o Virtuoso. En este paso, el usuario especifica la estructura de la IRI para las instancias generadas. Por defecto se utiliza la IRI base de la ontología.

En el segundo paso, el usuario define las reglas de correspondencia entre los modelos de entrada y de salida o los modelos de entrada y los patrones de diseño. En este paso, correspondencias definidas anteriormente pueden ser cargadas en la herramienta y reutilizadas.

En el tercer paso, el usuario define las reglas de identidad haciendo uso del modelo de salida.

Finalmente, una vez que las reglas de transformación se han definido, se ejecutan para generar el repositorio resultado. En el proceso se aplican las reglas de correspondencia a los datos de entrada para generar el contenido semántico, comprobando las reglas de identidad para garantizar que no se crean individuos redundantes. En este proceso también se usa razonamiento automático para asegurar que solo se transforma contenido consistente.

SWIT está implementado utilizando Java y la API Google Web Toolkit. Acepta como entrada bases de datos MySQL, esquemas XML o arquetipos

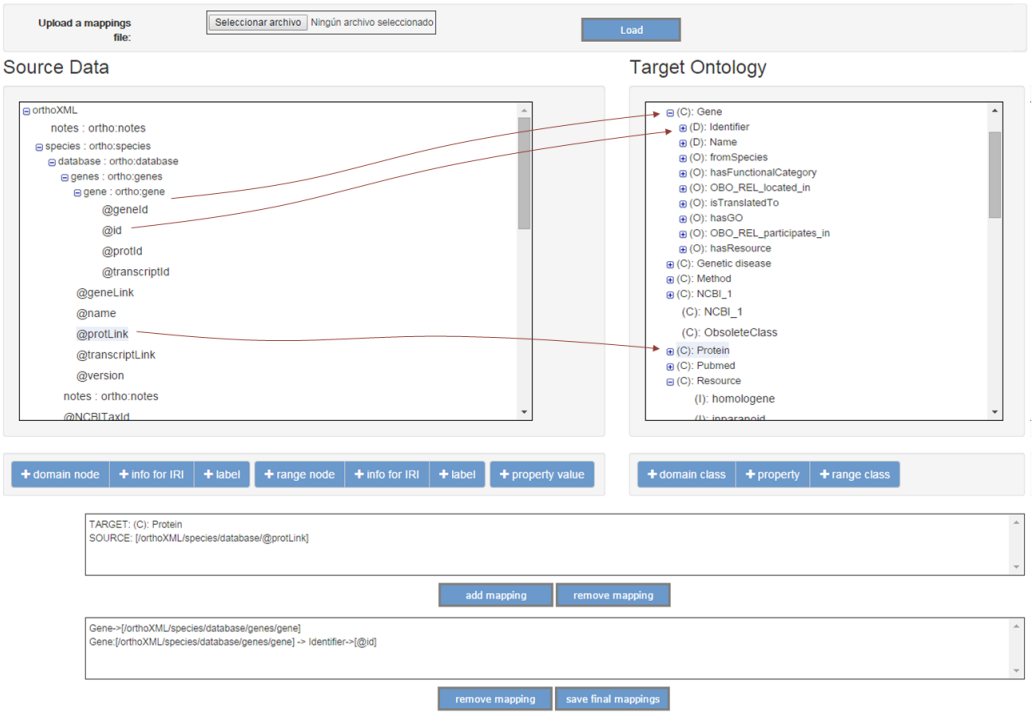

Figura 7.14: Interfaz SWIT para definición de reglas de correspondencia

ADL y ontologías OWL, además de patrones de diseño OPPL2. La generación de los repositorios semánticos se realiza utilizando la OWLAPI y Jena.

La figura 7.14 muestra la interfaz de SWIT para la definición de reglas de correspondencia. A la izquierda se muestra el modelo de entrada, en este caso, un esquema XML describiendo genes ortólogos. A la derecha la ontología que se utiliza como modelo de salida, y que modela la semántica del dominio del repositorio resultado. En la imagen las flechas muestras ejemplos de correspondencias definidas. En la parte baja de la imagen, el primer recuadro muestra una correspondencia que está siendo definida, en este caso, la correspondencia para definir nuevos individuos de la clase OWL *Protein*. El segundo recuadro muestra las correspondencias ya definidas, en este caso, la correspondencia de clase para definir nuevos individuos de la clase OWL Gene, y la correspondencia de propiedad para dar valor a la OWL dataProperty id.

La figura 7.15 muestra un ejemplo de interfaz para definir las reglas de correspondencia entre un arquetipo (izquierda) para la recogida de datos sobre el resultado de una histopatología, y un patrón de diseño que encapsula la definición de una instancia de tipo *Finding* en la ontología destino. A la

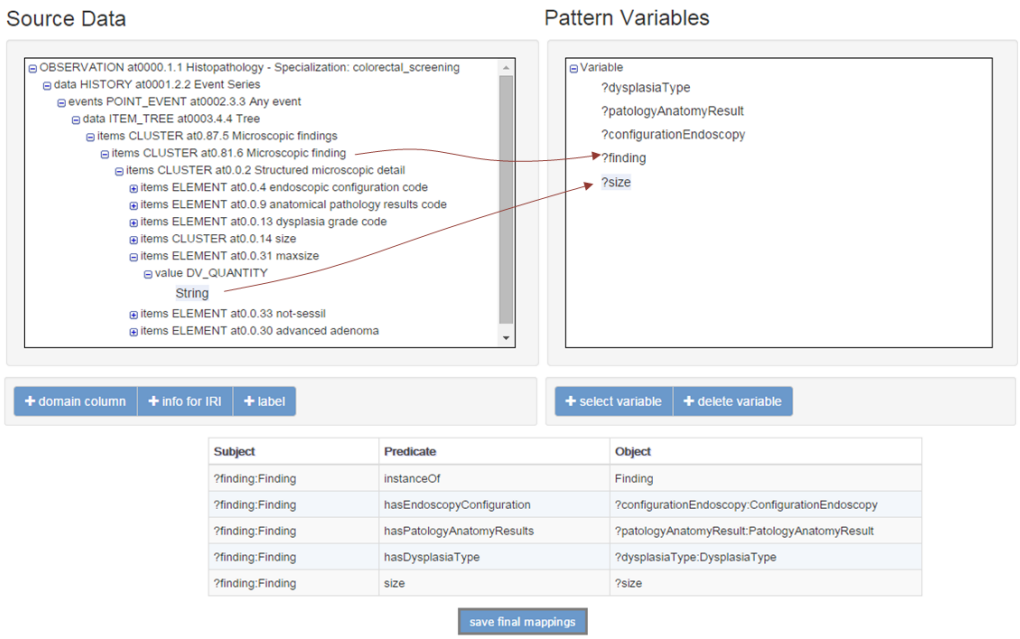

Figura 7.15: Interfaz SWIT para definición de reglas de correspondencia utilizando un patrón

derecha de la imagen aparecen las variables del patrón para las que se debe definir una regla de transformación, mientras que en la parte baja de la imagen, la tabla muestra la estructura del patrón.

7.4 Discusión

En este capítulo se ha presentado un modelo de transformación flexible, extensible y genérico que se basa en la definición de correspondencias entre un modelo de entrada, que define la estructura de la información en origen, y un modelo de salida o modelo ontológico que define la estructura destino de la información. La transformación está dirigida por el dominio, representado en el modelo de salida. Las correspondencias se definen de forma declarativa, permitiendo que sean reutilizables y que una vez definidas el método sea automático. La incorporación de patrones simplifica la definición de correspondencias y favorecen su reutilización.

El modelo de transformación se instancia para la utilización de una arquitectura ontológica como modelo de salida, formada por una ontología OWL y, opcionalmente, patrones de diseño ontológicos de contenido. La represen-

tación resultado permite explotar los recursos de entrada en un dominio de aplicación, explotando la representación con una semántica precisa que proporciona OWL. Este modelo de transformación ha sido implementado en una herramienta web, SWIT, que permite la creación de repositorios semánticos de datos abiertos de nivel 4 estrellas en el esquema propuesto por Berners-Lee (ver sección 3.4.2).

Capítulo 8

Integración de información biomédica basada en transformación

La investigación en biomedicina requiere combinar datos procedentes de varias fuentes heterogéneas. La integración de estos datos es compleja y propensa a errores por las diferencias de almacenamiento, estructura de representación, nomenclatura y nivel de detalle entre ellos. La Web Semántica ofrece tecnologías para la representación de datos que facilitan actividades de integración y explotación de los mismos.

La integración aquí expuesta se basa en la transformación de las fuentes para su homogeneización. El modelo de transformación expuesto en el capítulo anterior se aplica a cada una de las fuentes, utilizando como modelo de salida una arquitectura ontológica OWL que proporciona la semántica del dominio. El uso de este tipo de representación de salida permite transformar las fuentes de entrada en contenido legible por una máquina, y al ser un modelo global, los datos fuente quedan conectados en el dominio de representación.

8.1 Diseño del modelo de integración

El modelo de integración de varios recursos heterogéneos se basa en la aplicación del modelo de transformación a cada uno de los distintos recursos utilizando un modelo de salida global. El modelo de integración se define como la tupla $\langle \langle f_1, f_2, \ldots, f_N \rangle , \langle rc_1, rc_2, \ldots, rc_N \rangle , ri, S \rangle$, dónde $\langle f_1, f_2, \ldots, f_N \rangle$ es el conjunto de fuentes utilizadas en el proceso de integración, siendo f_i un recurso compuesto por un modelo de entrada (que cumple las

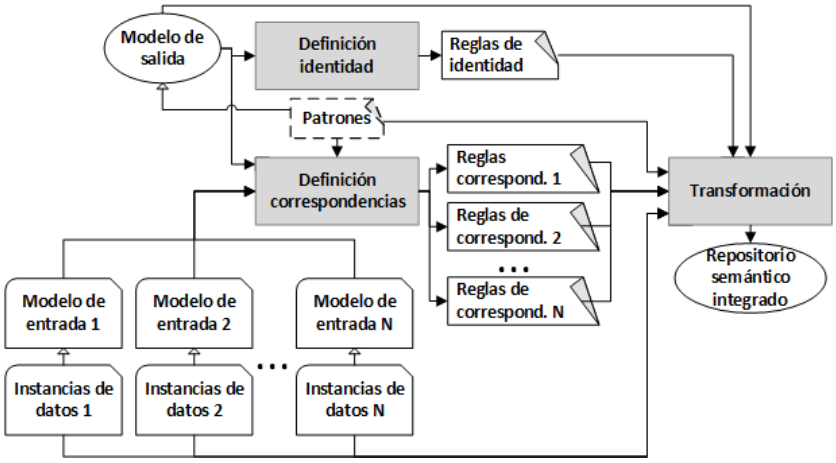

Figura 8.1: Arquitectura de integración

características descritas en la sección 7.1.1) e instancias de datos; $\langle\ rc_1, rc_2, \ldots, rc_N\ \rangle$ es el conjunto de reglas de correspondencia definidas para cada uno de los repositorios de entrada; ri corresponde con las reglas de identidad, definidas sobre el modelo de salida S. El modelo S cumple los requisitos definidos en la sección 7.1.1.

La figura 8.1 muestra la arquitectura del modelo de integración. Este modelo no está orientado a la integración de varios repositorios completos, sino a la obtención e integración de aquellos datos en los repositorios fuente, relevantes para un estudio o investigación concreta, o requeridos para su publicación. Por lo tanto, el modelo de salida se define de forma independiente a los recursos de entrada y la integración está dirigida por la semántica del dominio representada por éste.

8.1.1 Modelo de datos

El modelo de integración que se presenta en este capítulo utiliza un modelo de datos global definido de forma independiente a las fuentes de datos. Por lo tanto, la construcción del modelo de salida no requiere un alineamiento entre los esquemas de las fuentes de datos, ni es el resultado de la integración de los esquemas. La definición del modelo de salida depende del uso específico que se haga de los datos. Pueden darse dos casos: (1) el objetivo de la integración no requiere expresar toda la semántica contenida en los repositorios de entrada y, por lo tanto, solo se transforma parcialmente el contenido de los repositorios de entrada, por ejemplo, en un repositorio integrado sobre genes ortólogos, la inclusión de un nuevo organismo transformaría su información

disponible sobre genes, proteínas y sus ortólogos y podría ignorar información de secuencias genómicas; (2) el objetivo es la creación de un repositorio integrado con toda la información proveniente de varias fuentes, lo que requeriría una transformación completa de los recursos fuente y un modelo global expresando toda la semántica de origen, por ejemplo, la publicación en la Web de Datos de varios recursos integrados siguiendo los principios de Linked Data.

El modelo de datos de este sistema de integración se basa en una arquitectura ontológica. El modelo global es una ontología que expresa la semántica del dominio de aplicación. La ontología puede ir acompañada de un conjunto de patrones semánticos de contenido que ayuden en el proceso de integración. La modularización del modelo de salida por medio de patrones facilita la reutilización de reglas de transformación, pues es más probable encontrar coincidencias localizadas entre los esquemas heterogéneos de entrada, en comportamientos concretos, que encontrar las coincidencias de forma global.

La figura 8.2 muestra un extracto de la ontología utilizada como modelo global de un proyecto de integración. El proyecto realiza una integración de repositorios de genes ortólogos, enfermedades genéticas e información resultado de procesos de anotación de genomas. El repositorio integrado se construye con el propósito de explotar las relaciones entre genes ortólogos y la influencia de genes en enfermedades genéticas durante el proceso de anotación de genomas. El conjunto de fuentes a integrar está formado por el repositorio integrado OGO [224] de bases de datos de ortólogos (Inparanoid, OrthoMCL, Homologene, KOG) y de enfermedades (OMIM), junto a nuevos recursos de ortólogos en formato OrthoXML y bases de datos relacionales sobre información resultado de procesos de anotación genómica para tres organismos de la familia *Mucoraceae*. La ontología de la figura 8.2 es el resultado de ampliar la ontología del proyecto OGO para incluir el dominio de las anotaciones genómicas. Por lo tanto, el modelo definido contiene entidades en el dominio de las secuencias genómicas, *Species, Supercontig, Gene, Transcript*, o *Polypeptide*; ortólogos, *OrthologousCluster*; o enfermedades genéticas, *Disorder*.

Se pueden definir patrones para las entidades más significativas a transformar, por ejemplo, la figura 8.3 muestra el patrón que encapsula la definición de una proteína (*Polypeptide*) en la ontología de salida. Este patrón evita al usuario conocer de antemano que la definición de una proteína en el modelo de salida requiere de la definición de una instancia de *CDS* y de *Transcript*.

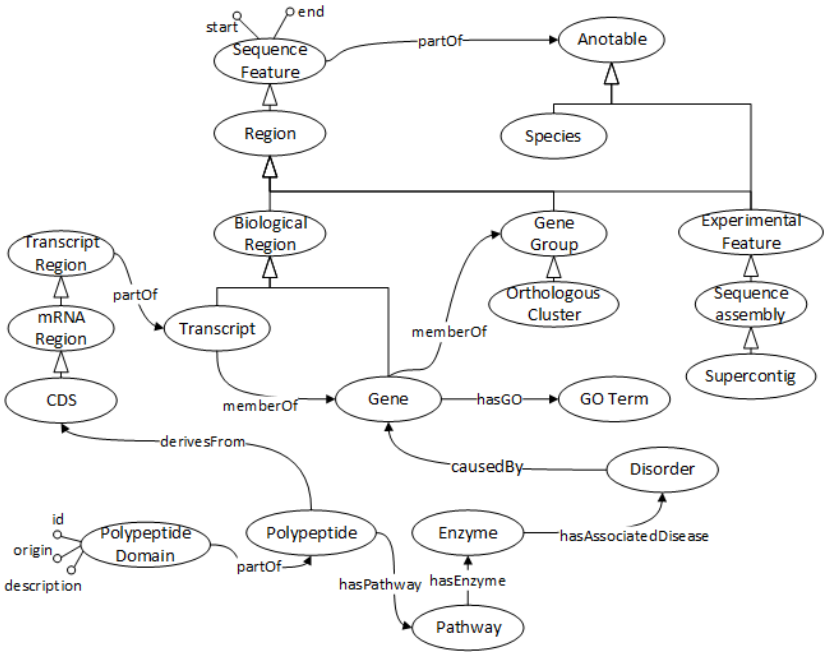

Figura 8.2: Modelo de datos para la integración de proyectos de anotación con información de genes ortólogos

```
?protein:INDIVIDUAL,
?cds:INDIVIDUAL,
?transcript:INDIVIDUAL
BEGIN
ADD ?protein instanceOf Polypetide,
ADD ?protein derivesFrom ?cds,
ADD ?cds instanceOf CDS,
ADD ?cds partOf ?transcript,
ADD ?transcript instanceOf Transcript
END;
```

Figura 8.3: Patrón de definición de una proteína

El patrón de la figura 8.3 tiene asociadas una serie de plantillas de regla que generan las reglas de correspondencia necesarias.

La plantilla 8.1 es la plantilla para un regla de clase que asocia una entidad C_1 del modelo de entrada con la variable *?protein*, de manera que se crearán nuevas instancias de la clase *Polypeptide*.

$$Plantilla\_regla\_clase(C_1, ?protein)\{ \\ ?protein \in Polypeptide \wedge \forall i_1 \in C_1, \exists i_2 \subseteq ?protein \mid i_2 \cong i_1\} \quad (8.1)$$

La plantilla 8.2 genera una regla de clase que asocia una entidad C_1 del modelo de entrada con la variable *?cds*, de manera que se crearán nuevas instancias de la clase *CDS*.

$$Plantilla\_regla\_clase(C_1, ?cds)\{ \\ ?cds \in CDS \wedge \forall i_1 \in C_1, \exists i_2 \subseteq ?cds \mid i_2 \cong i_1\} \quad (8.2)$$

La plantilla 8.3 genera una regla de clase que asocia una entidad C_1 del modelo de entrada con la variable *?transcript*, de manera que se crearán nuevas instancias de la clase *Transcript*.

$$Plantilla\_regla\_clase(C_1, ?transcript)\{ \\ ?transcript \in Transcript \wedge \forall i_1 \in C_1, \exists i_2 \subseteq ?transcript \mid i_2 \cong i_1\} \quad (8.3)$$

La plantilla 8.4 genera una regla de relación que asocia las entidades C_1 y C_2 del modelo de entrada, relacionadas a través de una relación R_1, con las variables *?protein* y *?cds*. Como resultado se generan dos nuevas instancias de *Polypeptide* y *CDS* relacionadas a través de la relación *derivesFrom*.

$$Plantilla\_regla\_relacion((C_1, R_1, C_2), (?protein, ?cds))\{ \\ ?protein \in Polypeptide \wedge ?cds \in CDS \wedge \\ Plantilla\_regla\_clase(C_1, ?protein) \wedge \\ Plantilla\_regla\_clase(C_2, ?cds) \wedge \\ \{\forall i_1 \in C_1, i_2 \in C_2, \exists i_3 \subseteq ?protein, i_4 \subseteq ?cds \mid \\ relacion(i_1, R_1, i_2) \wedge relacion(i_3, derivesFrom, i_4)\} \\ \Rightarrow R_1 \cong derivesFrom\} \quad (8.4)$$

La plantilla 8.5 genera una regla de relación que asocia las entidades C_1 y C_2 del modelo de entrada, relacionadas a través de una relación R_1, con las variables *?cds* y *?transcript*. Como resultado se generan dos nuevas instancias de *CDS* y *Transcript* relacionadas a través de la relación *partOf*.

$$Plantilla\_regla\_relacion((C_1, R_1, C_2), (?cds, ?transcript))\{ \\ ?cds \in CDS \wedge ?transcript \in Transcript \wedge \\ Plantilla\_regla\_clase(C_1, ?cds) \wedge \\ Plantilla\_regla\_clase(C_2, ?transcript) \wedge \\ \{\forall i_1 \in C_1, i_2 \in C_2, \exists i_3 \subseteq ?cds, i_4 \subseteq ?transcript \mid \\ relacion(i_1, R_1, i_2) \wedge relacion(i_3, partOf, i_4)\} \\ \Rightarrow R_1 \cong partOf\} \quad (8.5)$$

8.1.2 Transformación e integración

El proceso de integración se realiza a través de la transformación secuencial de los repositorios fuente. Primero, las reglas de transformación se utilizan para extraer los datos del recurso fuente y transformarlos a nuevas instancias en la ontología de salida. Durante el proceso de integración, las reglas de identidad cobran importancia para impedir la redundancia de datos. Se aplican para encontrar en el repositorio de salida instancias idénticas a las nuevas creadas. La representación semántica OWL de los recursos integrados sigue los principios de la Web Semántica, todos los elementos resultado están identificados por una URI de forma unívoca, de manera que dos elementos con la misma URI son considerados el mismo elemento.

La integración de recursos heterogéneos debe superar algunos problemas debido a las diferencias entre unos recursos y otros en la representación de contenidos, como son los conflictos de nombrado, datos incompletos en algunas de las fuentes e inconsistencias entre las entidades.

8.1.2.1 Conflictos de nombrado

Diferentes esquemas de entrada pueden emplear diferentes terminologías para un mismo elemento. Es decir, elementos congruentes entre sí en distintos esquemas de entrada utilizan distintos términos para ser nombrados. Estos conflictos de nomenclatura pueden afectar a cualquier entidad, atributo o relación de los esquemas de entrada.

La resolución de este tipo de conflictos en el modelo de integración viene dada a través de las reglas de correspondencia. La ontología global de salida ofrece un vocabulario homogéneo para el repositorio final integrado, de manera que los conflictos de nombrado se resuelven a través de la definición de correspondencias entre cada recurso de entrada con la ontología global.

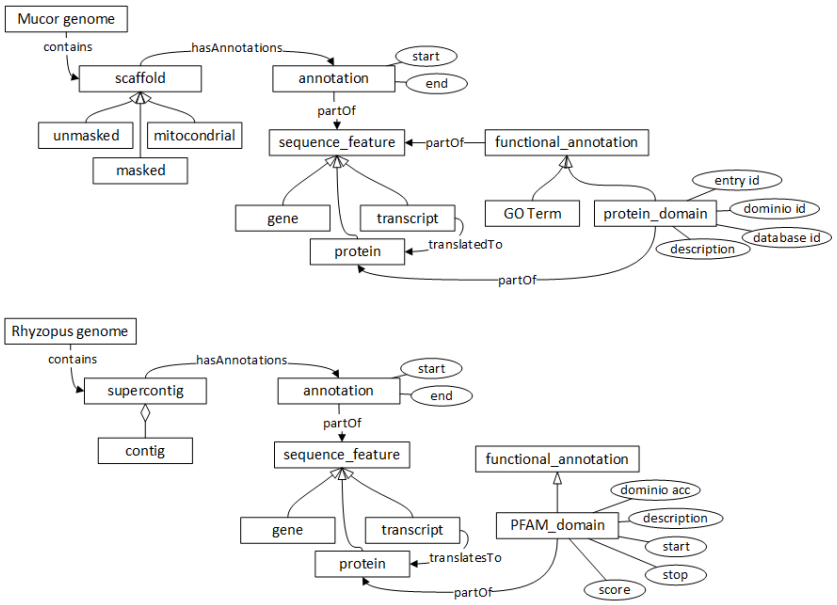

Figura 8.4: Esquemas heterogéneos de información sobre anotaciones de genomas

La figura 8.4 muestra los diagramas correspondientes a parte del esquema relacional de dos bases de datos distintas sobre organismos de la familia *Mucoraceae*. Ambos modelan información sobre secuencias genómicas y resultados de sus anotaciones. El repositorio del organismo *Mucor* (figura 8.4 arriba) divide su genoma en *scaffold*, mientras que el repositorio del organismo *Rhyzopus* (figura 8.4 abajo) lo divide en *supercontig*, que a su vez están compuestos por *contig*. Las entidades *scaffold* y *supercontig* son congruentes entre sí pero su integración causaría un conflicto de nombrado. En la integración a un repositorio modelado por la ontología de la figura 8.2, este conflicto se resuelve al definir las siguientes correspondencias:

```
Regla_clase(scaffold, Supercontig)
Regla_clase(supercontig, Supercontig)
```

La primera regla de clase se define entre el esquema *Mucor* y la ontología, y la segunda regla de clase entre el esquema *Rhyzopus* y la ontología. Es decir, todas las instancias *scaffold* de *Mucor* y todas las instancias de *supercontig* en *Rhyzopus* serán instancias de *Supercontig* en el repositorio final.

8.1.2.2 Redundancia de datos

La redundancia de datos se da cuando dos o más instancias de los repositorios de entrada describen el mismo concepto del dominio y, por lo tanto, deben corresponderse con una misma y única instancia en el repositorio final de salida.

La resolución de este problema en el modelo de integración viene dada por las reglas de identidad. Estas reglas se definen sobre la ontología global y definen los atributos y relaciones que permiten distinguir una instancia de otra.

Ambos esquemas de la figura 8.4 tienen referencias a dominios de proteínas, en *Mucor* a través de la entidad *protein_ domain* y en *Rhyzopus* a través de la entidad *PFAM_ domain*. Ambas entidades son congruentes a la entidad *PolypeptideDomain* en la ontología global, por lo que pueden aparecer dominios redundantes en ambos repositorios de entrada. Para detectar esta redundancia se define una regla de identidad para la entidad *PolypeptideDomain* tal que:

```
PI (PolypeptideDomain) = {id, origin}
```

Es decir, el conjunto PI de atributos y relaciones que proporcionan la identidad de *PolypeptideDomain* está formado por *id* y *origin*. Si existe una instancia en el repositorio *Mucor* definida como:

```
protein_domain_1 = { entry_id=``IPR000719'', dominio_id=
``PF00069'', database_id=``Pfam'', description=``Protein Kinase''}
```

Su transformación a una representación basada en la ontología global viene dada por las reglas de correspondencia:

```
Regla_clase(protein_domain, PolypeptideDomain)
Regla_propiedad((protein_domain, dominio_id),
(PolypeptideDomain, id))
Regla_propiedad((protein_domain, database_id),
(PolypeptideDomain, origin))
Regla_propiedad((protein_domain, description),
(PolypeptideDomain, description))
```

Que dan como resultado la instancia:

```
PolypeptideDomain_1 = { id=``PF00069'', origin=``Pfam'',
description=``Protein Kinase''}
```

Mientras que la instancia en el repositorio *Rhyzopus* definida como:

```
PFAM_domain_1 = { dominio_acc=‘‘PF00069’’, description=‘‘Protein
 kinase domain’’, start=‘‘7542", stop=‘‘8315", score=‘‘190.4"}
```

Se transforma a una representación basada en la ontología global con el patrón:

```
?domain:INDIVIDUAL,
?id:CONSTANT,
?desc:CONSTANT
 BEGIN
 ADD ?domain instanceOf PolypeptideDomain,
 ADD ?domain origin ‘‘Pfam’’,
 ADD ?domain id ?id,
 ADD ?domain description ?desc
 END;
```

El cual, al enlazar las variables con entidades del recurso de entrada, instancia las plantillas de reglas de la siguiente forma:

```
Plantilla_regla_clase(PFAM_domain, ?domain)
Plantilla_regla_propiedad(PFAM_domain, dominio_acc, ?id)
Plantilla_regla_propiedad(PFAM_domain, description, ?desc)
```

En este caso, el patrón permite añadir un origen fijo *“Pfam”*, pues la información de *Rhyzopus* sólo proviene de esa fuente. Aplicar estas reglas de correspondencia da como resultado la instancia:

```
PolypeptideDomain_2 = { id=‘‘PF00069’’, origin=‘‘Pfam’’,
description=‘‘Protein kinase domain’’}
```

Si primero se crea la instancia *PolypeptideDomain_1*, al crear la instancia *PolypeptideDomain_2*, la aplicación de la regla de identidad de *PolypeptideDomain* detectará la identidad con la anterior instancia. Como resultado no se creará la instancia *PolypeptideDomain_2* y la instancia *PolypeptideDomain_1* adquirirá los atributos y relaciones de *PolypeptideDomain_2*, quedando:

```
PolypeptideDomain_1 = { id=‘‘PF00069’’, origin=‘‘Pfam’’,
description={‘‘Protein Kinase’’, ‘‘Protein kinase domain’’}}
```

La transformación de instancias a través de las reglas de correspondencia da la posibilidad de utilizar campos de los datos de entrada para construir las URIs de las nuevas instancias creadas. Si las URI de las instancias de *PolypeptideDomain* se crean con la estructura *URI_ ontologia#PolypeptideDomain-_ id_ description*, las dos instancias del ejemplo anterior tendrán la misma URI y, por lo tanto, serán consideradas la misma instancia por lo que no es necesario chequear las reglas de identidad. No siempre es posible definir la identidad a través de la creación de la URI, por lo que es preferible la definición de las reglas de identidad.

8.1.2.3 Inconsistencias en los datos

En los procesos de transformación e integración de recursos de entrada pueden aparecer inconsistencia entre los recursos de entrada y el modelo global de salida. Estas inconsistencias deben ser solucionadas para finalizar la integración con éxito. Uno de los principales motivos de la aparición de estas inconsistencias es la definición del modelo global de integración de forma independiente a los recursos fuente, lo que provoca que en ocasiones estos no puedan cumplir con las restricciones definidas por éste. En otros casos, datos incompletos en las fuentes o no actualizados provocan inconsistencias con el modelo global y entre los distintos repositorios.

- **Inconsistencias debidas a datos incompletos:** este problema aparece cuando el modelo ontológico de salida requiere expresar una instancia con una serie de relaciones y atributos asociados para los que los recursos fuente no tienen toda la información. Los datos incompletos no siempre provocan una inconsistencia en el resultado final. Por ejemplo, en la figura 8.2, la clase *Enzyme* tiene una asociación con *Disorder* a través de la relación *hasAssociatedDisease*. Sin embargo, no es una asociación necesaria para definir una instancia de *Enzyme* y por lo tanto se pueden crear instancias de esta clase sin esta relación asociada. Si un recurso de entrada contiene las entidades *enzyme* y *disease*, congruentes a las clases de la ontología *Enzyme* y *Disorder* respectivamente, y la relación entre ellas *linkedDisease*, congruente a *hasAssociatedDisease*, se definirían las siguientes reglas de correspondencia:

```
Regla_clase(enzyme, Enzyme)
Regla_clase(disease, Disorder)
Regla_relación((enzyme, linkedDisease, disease),
(Enzyme, hasAssociatedDisease, Disorder))
```

El resultado de estas reglas de transformación es la creación de tantas instancias de *Enzyme* como instancias de *enzyme* existan en el repositorio de entrada. Para aquellas instancias de *enzyme* que no tenga una asociación con una enfermedad (*linkedDisease disease*), la regla de relación definida no obtendrá ningún dato y la transformación se realizará con normalidad.

Si la ausencia de datos provoca que la instancia se cree de forma incompleta de manera que no se puedan comprobar las reglas de identidad o el resultado sea inconsistente en el repositorio final, la instancia es descartada.

- **Inconsistencia de las fuentes con el modelo global:** debido a que el modelo de datos global se define de forma independiente a los recursos fuente, pueden aparecer diferencias entre los esquemas que dificultan la definición de reglas de transformación y el proceso de integración.

  En la ontología global de la figura 8.2, una proteína se define como *Polypeptide derivesFrom CDS* y a su vez se define *CDS partOf Transcript*, es decir, la relación entre las entidades *Polypeptide* y *Transcript* se realiza a través de la entidad *CDS*. Sin embargo, para los esquemas de los recursos de entrada de la figura 8.4, existe una relación directa entre la entidad *protein* (congruente a *Polypeptide*) y la entidad *transcript* (congruente a *Transcript*), en lugar de establecerse a través de una entidad congruente a *CDS*. No es posible definir en el modelo de salida la relación sin hacer uso de la instancia *CDS*, sin embargo, las instancias de *protein* y su relación con las instancias *transcript* pueden ser de bastante interés como para requerir su integración en el repositorio integrado final. La forma más sencilla de solucionar este problema es a través del uso de patrones. Para transformar las instancias *protein* del esquema *Mucor* se sustituye el patrón de la figura 8.4 por el siguiente patrón:

```
?protein:INDIVIDUAL,
?cds:INDIVIDUAL= create(?protein.RENDERING+"_CDS"),
?transcript:INDIVIDUAL
BEGIN
ADD ?protein instanceOf Polypeptide,
ADD ?protein derivesFrom ?cds,
ADD ?cds instanceOf CDS,
ADD ?cds partOf ?transcript,
ADD ?transcript instanceOf Transcript
END
```

Este patrón genera la variable *?cds* a partir de la variable *?protein*, por lo que no es necesario enlazar la variable *?cds*, por cada instancia única que se enlace a *?protein*, se generará automáticamente una instancia de *CDS*. Al enlazar las variables *?protein* y *?transcript* con entidades del recurso de entrada, las plantillas de reglas se instancian de la siguiente forma:

```
Plantilla_regla_clase(protein, ?protein)
Plantilla_regla_clase(transcript, ?transcript)
```

- **Inconsistencias entre las fuentes:** El problema de la inconsistencia entre las fuentes se produce cuando una misma instancia en el modelo global es creada a través de varias instancias en los recursos fuentes, las cuales tienes distintos valores de sus atributos y relaciones.

  Si los valores inconsistentes no coinciden con valores de identidad, se trata de un problema de redundancia, pues la identidad de las instancias se puede descubrir a través de las reglas de identidad y la redundancia se resuelve como se ha explicado anteriormente. En el ejemplo de genes ortólogos, podría ocurrir que para un mismo gen, dos recursos distintos den un listado diferentes de los grupos de ortólogos a los que pertenece, el resultado final en el repositorio integrado es que los grupos de ortólogos a los que pertenece la instancia de dicho gen es la unión de los conjuntos dados por cada uno de los recursos.

  Si los valores inconsistentes coinciden con valores de identidad, las reglas de identidad identifican las instancias como dos distintas, y ambas son creadas en el repositorio final. Sólo en el caso de que por la semántica definida en la ontología resultara en un repositorio inconsistente, la última instancia en ser creada sería descartada.

Capítulo 9

Gestión de la información biomédica

La consecución de la medicina traslacional requiere la adecuada gestión de la información biomédica y la consecución de la interoperabilidad semántica, que permitan un correcto uso y explotación de información. En las propuestas de gestión de la información biomédica, la gestión de información clínica está ligada a la gestión de datos clínicos definidos y representados haciendo uso de estándares de HCE, por lo tanto los métodos de gestión deben necesariamente incorporar arquetipos y terminologías clínicas.

En este capítulo propongo soluciones para la gestión de la información biomédica utilizando OWL como framework para la explotación conjunta de arquetipos ADL, ontologías y terminologías para la interoperabilidad semántica.

El modelo de integración expuesto en el capítulo anterior hace uso del modelo de transformación basada en el dominio. Este tipo de transformación permite realizar una integración independiente de la estructura de origen de representación, y una explotación del conocimiento contenido basada en un dominio específico. Sin embargo, en el ámbito de la gestión clínica, la gestión de los modelos clínicos que se basan en una arquitectura de dos niveles necesita otra orientación. Estos modelos se definen por medio de restricciones sobre un modelo de referencia, están pensados para su desarrollo de forma independiente y su compartición entre distintas instituciones y organizaciones que trabajan con estándares de HCE. Por lo tanto, es importante en este ámbito prestar atención a la estructura de definición de los modelos clínicos, pues su interés no viene dado por el conocimiento que pueden representar, sino como artefactos de recogida de información, que puede ser validado, comparado y reutilizado. Una transformación semántica basada en la estructura

de los modelos clínicos permitirá crear un repositorio de arquetipos donde se puedan validar la consistencia de estos, compararlos con el resto de modelos almacenados en el repositorio, y anotarlos con terminologías biomédicas de manera que se pueda hacer una explotación de arquetipos y terminologías utilizando un mismo formalismo de representación. Para otros usos de los modelos clínicos, como la explotación de datos clínicos basados en dichos modelos, la transformación basada en el dominio es aplicable.

9.1 Modelos clínicos en OWL

Para la gestión semántica de arquetipos utilizo dos tipos de representación OWL diferentes, tanto para ISO 13606 como para openEHR. Estas representaciones corresponden a las mencionadas en la sección 3.5 para openEHR e ISO 13606. Ambas tienen en común que el modelo de información es representado en OWL, pero difieren en el tipo de entidad OWL usada para representar los arquetipos:

- Los arquetipos se representan como individuos de la ontología del modelo de información correspondiente (de ISO 13606 u openEHR). Esta representación se utiliza para la transformación de arquetipos entre openEHR e ISO 13606 y para anotar los arquetipos con ontología y terminologías externas. A su vez, estas anotaciones son explotadas por otros métodos, como son la comparación y búsqueda de arquetipos.
- Los arquetipos se representan como clases OWL para realizar tareas que requieren realizar un razonamiento automático sobre el contenido, como validación de la corrección de la especialización de los arquetipos.

9.2 Datos clínicos en OWL

Tanto ISO 13606 como openEHR representan los extractos de datos de HCE en XML. Este formato tiene numerosas limitaciones para el procesado semántico de información y no proporcionan soporte para el razonamiento automático como lo hace OWL.

Para la transformación de extractos clínicos basados en estándares de HCE a OWL se utiliza el modelo de transformación presentado en la sección 7.1 para realizar una transformación dirigida por el dominio. Para ello se definen reglas de correspondencia entre los arquetipos utilizados para capturar los datos (modelo de entrada) y una arquitectura ontológica del dominio destino (modelo de salida).

El objetivo de esta transformación es el procesamiento semántico de la información clínica, utilizando facilidades proporcionadas por OWL, como el razonamiento automático. Por lo tanto, el modelo de salida utilizado en la transformación dependerá de la investigación a realizar sobre la información.

El modelo de transformación permite el uso de patrones de diseño de contenido ontológicos para facilitar la definición de reglas de correspondencia. Al tratarse de información clínica, los patrones definidos en el contexto del proyecto europeo SemanticHealthNet (SHN) (sección 3.2.3.2) podrían ser utilizados.

9.3 Métodos de gestión de la información clínica

9.3.1 Anotación semántica

Haciendo uso de la anotación semántica podemos enriquecer con metadatos semánticos tanto los arquetipos como los extractos de HCE. Estas anotaciones son proporcionadas por terminologías, ontologías y recursos semánticos externos del dominio biomédico.

Los arquetipos están ya enlazados a terminologías por medio de enlaces terminológicos (*terminological bindings*). Sin embargo, los enlaces terminológicos no están siempre definidos y, dependiendo del uso específico del arquetipo, puede ser necesario añadir más significado semántico, por ejemplo, para realizar una clasificación personalizada de pacientes. El método de anotación definido combina métodos manuales y semi-automáticos que facilitan esta tarea. Dado un repositorio específico de ontologías, vocabularios controlados y terminologías en formato OWL, el método de recomendación sugiere anotaciones basándose en el contenido textual del arquetipo, las cuales son términos de las entidades incluidas en el repositorio con coincidencia exacta o parcial con el contenido textual del arquetipo.

Por lo tanto, a los enlaces terminológicos del arquetipo se le unen las anotaciones añadidas por el usuario. El grupo de todas las anotaciones son una generalización representativa del conocimiento semántico contenido en el arquetipo y crea lo que se llama el perfil semántico del mismo (ver siguiente sección).

La anotación utiliza la representación de arquetipos como individuos OWL. Las anotaciones se representan en formato OWL, para que puedan ser explotadas conjuntamente con el contenido del arquetipo. Tanto estas anotaciones como los enlaces terminológicos son explotados por diferentes

tareas, como comparación de arquetipos y consulta. Los extractos clínicos de la HCE se anotan de forma indirecta, adquiriendo las anotaciones de los arquetipos utilizados para capturar los datos.

9.3.2 Perfiles semánticos

Un perfil semántico se define como la descripción semántica de un conjunto de datos, que constituye una interpretación semántica de los mismos [232]. Dicha interpretación semántica se utiliza, en lugar de toda la información disponible de la entidad, para un procesamiento eficiente y efectivo de los datos. Utilizar ontologías para la construcción del perfil permite tomar decisiones y hacer recomendaciones basadas en una especificación formal del conocimiento del dominio.

En los arquetipos y extractos clínicos, sus anotaciones, directas o indirectas, representan descripciones semánticas de las entidades y datos de los mismos, convirtiéndose en los elementos básicos para construir los perfiles semánticos. En términos de representación, un perfil semántico es representado como un conjunto de anotaciones semánticas.

La fórmula 9.1 muestra la definición formal del perfil semántico de un arquetipo, definido como el conjunto unión de todos sus enlaces terminológicos y sus anotaciones. En la fórmula T_i representa un concepto proveniente de una de las terminologías enlazada en el arquetipo, mientras que C_j representa un concepto proveniente de una terminología externa, utilizada en la tarea de anotación semántica.

$$Perfil\_semantico(A) \quad = \quad \{T_1, T_2, \ldots, T_n\} \ \cup \ \{C_1, C_2, \ldots, C_m\} \quad (9.1)$$

El perfil semántico de un extracto de HCE se obtiene desde dos fuentes, el arquetipo y los datos clínicos. La fórmula 9.2 muestra la definición formal de este perfil. Primero, el perfil semántico de los datos clínicos se obtiene del arquetipo utilizado para capturar los datos, por ejemplo, el perfil semántico de un extracto sobre presión sanguínea será el perfil semántico del arquetipo de la presión sanguínea. Segundo, los datos clínicos del extracto añaden información al perfil semántico, haciéndolo más preciso. Por ejemplo, en caso de tener un tensión baja, el perfil semántico podría incluir la anotación "hipotensión". La obtención de anotaciones adicionales a partir de los datos clínicos en el extracto se obtiene a partir de un proceso de clasificación. La transformación basada en el dominio se aplica sobre los datos clínicos, haciendo uso de una ontología de clasificación (S_i) como modelo de salida. La ontología de clasificación contiene reglas de clasificación que por medio

de razonamiento permitan obtener esas nuevas anotaciones, de esta manera, la función *"Clasificacion"* en la fórmula 9.2 devolverá anotaciones que serán conceptos de la ontología (S_i). La clasificación de datos clínicos se amplía en la sección 9.4.1.

$$Perfil\_semantico(E, A) = Perfil\_semantico(A) \cup Clasificacion(E, S_i) \tag{9.2}$$

9.3.3 Similitud semántica

La representación OWL de arquetipos y extractos de HCE y su anotación semántica con entidades de ontologías externas permite realizar una comparación semántica de los mismos. Estudiar la similitud entre arquetipos es útil porque el mismo concepto clínico pueden ser expresado de muchas formas diferentes usando el mismo o diferente modelo de referencia. Esto hace la interoperabilidad del contenido clínico y su intercambio mucho más difícil, por lo que encontrar similitudes puede ayudar a acercar diferentes representaciones.

En la figura 9.1 se muestra el diagrama del método de similitud, el cual recibe como entrada dos arquetipos a comparar y la salida del método es la puntuación de similitud en el rango [0,1]. El uso de este método en una arquitectura integrada permite recuperar los arquetipos desde un repositorio junto a su perfil semántico, que incluye las anotaciones semánticas y los enlaces terminológicos. Además, las ontologías del modelo de información y del modelo de arquetipos proporcionan el contexto semántico para comparar los arquetipos.

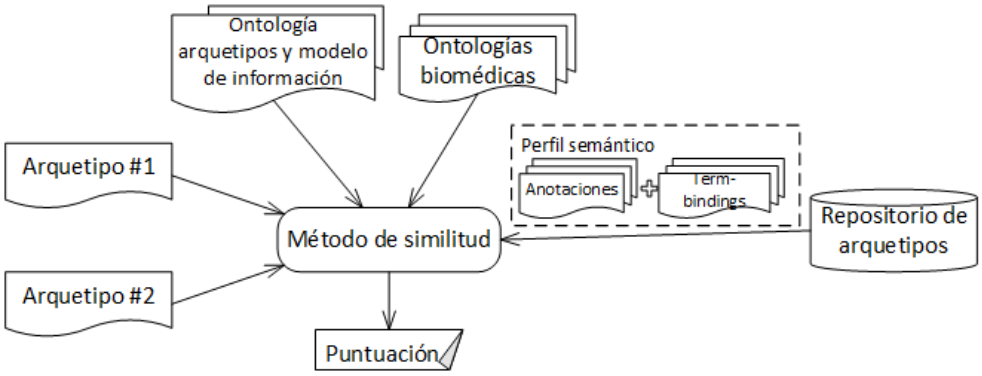

Figura 9.1: Diagrama del proceso de cálculo de similitud semántica

El método de similitud propuesto es basado en nodos, ya que explota los enlaces terminológicos, las anotaciones semánticas y la estructura jerárquica de las ontologías. El método compara todos los pares de elementos en el perfil semántico de los arquetipos, obteniendo una puntuación de similitud para cada par. A partir de este análisis de pares selecciona un subconjunto de los mismos aplicando los siguientes pasos:

1. Compara todos los pares y selecciona aquellos con puntuación mayor que un umbral dado.

2. Del conjunto de pares del paso 1 obtiene el subconjunto que incluye solo un par por elemento de perfil semántico de cada arquetipo y para el que la suma de puntuación de similitud de todos sus pares es máxima.

La función de similitud por pares para dos elementos del perfil semántico utiliza las siguientes similitudes:

- Similitud taxonómica (d): Mide la distancia jerárquica entre las clases asociadas con los dos elementos C_i y C_j, es decir, a través de enlaces taxonómicos. Esta función utiliza el conjunto unión de ancestros de cada clase y el conjunto de ancestros comunes de ambas clases. Por lo tanto, la puntuación se calcula como se muestra en la fórmula 9.3:

$$d(C_i, C_j) = 1 - \frac{|anc(C_i) \bigcup anc(C_j)| - |anc(C_i) \bigcap anc(C_j)|}{|anc(C_i) \bigcup anc(C_j)|} \tag{9.3}$$

  Nótese que las clases pueden presentar herencia múltiple, lo que implicaría diferentes caminos taxonómicos y, por lo tanto, diferente puntuación de similitud taxonómica. En ese caso, la distancia más corta es devuelta por la función.

- Similitud de propiedades (ps): Similitud entre el conjunto de propiedades asociadas con las clases asociadas con los dos elementos,calculada como se muestra en la fórmula 9.4:

$$ps(C_i, C_j) = \frac{|comun(C_i, C_j)|}{|comun(C_i, C_j)| + y_1 * |dif(C_i, C_j)| + y_2 * |dif(C_j, C_i)|} \tag{9.4}$$

  donde y_k se refiere al peso dado a cada una de las métricas, $0 \leq y_k \leq 1$, $\sum y_k = 1$.

- Similitud lingüística (ls): Un cálculo basado en la cadena de caracteres de los términos asociados con los elementos ontológicos comparados. Si se comparan dos conceptos de la representación OWL de dos arquetipos, este cálculo utiliza la definición de términos de ambos conceptos. Cuando se comparan dos conceptos de una terminología, utiliza las etiquetas o el nombre local de los conceptos comparados. La implementación actual utiliza la distancia de Levenshtein [233].

Combinando estas tres funciones, la similitud entre dos elementos del perfil semántico de los arquetipos es calculada como se muestra en la fórmula 9.5:

$$similitud\_pares(C_i, C_j) = w_1 * ls(C_i, C_j) + w_2 * ps(C_i, C_j) + w_3 * d(C_i, C_j) \quad (9.5)$$

donde w_k se refiere al peso dado a cada una de las métricas, $0 \leq w_k \leq 1$, $\sum w_k = 1$.

La suma de todas las puntuaciones de similitud por pares obtenida entre todos los pares de elementos seleccionados de los perfiles semánticos constituye la similitud del perfil semántico de los arquetipos como se muestra en la fórmula 9.6:

$$similitud\_perfil(A_1, A_2) = \sum similitud\_pares(C_i, C_j) \quad (9.6)$$

El método de similitud incluye otro factor que toma en cuenta el tipo estructural del arquetipo comparado en el contexto de la ontología del modelo de información. Este factor, similitud estructural, asume que dos arquetipos del mismo tipo COMPOSITION son más similares que dos arquetipos de tipos diferentes, por ejemplo, COMPOSITION y SECTION. Esta puntuación es obtenida al aplicar la función de similitud taxonómica a los tipos de ambos arquetipos. La similitud semántica final de dos arquetipos se obtiene según la fórmula 9.7:

$$similitud\_semantica(A_1, A_2) = z_1 * similitud\_estructural(A_1, A_2) + z_2 * similitud\_perfil(A_1, A_2) \quad (9.7)$$

donde z_k se refiere al peso dado a cada una de las métricas, $0 \leq z_k \leq 1$, $\sum z_k = 1$.

9.4 Uso secundario de la información biomédica

9.4.1 Clasificación de datos con razonamiento OWL

Una de las motivaciones para utilizar OWL en la representación de información biomédica es su capacidad para realizar razonamiento automático sólido y completo. Utilizando el axioma de condición necesaria y suficiente (`equivalentClass`, ver sección 3.1.2.3) se pueden definir criterios de inclusión/ exclusión de un individuo en una clase OWL, de manera que por medio del razonamiento se pueden clasificar automáticamente los datos clínicos en grupos con un interés concreto.

Aplicar razonamiento OWL sobre datos requiere tener un conjunto de datos basado en una ontología OWL. Dicha ontología debe proporcionar una descripción del conocimiento del dominio y las reglas de clasificación. El objetivo es aplicar razonamiento OWL para clasificar pacientes según los datos de HCE disponibles. Los pacientes se agruparán en diferentes categorías atendiendo a un criterio clínico concreto. Mientras que el conocimiento del dominio es modelado normalmente usando axiomas `subClassOf`, las reglas de clasificación se especifican usando axiomas `equivalentClass`. La ontología del dominio desarrollada se utiliza para transformar los datos a una representación semántica como se indica en la sección 9.2. Idealmente, las reglas de clasificación son implementadas en una ontología que reutiliza la ontología del dominio, lo que permitirá la explotación conjunta de reglas de clasificación, conocimiento del dominio y datos HCE utilizando razonamiento automático. Llamo a la ontología de reglas de clasificación una ontología de clasificación y contiene, por lo menos, una clase por grupo de interés.

Una vez que la ontología está lista, un razonador OWL-DL como Hermit [108] puede aplicarse sobre el conjunto semántico de datos completo para inferir toda la información posible dados los datos. El resultado de dicho proceso de inferencia podrían ser las clasificaciones resultantes, las cuales pueden ser recuperadas utilizando lenguajes de consulta semánticos como DL-query [234] o SPARQL, o a través de librerías de programación como OWLAPI [235].

El método de clasificación requiere que las categorías puedan ser especificadas en términos de reglas expresadas como clases definidas OWL DL (clases con condiciones necesarias y suficientes `equivalentClass`) y que los datos HCE están disponibles en OWL. Dado un paciente, el método de clasificación tiene dos entradas: la ontología OWL con las reglas de clasificación y la representación OWL de los datos HCE del paciente. Las reglas de clasifica-

ción se aplican a los datos utilizando razonamiento DL, que permite obtener las categorías a las que pertenece el paciente. Esta propuesta es genérica, y los mismos datos de pacientes pueden ser analizados automáticamente y clasificados aplicando ontologías de clasificación diferentes.

Por ejemplo, una regla podría establecer que un paciente tiene hipotensión cuando la presión sanguínea sistólica es menor que 90 mm Hg o la presión diastólica es menor que 60 mm Hg. Esta regla podría ser codificada en OWL-DL como sigue:

```
Hipotension equivalentClass
(Paciente and ((sistolica some integer[<= 90 ])
or (diastolica some integer[<= 60 ])))
```

Por lo tanto, si nuestro conjunto de datos contiene datos sobre pacientes con propiedades `sistolica value 80` o `diastolica value 50`, dichos pacientes serán clasificados como miembros de la clase `Hipotension`. Dicha clasificación es utilizada para enriquecer el perfil semántico del paciente, ya que pueden ser representadas como nuevas anotaciones asociadas con un extracto HCE dado.

9.4.2 Recomendación de recursos formativos

La disponibilidad de la información clínica contenida en la HCE en formato OWL permite su uso en métodos de recomendación de recursos para la formación de profesionales y ciudadanos. En esta sección se propone un método de recomendación de contenidos formativos directamente relacionados con la información clínica de los pacientes.

El método de recomendación de recursos clínicos se enmarca en una arquitectura integrada de datos clínicos proveniente de HCE, dónde recibe como entrada un usuario del sistema y devuelve como salida un conjunto de recursos formativos adecuados para ese usuario. El método puede recuperar la información clínica relacionada con el usuario y su perfil semántico asociado. El perfil semántico de un paciente es el conjunto de perfiles semánticos de sus extractos clínicos, aunque en este método, las recomendaciones se hacen a nivel de extracto clínico, en lugar de a nivel de HCE completa. El método hace uso de las anotaciones y los enlaces terminológicos para obtener un conjunto relevante de recursos formativos relacionados con el contenido semántico de la HCE del usuario.

El método de recomendación propuesto considera la existencia de uno o más repositorios de recursos formativos, formados por cualquier tipo de documento que será analizado y filtrado desde una perspectiva clínica. El único

requisito del método de recomendación es que los recursos estén enriquecidos con metadatos semánticos, es decir, anotados haciendo uso de ontologías y terminologías disponibles.

La recomendación de recursos formativos se hace en base a la información clínica proveniente de la HCE, por ello, el paciente se toma como usuario base del método para el calculo de la recomendación. Sin embargo, normalmente la HCE de un paciente está formada por varios extractos clínicos, comprendidos a lo largo del tiempo y que pueden corresponder a consultas clínicas no necesariamente relacionadas entre sí, por lo que la recomendación de recursos formativos se hace para un extracto clínico concreto de un paciente específico.

El método de recomendación utiliza el método de similitud comentado en la sección 9.3.3 para comparar el perfil semántico de un extracto clínico con los perfiles semánticos de los recursos de aprendizaje siguiendo los siguientes pasos:

1. Aplica la fórmula 9.6 tantas veces como recursos formativos disponibles, comparando un extracto clínico con cada recurso formativo. El perfil semántico de un extracto clínico se obtiene como se describe en la sección 9.3.2, mientras que el perfil semántico del recurso formativo se obtiene como el conjunto unión de todas sus anotaciones. El resultado de este paso es la asignación de una puntuación de similitud en el rango [0, 1] a cada recurso formativo para el extracto clínico seleccionado.

2. Se seleccionan aquellos recursos formativos cuya puntuación de similitud con el extracto clínico sea superior a un umbral de similitud preestablecido.

Los contenidos de los recursos de aprendizaje seleccionados como similares a un extracto clínico concreto pueden no ser adecuados al nivel de formación del usuario. Muchos recursos de aprendizaje están dirigidos a profesionales clínicos, como son publicaciones científicas y guías clínicas. Para evitar que este tipo de recursos llegue a los pacientes, el método de recomendación utiliza la puntuación de experto. La puntuación de experto es la puntuación que cada experto clínico le da a un recurso formativo, da un valor de su idoneidad para sus pacientes. Cada recursos formativo está formado por uno o varios documentos, y cada documento puede recibir una puntuación de idoneidad, de manera que la idoneidad final de un recurso formativo se calcula como se muestra en la fórmula 9.8:

$$idoneidad(R_i) = \frac{|\sum puntuacion\_experto(documentos(R_i))|}{|documentos(R_i)|} \tag{9.8}$$

Así, la puntuación de recomendación para un recurso formativo concreto final se calcula como indica la fórmula 9.9:

$$recomendacion(E_1, R_1) = z_1 * idoneidad(R_1) + z_2 * similitud\_perfil(E_1, R_1) \quad (9.9)$$

donde z_k se refiere al peso dado a cada una de las métricas, $0 \leq z_k \leq 1$, $\sum z_k = 1$.

Los recursos formativos seleccionados para un extracto clínico serán aquellos cuya puntuación de recomendación final esté por encima de un umbral seleccionado. La definición de umbrales permite personalizar el nivel de exigencia del método, más alto (umbral con valores cercanos a 1) o más bajo (umbral por debajo de 0,5).

Como el método de recomendación se hace partiendo de un extracto clínico de un paciente, las recomendaciones para los profesionales clínicos, que tienen asociados varios pacientes con sus HCEs, vienen dadas a partir de los extractos clínicos de sus pacientes. Así, un profesional médico recibe recomendaciones de contenidos formativos para cada uno de sus pacientes clínicos.

La efectividad de este método depende de su utilización en una arquitectura integrada donde además de extractos clínicos y sus arquetipos anotados utilizando ontologías y terminologías clínicas existan:

- Diferenciación de perfiles de usuario, entre paciente clínico y profesional médico.
- Acceso a un repositorio de recursos formativos anotados con terminologías clínicas y ontologías.

9.4.3 Aplicación de indicadores de calidad de la atención sanitaria

El grupo de informática médica del Academic Medical Center de Amsterdam (AMC) ha desarrollado una metodología, CLIF [236], para la formalización de indicadores de calidad de la atención sanitaria. Un indicador de calidad se define como un elemento medible del rendimiento de una práctica para el que hay evidencia o consenso de que puede ser usado para evaluar la calidad de la atención dada. Un ejemplo de indicador se puede ver en la tabla 9.1.

Para solucionar los problemas de ambigüedad creados por la redacción en lenguaje natural de los indicadores, la metodología CLIF está compuesta de 8 pasos para formalizar dichos indicadores de calidad.

- Paso 1. Codificación de conceptos relevantes de los indicadores por conceptos de una terminología: Los conceptos relevantes del texto del indi-

Tabla 9.1: Ejemplo de indicador de calidad

Indicador	Número de nodos linfáticos examinados después de una extirpación
Numerador	Número de pacientes a los que se les examinaron 10 o más nodos linfáticos después de la extirpación de un carcinoma primario de colon
Denominador	Número de pacientes a los que se les examinaron nodos linfáticos después de la extirpación de un carcinoma primario de colon
Criterios de exclusión	Haber recibido radioterapia con anterioridad o padecer de carcinomas de colon recurrentes

cador se anotan con términos provenientes de una terminología clínica. En el ejemplo de la tabla 9.1, los conceptos relevantes serían: "nodo linfático", "extirpación", "carcinoma primario de colon", "radioterapia" y "carcinoma de colon recurrente". Estos conceptos se pueden asociar con conceptos relacionados de una terminología como SNOMED-CT.

- Paso 2. Definición del modelo de información: El método CLIF requiere obtener una definición de cómo los datos clínicos están representados.

- Paso 3. Formalización de restricciones temporales: Cualquier evento en el texto del indicador que tenga una restricción temporal con respecto a una fecha específica o con respecto a otro evento ha de formalizarse haciendo uso de los campos del modelo de información. Por ejemplo, en el caso del indicador de la tabla 9.1 habría que formalizar que el examen de nodos linfáticos se hace después de una extirpación de carcinoma primario de colon.

- Pasos 4, 5 y 6. Formalización de restricciones numéricas, formalización de restricciones textuales y formalización de restricciones booleanas: Cualquier evento en el texto del indicador que tenga una restricción de número, de texto, o de verdadero/falso debe formalizarse haciendo uso de los campos del modelo de información. Por ejemplo, el indicador de la tabla 9.1 tiene una restricción numérica de que se deben examinar 10 o más nodos linfáticos.

- Paso 7. Identificación de criterios de exclusión: Los indicadores suelen especificar casos en los que no se debe aplicar el indicador. El método exige que se especifiquen esos criterios concretos.

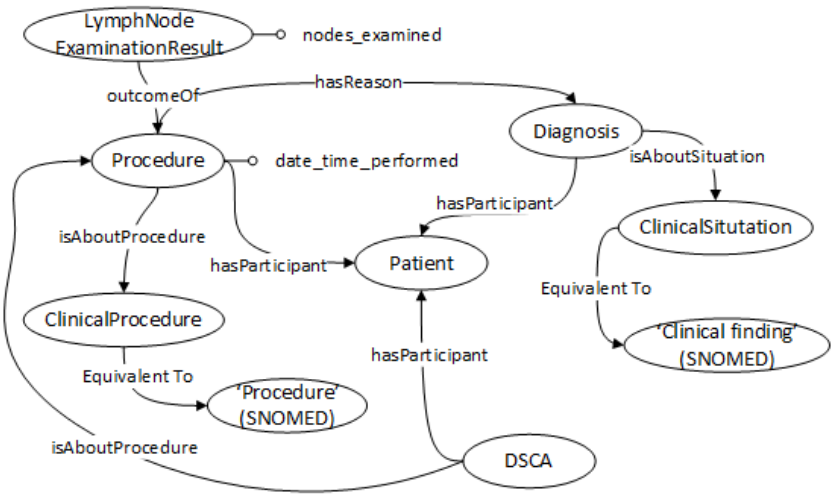

Figura 9.2: Diagrama de una ontología en el dominio de la diagnosis y los procedimientos

- Paso 8. Identificación de restricciones que solo afectan al numerador: El método exige que se especifiquen los criterios que solo afectan al numerador. En el ejemplo de la tabla 9.1, la restricción numérica de examinar 10 o más nodos linfáticos solo se aplica al numerador.

Estos ocho pasos proporcionan a la metodología CLIF la información suficiente para construir una consulta. La ejecución de la consulta sobre un recurso de datos clínicos que siga el modelo de información definido en el paso 2 de la metodología permite obtener los datos necesarios para calcular el indicador de calidad.

La aplicación de esta metodología a datos provenientes de extractos clínicos y representados en OWL es fácilmente aplicable. Por ejemplo, teniendo como modelo de información la ontología cuyo diagrama aparece en la figura 9.2:

- Paso 1. Este paso depende del texto del indicador, no del modelo seguido por los datos clínicos. Sin embargo, para que la metodología sea capaz de recuperar los datos, las anotaciones utilizadas para codificar los conceptos relevantes del texto deben formar parte de las terminologías utilizadas en el perfil semántico del extracto clínico o en la ontología del dominio utilizada para la representación OWL. En el ejemplo de la figura 9.2, la ontología enlaza con la terminología SNOMED-CT, por

lo que el indicador de la tabla 9.1 se codifica según se muestra en la tabla 9.2.

Tabla 9.2: Codificación de conceptos del indicador de calidad

Concepto del indicador de calidad	**Término de SNOMED-CT**
extirpación de colon (colon resection)	SCT_23968004 (Colectomy (procedure))
carcinoma primario de colon (primary colon carcinoma)	SCT_93761005 (Primary malignant neoplasm of colon)
examen de nodos linfáticos (lymph nodes examination)	SCT_284427004 (Examination of lymph nodes)
radioterapia (radiotherapy)	SCT_108290001 (Radiotherapy)
carcinomas de colon recurrentes (recurrent colon carcinoma)	SCT_314965007 (Local recurrence of malignant tumor of colon)

- Paso 2. El modelo de información es la ontología usada como modelo de salida y, opcionalmente, patrones de diseño de contenido. Por lo tanto este paso se hace guiado por la ontología y los patrones. En concreto, se define una variable por tipo de instancia a consultar y las relaciones entre las variables en el modelo de información. Por ejemplo, para la tabla 9.1 con el modelo de la figura 9.2, algunas de las variables definidas son *?resection, ?examination* y *?procedure* asociadas a la clase *Procedure*, la variable *?coloncarcinoma* asociada a la clase *Diagnosis*, la variable *?resultExamination* asociada a la clase *LymphNodeExaminationResult* y la asociación *?resection hasReason ?coloncarcinoma*, indicando que el procedimiento de extirpación (*?resection:Procedure*) se debe al diagnóstico de un carcinoma primario de colon (*?coloncarcinoma:Diagnosis*).

- Paso 3. La restricción temporal se formaliza con un `OWL:dataProperty` de la ontología que hace de modelo de información. Para el ejemplo, se formaliza que *?examination:Procedure.data_time_performed* $\geq$ *?resection:Procedure.data_time_performed*, indicando que el examen de nodos linfáticos se hace después de la extirpación.

- Pasos 4, 5 y 6. Como en el caso anterior, se hace uso de propiedades `OWL:dataProperty` de la ontología para formalizar las restricciones numéricas, textuales y booleanas. Para el ejemplo se formaliza que *?resultExamination.nodes_examined* $\geq$ *10*, indicando que el resultado del examen de nodos linfáticos tiene 10 o más nodos examinados.

```
PREFIX sct:<http://www.ihtsdo.org/>
PREFIX clif:<http://sele.inf.um.es/ontologies/clif/colorctal-domain#>
SELECT DISTINCT ?patient
WHERE{
    ?patient a clif:Patient .
    ?resultExamination a clif:LymphNodeExaminationResult .
    ?examination a clif:Procedure .
    ?resection a clif:Procedure .
    ?coloncarcinoma a clif:Diagnosis .

    ?resection clif:hasReason ?coloncarcinoma .
    ?examination clif:hasParticipant ?patient .
    ?coloncarcinoma clif:hasParticipant ?patient .
    ?examination clif:hasReason ?coloncarcinoma .
    ?resultExamination clif:hasParticipant ?patient .
    ?resultExamination clif:outcomeOf ?examination .

    ?resection clif:isAboutProcedure ?23968004 .
    ?23968004 a sct:SCT_23968004 .
    ?coloncarcinoma clif:isAboutSituation ?93761005 .
    ?93761005 a sct:SCT_93761005 .
    ?examination clif:isAboutProcedure ?284427004 .
    ?284427004 a sct:SCT_284427004 .

    ?resultExamination clif:nodes_examined ?num15
    FILTER ( ?num15 >= 10)

    ?examination clif:date_time_performed ?tempRel120 .
    ?resection clif:date_time_performed ?tempRel220 .
    FILTER (xsd:dateTime(?tempRel120) >= xsd:dateTime(?tempRel220))
}
```

Figura 9.3: Consulta SPARQL generada para el numerador del indicador de calidad ejemplo

- Pasos 7 y 8. Esos pasos dependen del texto del indicador, por lo que no requieren ninguna consideración adicional.

El resultado de los ocho pasos es la formalización del indicador en una consulta SPARQL o DL-query sobre la representación OWL de los datos clínicos. La figura 9.3 muestra parte de la consulta generada para el numerador del ejemplo de la tabla 9.1. La consulta completa tendría más filtros para indicar los conceptos de exclusión (pacientes que han recibido radioterapia y con cáncer de colon recurrente diagnosticado). En las dos primeras secciones de la consulta se define el modelo de información. En la tercera sección se codifican los conceptos con términos de SNOMED-CT. La siguiente sección es el filtro que indica que se deben haber examinado 10 o más nodos linfáticos y por último aparece el filtro que establece que el procedimiento de examen se debe haber realizado después de un procedimiento de extirpación.

9.5 Herramienta de gestión de información biomédica

La plataforma ArchMS [237] es una herramienta prototípica que integra todos los métodos de transformación, integración, gestión y explotación de información biomédica expuestos en esta tesis. ArchMS se enfoca en la gestión de arquetipos y datos clínicos. Por medio de la aplicación de los diferentes métodos expuestos facilita la integración de arquetipos y datos clínicos con recursos externos y facilita la explotación y reutilización de datos en múltiples escenarios aplicando tecnologías de la Web Semántica. La figura 9.4 muestra una vista de la arquitectura de la herramienta. Una característica de ArchMS es que su repositorio de recursos semánticos contiene tanto recursos locales, incluyendo ontologías de HCE y ontologías locales cargadas por los usuarios, como recursos externos, en este caso, ontologías de BioPortal. Dichos recursos semánticos son ontologías, terminologías y vocabularios controlados disponibles en formato OWL, que se utilizan para la tarea de anotación de arquetipos, similitud, transformación de datos, creación de perfil semántico, clasificación y explotación secundaria de los datos. A continuación describo la funcionalidad principal de ArchMS con respecto a su uso con arquetipos y datos clínicos.

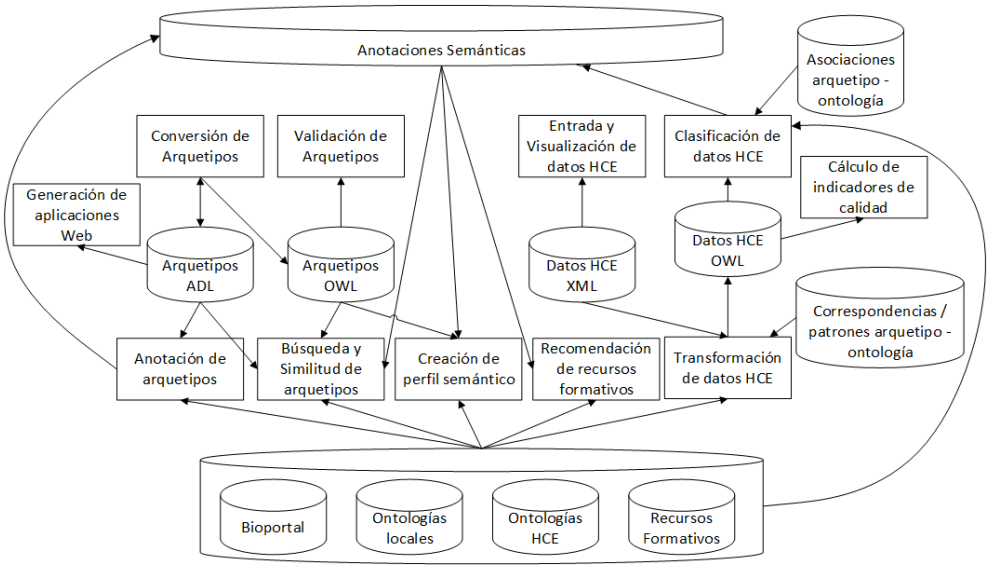

Figura 9.4: Arquitectura de ArchMS

9.5.1 Funcionalidad de arquetipos

Las principales actividades que pueden ejecutarse sobre los arquetipos son: conversión, validación, anotación, búsqueda de arquetipos similares o de arquetipos con propiedades concretas, y generación de aplicaciones.

- Gestión de arquetipos: el sistema permite importar arquetipos ADL para la representación openEHR e ISO 13606. Incluye la funcionalidad proporcionada por las herramientas previamente desarrolladas, Archeck [175] y PoseacleConverter [173]. La primera de ellas permite comprobar la consistencia de un arquetipo especializado con respecto a sus padres. PoseacleConverter permite transformar arquetipos desde openEHR a ISO 13606 y viceversa y representarlos en OWL. Se utilizan diferentes propuestas para proporcionar persistencia a información relacionada con arquetipos. Por un lado, ArchMS utiliza una base de datos relacional MySQL para almacenar propiedades de los arquetipos como nombre, lenguaje, propósito, fichero ADL, etc. Básicamente, este repositorio contiene el contenido no semántico del arquetipo. Por otro lado, ArchMS utiliza un repositorio semántico implementado utilizando Jena [238] para almacenar la representación OWL de los arquetipos, permitiendo la formulación de consultas SPARQL. Para acelerar consultas se obtiene un índice Lucene [239] de arquetipos analizando las secciones de ontología y arquetipos. Cuando un arquetipo ADL se importa al sistema, se realizan las siguientes actividades:

  1. Comprobar la corrección usando Archeck, lo que requiere utilizar la representación del arquetipo basada en clases OWL.
  2. Si el arquetipo no es correcto, los siguientes pasos no se ejecutan.
  3. Almacenar la representación OWL del arquetipo en el repositorio de arquetipos OWL.
  4. Almacenar el contenido ADL en el repositorio de arquetipos ADL y crear el índice Lucene.
  5. Aplicar PoseacleConverter para obtener la representación OWL basada en individuos del arquetipo. Si la transformación es correcta, se almacena el contenido del arquetipo OWL en el repositorio.

  ArchMS integra como servicio la herramienta ArchForms [240], que genera aplicaciones web a partir de arquetipos. En ArchMS, el administrador puede generar aplicaciones ArchForms y hacerlas disponibles para el resto de usuarios para descarga y mayor desarrollo.

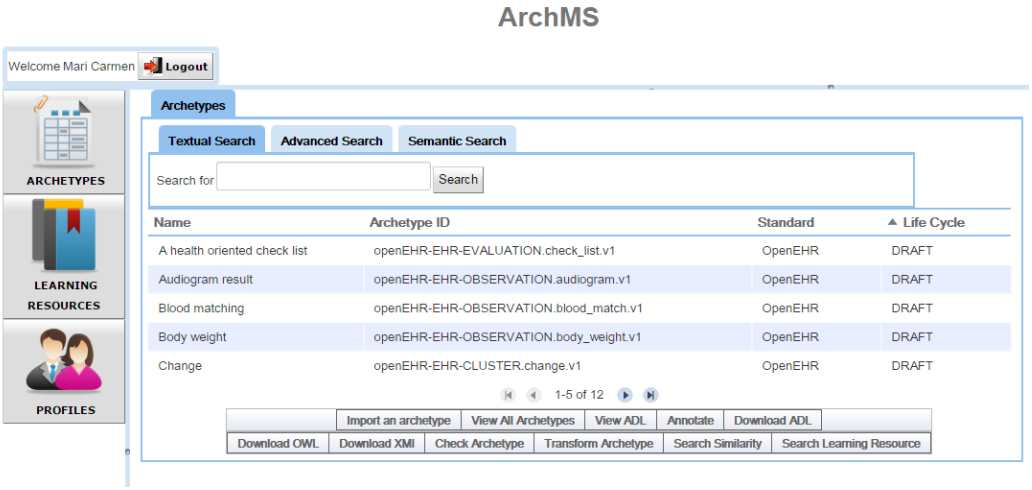

Figura 9.5: Interfaz de gestión de arquetipos de ArchMS

El usuario puede descargar el arquetipo en formatos ADL y OWL y ejecutar, en cualquier momento, la validación del arquetipo y la transformación del arquetipo desde ISO 13606 a openEHR y viceversa. La figura 9.5 muestra la interfaz principal de gestión de arquetipos.

- Anotación de arquetipos: ArchMS implementa métodos de anotación manual y semi-automática para la anotación de arquetipos. BioPortal, que contiene más de 400 ontologías, terminologías y vocabularios controlados pertenecientes al dominio biomédico, es la fuente principal de recursos de anotación para los usuarios. Para obtener términos de recomendación desde BioPortal, ArchsMS utiliza los servicios web de BioPortal, que proporcionan términos candidatos para un contenido textual dado. Además de estos servicios, utiliza un método de búsqueda basado en Lucene para obtener coincidencias parciales o totales entre el texto y el contenido de los recursos semánticos locales. Las anotaciones del arquetipo confirmadas o seleccionadas por el usuario se almacenan en el repositorio de anotaciones semánticas. La acción de anotar también puede producirse cuando un arquetipo se importa correctamente, ya que ArchMS sugiere anotaciones potenciales de los recursos semánticos procesando el contenido textual del arquetipo. La figura 9.6 muestra la interfaz de anotación de ArchMS. En la imagen se está anotando el arquetipo `OBSERVATION.audiogram.v1`, que ya está anotado con la terminología SNOMED-CT. A la izquierda aparecen términos que pertenecen al arquetipo, y se sugieren posibles anotacio-

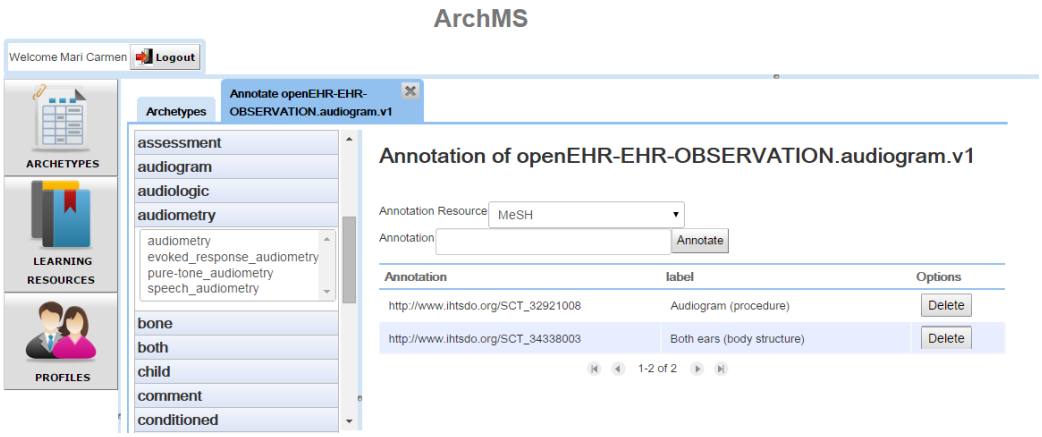

Figura 9.6: Interfaz de anotación de ArchMS

nes asociadas a dichos términos. En el ejemplo, vemos las sugerencias de anotaciones de la terminología MeSH a partir del concepto "audiometry".

- Búsqueda y similitud de arquetipos: las distintas opciones de consulta explotan los repositorios semánticos y relacionales.

    1. La interfaz de búsqueda textual utiliza el índice Lucene para encontrar arquetipos que contienen descripciones textuales que coincidan con la descripción textual de la consulta. Devuelve los arquetipos que contienen el texto buscado en su contenido textual, incluyendo las propiedades como nombre, lenguaje, propósito, etc. La figura 9.7 muestra el resultado de buscar arquetipos relacionados con el concepto "histopathology" utilizando esta interfaz de búsqueda.
    2. La búsqueda avanzada explota la base de datos relacional para encontrar arquetipos según propiedades y anotaciones, en caso de estar disponibles. Esto puede ser considerado una búsqueda detallada, ya que permite encontrar arquetipos por su contenido original (propiedades como lenguaje, arquetipo, nombre) y por metadatos añadidos posteriormente (anotaciones).
    3. La búsqueda semántica explota la representación de arquetipos como individuos OWL de dos formas diferentes. Por un lado, ejecuta una consulta SPARQL contra el repositorio semántico de arquetipos. Por otro lado, permite explotar la estructura jerárquica de las

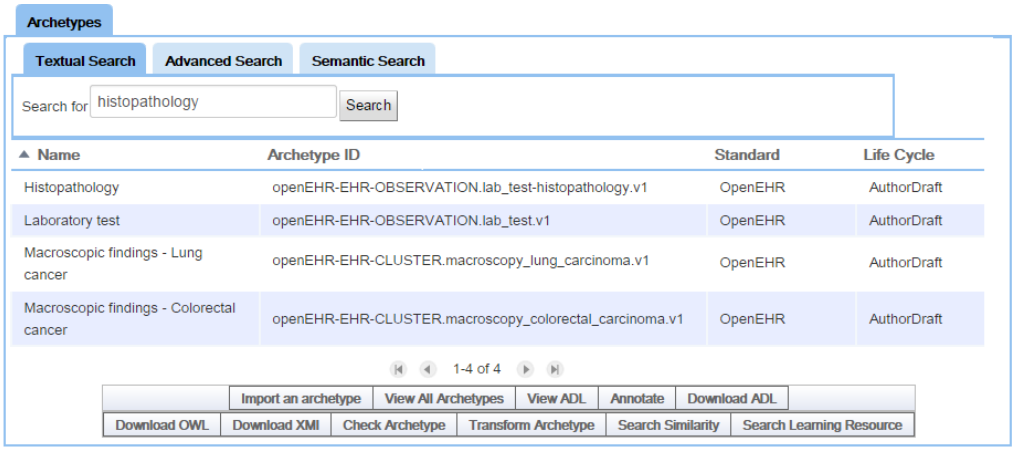

Figura 9.7: Interfaz de búsqueda textual de ArchMS

ontologías, terminologías y vocabulario controlado, aplicando medidas de similitud semántica para recuperar arquetipos similares a los que se utilizan para realizar la búsqueda.

El método de búsqueda semántica de consultas SPARQL está basado en el trabajo presentado en [241] y permite formular consultas seleccionando las entidades correspondientes de las ontologías contenidas en el repositorio, y que son traducidas automáticamente a SPARQL. Por lo tanto, proporcionamos un lenguaje de consulta específico para arquetipos basado en SPARQL.

La búsqueda de arquetipos similares implementa la función de similitud semántica descrita en la sección 9.3.3. En ArchMS se utiliza esta función en dos situaciones diferentes: (1) cuando un usuario decide buscar arquetipos similares a uno dado, la interfaz le permite especificar un umbral de similitud y los pesos, dicho umbral puede tomar automáticamente el valor 1 en el método de comparación estricto, que significa total equivalencia; (2) cuando un arquetipo se importa al sistema correctamente, ArchMS busca los arquetipos similares usando el umbral de similitud por defecto, dicha búsqueda tiene como objetivo comprobar si un arquetipo equivalente existe en el repositorio y recomendar anotaciones asociadas con arquetipos similares. La figura 9.8 muestra un ejemplo de búsqueda semántica.

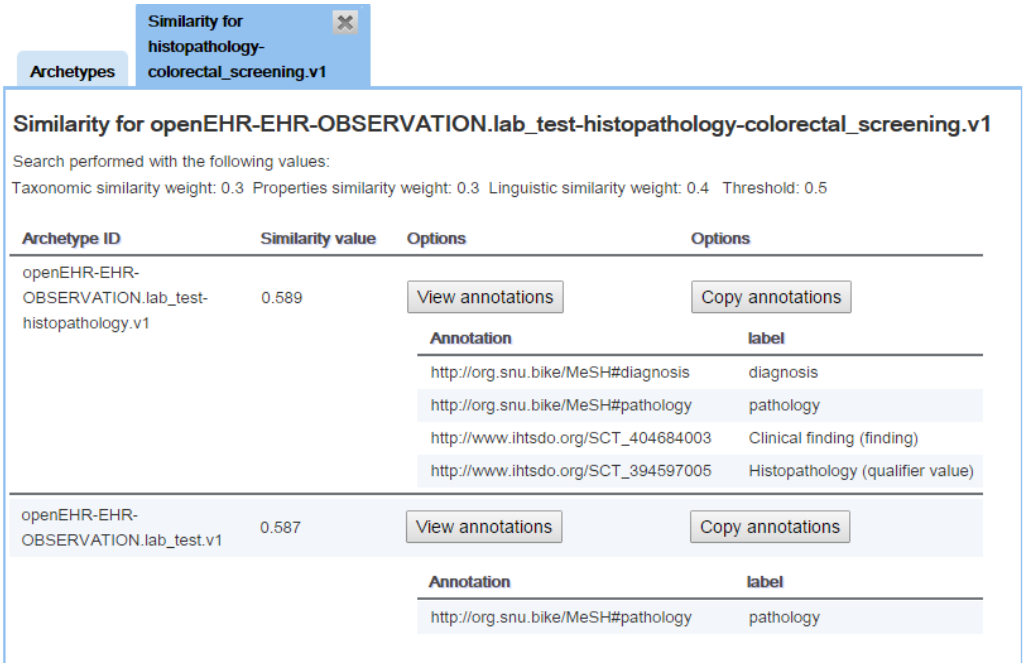

Figura 9.8: Interfaz de búsqueda semántica de ArchMS

9.5.2 Funcionalidad de datos

Las actividades principales que pueden realizarse con datos HCE son la obtención de una representación OWL a partir de extractos XML, visualizar datos HCE y cargar nuevos datos HCE, obtener el perfil semántico de extractos HCE, clasificar datos HCE, recomendar recursos formativos y evaluar la calidad de la asistencia sanitaria.

- Gestión de datos: ArchMS procesa extractos HCE en XML para las especificaciones openEHR e ISO 13606 que deben ser importadas en ArchMS. Las aplicaciones ArchForms permiten capturar datos según los arquetipos usados para crear la aplicación ArchForms y exportar dichos datos como extractos XML. ArchMS acepta la importación de extractos provenientes tanto de aplicaciones ArchForms, como aquellos generados con cualquier otro sistema de gestión de arquetipos ISO 13606 u openEHR. ArchMS proporciona una visualización simple del contenido del extracto que ha sido reutilizada de la interfaz usada en las aplicaciones ArchForms. Dicha opción está disponible una vez que el extracto ha sido correctamente importado.

- Transformación de datos y creación de perfil: ArchMS permite realizar

actividades semánticas en datos HCE transformando extractos HCE en OWL. Esta función utiliza las correspondencias entre los arquetipos y las ontologías, que han sido previamente cargadas a ArchMS. Dichas correspondencias se crean utilizando la herramienta SWIT, que implementa el método descrito en la sección 7.1. Los servicios de SWIT se invocan desde ArchMS para ejecutar automáticamente la transformación de datos una vez que el extracto ha sido importado a ArchMS. El contenido correspondiente en OWL está almacenado en el repositorio de datos OWL HCE. Una vez transformado a OWL, el perfil semántico de los datos HCE es extraído automáticamente. Dicho perfil es actualizado cuando: (1) las anotaciones asociadas con los arquetipos usados para capturar los datos cambian; (2) nuevas clasificaciones de los datos están disponibles.

- Clasificación de datos: ArchMS permite clasificar datos HCE según las reglas que definen el estado clínico de los pacientes. Para este propósito, el usuario debe seleccionar la ontología de clasificación. Para ser efectivo, esta ontología de clasificación debe haber sido implementada en OWL-DL y utilizar axiomas *equivalentClass* para describir los grupos de clasificación de los pacientes. Este requisito se debe al uso de razonamiento OWL-DL en ArchMS. Más concretamente, la implementación actual utiliza Hermit como razonador. ArchMS permite asociar ontologías de clasificación a arquetipos, lo que significa que cada vez que datos HCE son capturados con dichos arquetipos, dichas ontologías pueden ser utilizadas. Existe una relación muchos a muchos entre los arquetipos y las ontologías de clasificación, por lo tanto, cuando se selecciona la opción de clasificación para un paciente, se muestran todos los grupos para los que el paciente pertenece, es decir, los grupos según todas las ontologías de clasificación asociadas con el arquetipo utilizado para capturar los datos. ArchMS almacena dichas clasificaciones como anotaciones de datos EHR en el repositorio de anotaciones semánticas.

- Recomendación de recursos formativos: ArchMS permite recomendar recursos educativos a los usuarios de la plataforma en base a los datos HCE. ArchMS utiliza dos fuentes diferentes de recursos educativos: PubMed, una base de datos de literatura biomédica y un repositorio de contenidos de aprendizaje en formato Sharable Content Object Reference Model (SCORM) [242]. Los recursos PubMed están anotados utilizando la terminología biomédica MeSH, por lo que dichas anotaciones pueden ser utilizadas para construir el perfil semántico de las publicaciones extraídas de PubMed. En el caso de los contenidos en

formato SCORM, ArchMS utiliza este formato por tratarse de un estándar muy utilizado para el intercambio de contenidos de aprendizaje. El repositorio SCORM es gestionado por el administrador del sistema y dichos recursos pueden ser anotados con ontologías biomédicas por los profesionales clínicos. El sistema de recomendación aplica el método de similitud para comparar el perfil semántico de un extracto clínico con el perfil semántico de un recursos educativo y decidir su selección como contenido recomendado. Los expertos clínicos reciben recomendaciones de contenidos dependiendo de los extractos clínicos de sus pacientes. Los recursos provenientes de PubMed están normalmente dirigidos a usuarios con un perfil de formación alto, por lo que la interfaz permite a los expertos clínicos aplicar una puntuación que valora como de idóneos son esos recursos para sus pacientes. Los pacientes clínicos reciben recomendaciones de recursos formativos para cada uno de sus extractos clínicos. Estas recomendaciones dependen de la puntuación de similitud entre los perfiles del extracto y del recurso educativo, de la puntuación de idoneidad dada al recurso y de los pesos y umbral establecido. La interfaz permite al paciente varias los valores de peso y umbral.

- Evaluación de la calidad de la asistencia sanitaria: ArchMS acepta consultas SPARQL que calculan un indicador de calidad. Las consulta son el resultado de la formalización de un indicador con la metodología CLIF.

9.5.3 Usuarios

ArchMS maneja tres tipos de usuarios en ArchMS, administrador, usuario y médico.

El administrador está a cargo de mantener el sistema y es responsable de tareas particulares, como asignar el rol de médico a los usuarios correspondientes o generar las aplicaciones ArchForms. El administrador también está a cargo de definir un perfil semántico de los arquetipos, anotarlos usando directamente terminologías biomédicas o recomendaciones de otros arquetipos, cargar las correspondencias entre arquetipos y ontologías y añadir o eliminar nuevos recursos formativos al repositorio SCORM. El resultado de este proceso es un conjunto de repositorios de arquetipos que pueden ser explotados semánticamente por los usuarios y un repositorio de recursos formativos SCORM que pueden ser anotados por los médicos.

Los pacientes tienen acceso a sus datos clínicos y conocimiento adicional basado en su perfil, dependiendo de recursos de clasificación, además, pueden obtener cursos y documentos de formación relacionada con sus historias

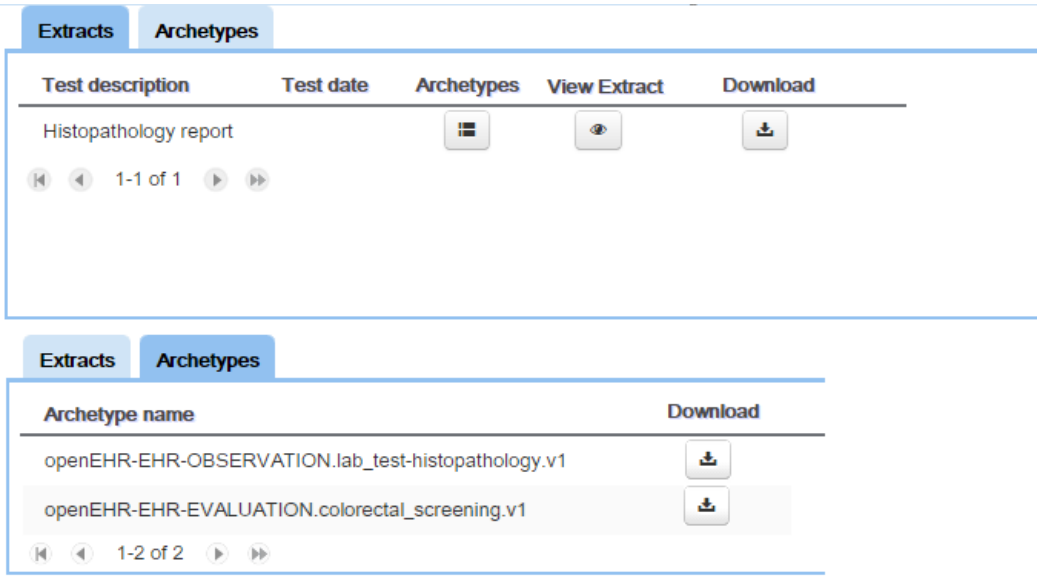

Figura 9.9: Interfaz de ArchMS para un paciente

clínicas. La figura 9.9 muestra la vista de un paciente de su informe clínico y los arquetipos asociados.

Los médicos pueden acceder a las historias clínicas de sus pacientes, obtener conocimiento adicional de los datos clínicos de los pacientes y usar procesos avanzados para transformar los datos clínicos y crear repositorios adecuados a su investigación. También pueden obtener recomendaciones de guías clínicas, cursos y publicaciones científicas relacionadas con sus casos clínicos en curso, además de evaluar dichos contenidos como adecuados o no para el aprendizaje de sus pacientes.

Capítulo 10

Escenarios de validación

Las distintas soluciones presentadas en esta tesis han sido utilizadas en varios escenarios de validación de forma conjunta o por separada y con distintos propósitos. En este capítulo se exponen los distintos dominios y los procesos llevados a cabo.

10.1 Programa de cribado de cáncer de colon y recto

En este escenario se han aplicado los métodos expuestos en esta tesis a un estudio sobre cáncer de colon y recto [243]. El estudio utiliza más de 20,000 registros de pacientes del programa de cribado de cáncer de colon y recto de la Región de Murcia. El objetivo del estudio es clasificar los pacientes en grupos de riesgo aplicando las guías clínicas de cáncer de colon y recto europea [244] y americana [245] y comprobar si existen discrepancias con respecto a la clasificación en la base de datos de origen, realizada por profesionales médicos. La guía europea define tres niveles de riesgo para los pacientes, bajo, intermedio y alto; mientras que la americana define solamente bajo y alto. En ambas guías clínicas, los datos que se utilizan para realizar la clasificación corresponden al número, tipo y tamaño de los adenomas encontrados durante una prueba de cribado, donde el tipo se define a partir del patrón de crecimiento (tubular, serrado, velloso, etc), grado de displasia, y si crece de forma plana (sésil) o no. Como el estudio trabaja con el número de adenomas, el primer paso normaliza los registros de la base de datos y realiza el recuento de adenomas. Para ello se seleccionan una serie de arquetipos openEHR adecuados para la representación de los datos normalizados. Teniendo los datos normalizados disponibles en extractos XML según los arquetipos ope-

nEHR, el siguiente paso utiliza SWIT para la transformación de los extractos a una representación semántica según una ontología OWL del dominio. Esta transformación es necesaria para realizar las tareas de clasificación utilizando inferencia, tareas que no pueden ser realizadas en la representación XML debido a sus limitaciones a la hora de expresar el conocimiento del dominio. Finalmente, las categorías de clasificación de cada uno de los protocolos son creadas en una ontología de clasificación OWL que utiliza la representación como instancias OWL de la ontología del dominio para realizar la clasificación de registros por medio de razonamiento OWL. En esta ontología de clasificación, cada categoría de clasificación es una clase definida donde las reglas de clasificación se especifican usando axiomas `equivalentClass`.

En el estudio se realizó la evaluación de los resultados de clasificación para una selección de 503 registros. El resultado de la comparación de las clasificaciones puede verse en la tabla 10.1. 8 de los registros seleccionados no devolvieron ninguna clasificación por falta de datos. Para el resto, existe una coincidencia en el 64.4 % de los casos. Entre las discrepancias (35.6 % de los casos), 58 corresponden a registros clasificados como alto riesgo por los especialistas y riesgo intermedio por la clasificación OWL. Esto se debe a que los especialistas tienden a asignar un nivel más alto de riesgo en comparación con el protocolo estándar, por lo tanto, las discrepancias no significan necesariamente errores de clasificación, pues los especialistas no tienen que seguir los protocolos necesariamente. Una selección de 17 casos discrepantes fueron presentados a un especialista médico, que determinó que la clasificación OWL era correcta en el 100 % de los casos.

Tabla 10.1: Comparación de resultados de clasificación

	Clasificación en la base de datos		
	Riesgo alto	**Riesgo intermedio**	**Riesgo bajo**
Clasificación OWL			
Riesgo alto	69	5	2
Riesgo intermedio	102	44	24
Riesgo bajo	26	17	206

La tabla 10.2 muestra la media y la media del tiempo en milisegundos para realizar la transformación a representación OWL de los 503 registros y para realizar la clasificación de los mismos utilizando el razonador Hermit 1.3.7 y Protégé 4.2.0. T1 es el tiempo de creación del razonador, T2 es el tiempo de validación del contenido OWL, T3 es el tiempo de clasificación

OWL y Total se refiere a la suma de T1, T2 y T3.

Tabla 10.2: Media y mediana del tiempo en milisegundos para realizar la transformación a representación OWL y obtener la clasificación de los registros

	Tiempo de transformación	**Tiempo de razonamiento y clasificación**			
		T1	**T2**	**T3**	**Total**
Media	150,97	26,47	282,84	1861,77	2171,09
Mediana	68	22	60	249	376

En las siguientes secciones se detallan los pasos seguidos en este escenario y se muestran algunos ejemplos de uso de las funciones de gestión de información biomédica

10.1.1 Identificación de necesidades y selección de arquetipos

La definición de grupos de pacientes según el riesgo de desarrollar cáncer de colon y recto varía entre las guías clínicas europea y americana. La tabla 10.3 muestra los criterios de clasificación para cada guía, donde un adenoma normal se define como aquel adenoma con un nivel bajo de displasia, un tamaño menor de 10 mm y que es o bien tubular (sésil o no) o bien serrado y sésil, mientras que un adenoma avanzado se define como aquel adenoma que o bien es velloso con un tamaño mayor o igual a 10 mm, o bien es serrado y no sésil.

En el estudio, la fuente de datos fue una base de datos relacional con todos los datos sobre pruebas de cribado de cáncer de colon y recto. Estos datos se transformaron a extractos HCE en formato XML según el estándar openEHR. Dicha transformación requiere una serie de arquetipos que modelen los datos, por lo tanto, el primer paso fue encontrar arquetipos que registren información sobre histopatologías y cribado de cáncer de colon y recto. El entorno ArchMS ofrece herramientas que permiten buscar en su repositorio de arquetipos atendiendo a diversos criterios. Una de los buscadores ofrece una búsqueda textual que explota el contenido textual (campos *keywords* y *ontology*) del arquetipo. Como se requiere modelar información sobre histopatología, adenomas y sus tamaños, grado de displasia, etc., se puede realizar la búsqueda a partir de dichos términos *"histopathology","adenoma","size", "dysplasia"*, por separado o de forma combinada. En el momento de la búsqueda, la base de datos de ArchMS contiene arquetipos openEHR procedentes del

Tabla 10.3: Criterios de clasificación según guías clínicas europea y americana

Nivel de riesgo	Guía clínica europea	Guía clínica americana
Alto	El informe histopatológico cumple uno de los siguientes requisitos: (a) Al menos uno de los hallazgos encontrados es un adenoma avanzado, con un tamaño mayor o igual de 20 mm (b) Tiene 5 o más hallazgos	El informe histopatológico cumple uno de los siguientes requisitos: (a) Al menos uno de los hallazgos encontrados es un adenoma avanzado o existen más de tres hallazgos (b) Al menos uno de los adenomas tiene un tamaño igual o mayor a 20 mm
Intermedio	El informe histopatológico cumple uno de los siguientes criterios: (a) Contiene menos de 5 adenomas, de los cuales, al menos 1 es un adenoma avanzado, y el tamaño del más grande es menor que 20 mm (b) Contiene 3 o 4 adenomas y todos son normales	
Bajo	El informe histopatológico contiene como mucho 2 adenomas normales y ningún adenoma avanzado	El informe histopatólogico contiene como mucho 2 adenomas normales y ningún adenoma avanzado

gestor CKM relacionados con carcinomas. Una búsqueda textual según el término *"histopathology"* devuelve los resultados mostrados en la figura 10.1.

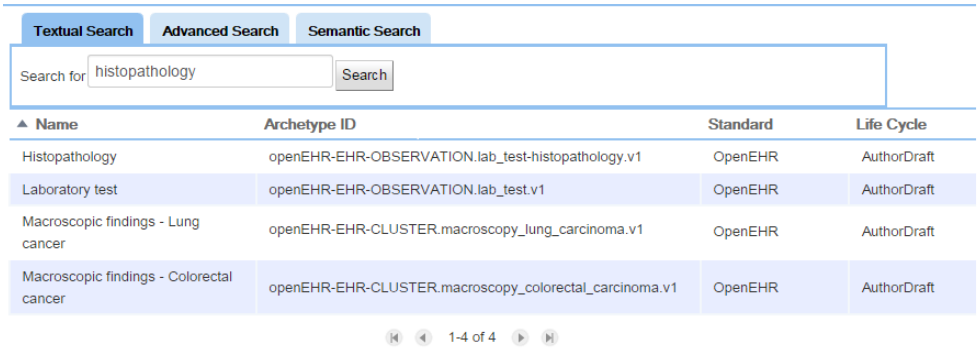

Figura 10.1: Interfaz de búsqueda textual y resultados a partir del término "histopathology"

Un arquetipo adecuado para la representación de datos sobre un informe histopatológico es *"Histopathology" (id: openEHR-EHR-OBSERVATION.lab_test-histopathology.v1)*. Sin embargo, búsquedas por términos como *"colorectal"* o *"screening"* no devuelven resultados apropiados para el registro de información sobre cribado de cáncer de colon y recto.

La información proporcionada por el arquetipo seleccionado *"Histopathology - Specialization: colorectal_screening"* es insuficiente para registrar información específica sobre hallazgos de adenomas (tipo, tamaño máximo,

grado de displasia, etc.) que es importante para el estudio. Por lo tanto este arquetipo fue especializado para crear uno nuevo que registrase toda la información necesaria. Al no haber un arquetipo apropiado para registrar información sobre el cribado de cáncer de colon y recto, otro nuevo arquetipo fue creado. La estructura de los dos arquetipos utilizados en este estudio se muestra en la figura 10.2. A la izquierda, el arquetipo especialización que registra información de una histopatología (*"Histopathology - Specialization: colorectal_screening"*) y a la derecha, el nuevo arquetipo de cribado (*"colorectal_screening"*).

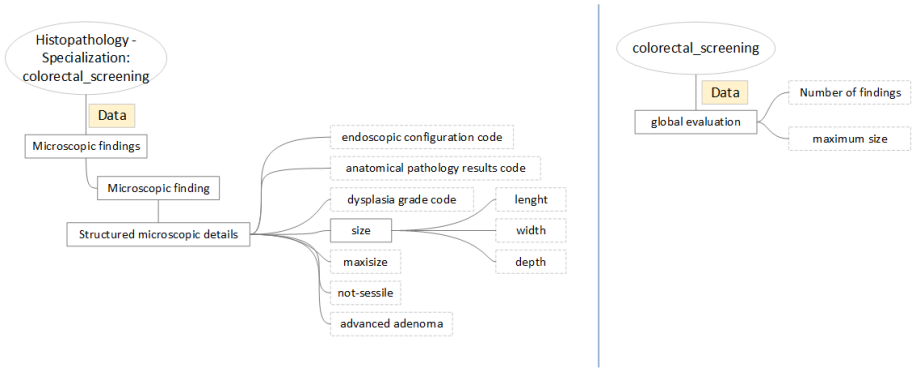

Figura 10.2: (Izquierda) Sección del diagrama de "Histopathology - Specialization: colorectal_screening"(id: openEHR-EHR-OBSERVATION.lab_test-histopathology-colorectal_screening.v1); (Derecha) Sección del diagrama de "colorectal_screening" (id: openEHR-EHR-EVALUATION.colorectal_screening.v1)

10.1.2 Importación y anotación de arquetipos

Los dos arquetipos seleccionados para modelar los datos del estudio son importados en ArchMS. El primer paso en el proceso de importación es la validación de los arquetipos con respecto a sus arquetipos padre, en caso de que tengan. En este caso, el arquetipo *"Histopathology - Specialization: colorectal_screening"* especializa al arquetipo *"Histopathology"*, por lo que se comprueba su corrección con respecto a este último. Una vez que los arquetipos han sido importados con éxito, la herramienta busca automáticamente arquetipos similares a ellos con el propósito de obtener recomendaciones de anotaciones.

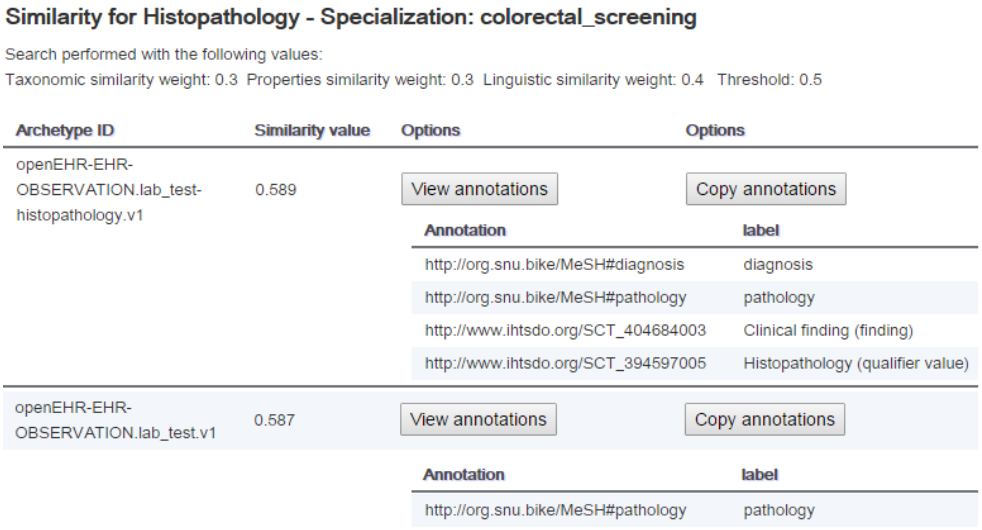

Figura 10.3: Arquetipos similares a openEHR-EHR-OBSERVATION.lab_test-histopathology-colorectal_screening.v1

En la figura 10.3 se muestra el resultado de esta acción para el arquetipo *"Histopathology - Specialization: colorectal_ screening"*. Ha encontrado dos arquetipos similares, con los cuales además tiene relación a través de especialización. La búsqueda por similitud hace uso del perfil semántica de los arquetipos (conjunto de *term-bindings* y anotaciones), sin embargo, al hacerlo inmediatamente después de la importación, el arquetipo aun no ha sido anotado, y además, en este caso específico, carece de *term-bindings*, por lo que su perfil semántico está vacío y, por lo tanto, la similitud se hace teniendo en cuenta las ontologías del modelo de arquetipos e información obtenida a partir de la transformación semántica del arquetipo ADL, es decir, la estructura del arquetipo, por ello es lógico que los arquetipos obtenidos sean aquellos a los que especializa y con los que comparte estructura de modelado.

Además de las anotaciones recuperadas de arquetipos similares, se pueden añadir anotaciones adicionales para enriquecer el perfil semántico del arquetipo. La herramienta proporciona un menú de anotación en el que proporciona acceso a terminologías de anotación y sugiere posibles términos para anotar relacionados con el arquetipo. Estas sugerencias se basan en el contenido textual de las secciones *keywords, purpose* y *ontology* del mismo. La figura 10.4 muestra la interfaz de anotación para el arquetipo *"Histopathology - Specialization colorectal_ screening"*. El arquetipo contiene en sus secciones *keywords, purpose* y *ontology* términos como *accession*, *adenoma*, *colorectal*, etc. La herramienta de anotación busca coincidencias de dichos términos en

una terminología seleccionada (en este caso MeSH), y muestra un panel con las sugerencias seleccionadas. En la figura 10.4, algunos de los términos procedentes de MeSH sugeridos a partir del término *adenoma* contenido en el arquetipo son *adenoma*, *adenomatoid_tumour* o *adenomatous_polyp*.

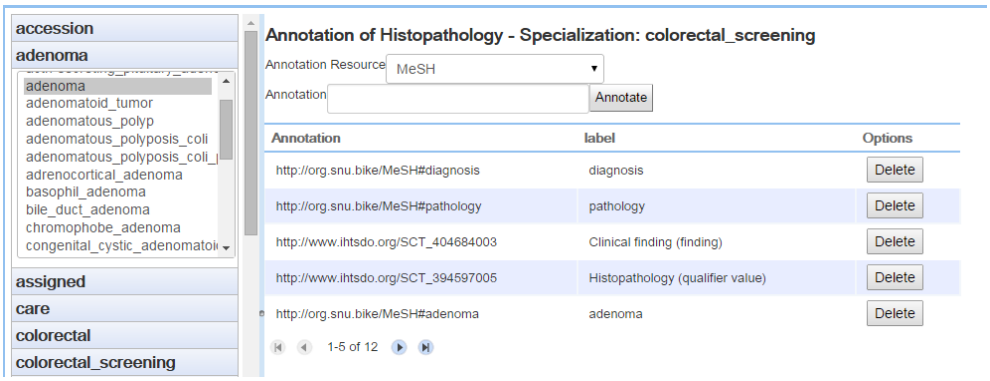

Figura 10.4: Anotación del arquetipo "Histopathology - Specialization: colorectal_screening"

Las tablas 10.4 y 10.5 muestran el conjunto final de anotaciones añadidas a los arquetipos *"Histopathology - Specialization colorectal screening"* y *"colorectal_screening"*, las cuales constituyen el perfil semántico de cada arquetipo.

Tabla 10.4: Anotaciones SNOMED-CT y MESH para el arquetipo "Histopathology - Specialization colorectal screening"

Código SNOMED-C	**Etiqueta**	**Código MeSH**	**Etiqueta**
25723000	Dysplasia	D000236	Adenoma
394597005	Histopathology	Q000175	Diagnosis
264267007	Colorectal	D003106	Colon
148322003	Screening	D010336	Pathology
404684003	Clinical Finding	D012007	Rectum
32048006	Adenoma	D008403	Mass Screening

10.1.3 Representación OWL de los extractos clínicos

Una vez seleccionados los arquetipos, los datos clínicos almacenados en una base de datos relacional fueron transformados a extractos clínicos EHR con-

Tabla 10.5: Anotaciones SNOMED-CT y MESH para el arquetipo "colorectal_screening"

Código SNOMED-CT	**Etiqueta**	**Código MeSH**	**Etiqueta**
264267007	Colorectal	Q000175	Diagnosis
148322003	Screening	D008403	Mass Screening

forme a dichos arquetipos. La ejecución de la clasificación de pacientes según las guías clínicas de cribado de cáncer de colon y recto requiere la representación de los datos clínicos según una ontología en dicho dominio. La figura 10.5 muestra un extracto del diagrama correspondientes a dicha ontología que contiene las entidades, propiedades y relaciones necesarias para modelar los datos sobre el informe histopatológico y de cribado de los pacientes clínicos. Un informe histopatológico (*HistopathologyReport*) está compuesto por una serie de hallazgos (*Findings*), que si contienen ciertas propiedades y relaciones pueden ser adenomas (clase equivalente *Adenoma*). Para un adenoma, se tiene información de su tamaño (propiedad *size*), grado de displasia (*hasDysplasiaType DysplasiaType*), patrón de crecimiento (*hasPatologyAnatomyResults PatologyAnatomyResults*) y si es sésil o no (*hasConfigurationEndoscopy ConfigurationEndoscopy*). El modelado de las subclases de *DysplasiaType, PatologyAnatomyResults* y *ConfigurationEndoscopy* se ha hecho haciendo uso de la terminología local utilizada en la recogida de los datos de los pacientes del programa de cribado de la Región de Murcia. De esta manera, cada una de las subclases es una clase equivalente, definida por un código proveniente de la terminología local, por ejemplo, la subclase de *DysplasiaType*, *HighDegree* se define como *DysplasiaType and (code value 284)*. Siendo 284 en los datos fuentes el código que identifica al grado de displasia alto.

Para representar los datos según la ontología del dominio se aplica el modelo de transformación a través de su implementación en la herramienta SWIT. En este caso de uso, los modelos de entrada son los dos arquetipos utilizados en la recogida de los datos clínicos, *"Histopathology - Specialization colorectal_screening"* y *"colorectal_screening"*, el modelo de salida es la ontología del dominio y los datos clínicos de cada paciente se encuentran almacenados como extractos de HCE en formato XML. El proceso de transformación requiere la definición de reglas de correspondencia entre los arquetipos y la ontología del dominio y de reglas de identidad sobre la ontología. Para agilizar la tarea de definición de reglas de correspondencia, se pueden

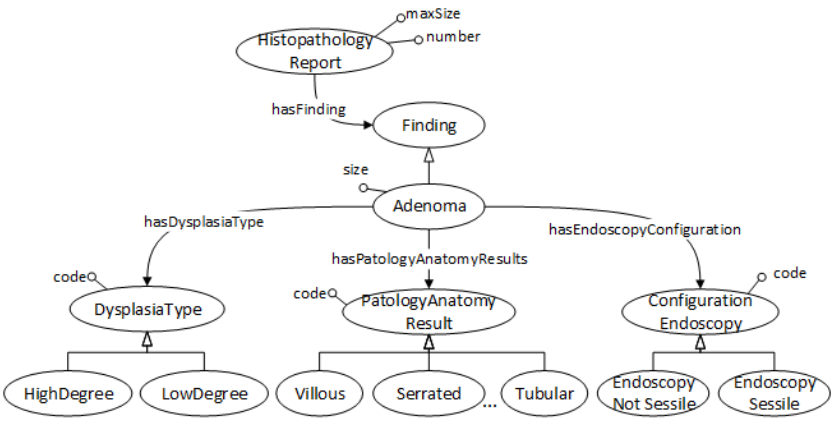

Figura 10.5: Ontología en el domino de informes histopatológicos

definir patrones que encapsulen la definición semántica de las entidades en la ontología del dominio. Por ejemplo, la figura 10.6 muestra el patrón que define una instancia de informe histopatológico (*HistopathologyReport*) en la ontología. Esta instancia registra un conjunto de hallazgos (*hasFinding ?finding*), el número total de estos (*number ?number*) y el tamaño del hallazgo más grande (propiedad *maxsize ?size*). El patrón permite que las instancias de *HistopathologyReport* se definan correctamente sin necesidad de conocer a fondo la ontología del dominio, pues especifica todas las propiedades y relaciones de dicha entidad.

```
?histopathologyReport:INVIDUAL,
?finding:INDIVIDUAL,
?size:CONSTANT,
?number:CONSTANT;
BEGIN
ADD ?histopathologyReport instanceOf HistopathologyReport,
ADD ?finding instanceOf Finding,
ADD ?histopathologyReport hasFinding ?finding,
ADD ?histopathologyReport number ?number,
ADD ?histopathologyReport maxsize ?size
END;
```

Figura 10.6: Patrón para la definición de instancias de "HistopathologyReport"

La figura 10.7 muestra la definición de las reglas de correspondencia con las variables del patrón. Cada extracto capturado utilizando el arquetipo

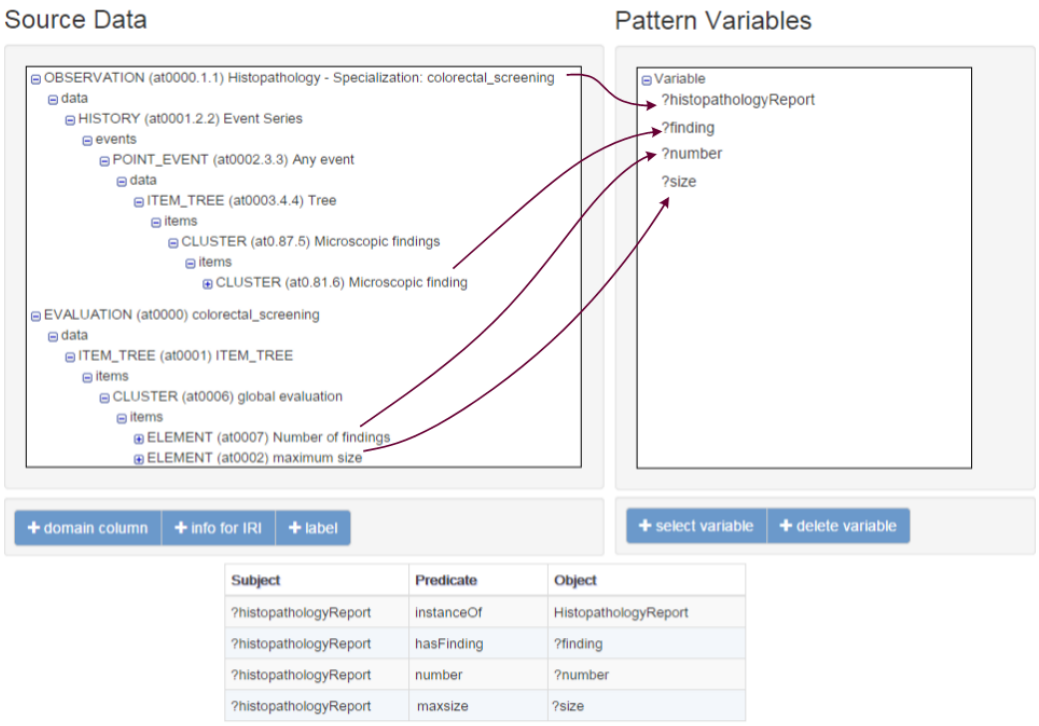

Figura 10.7: Interfaz de SWIT para el mapeo entre los dos arquetipos (izquierda) y las variables del patrón que define un informe histopatológico (derecha)

"Histopathology - Specialization colorectal_screening" corresponde con un informe histopatológico, así que se define una correspondencia entre la raíz del arquetipo y la variable *?histopathologyReport*. El elemento *"Microscopic finding"* representa el registro de un hallazgo, por lo que se define una correspondencia entre ese elemento y la variable *?finding*. Los valores para las variables *?size* y *number* se obtienen a partir de la información recogida en el extracto con el arquetipo *"colorectal_screening"*, por lo que se define una correspondencia entre el elemento *"maximun size"* y la variable *?size* y el elemento *"Number of findings"* y la variable *?number*.

La definición de una correspondencia para cada variable del patrón en SWIT genera las reglas de clase necesarias. La figura 10.8 muestra la regla de relación automáticamente generada a partir de la definición en SWIT de la correspondencia para *?histopathologyReport*.

Una instancia de *HistopathologyReport* se identifica por el paciente al que pertenece, mientras que un *Finding* se identifica a partir de la instancia *HistopathologyReport* a la que pertenece. Las clases *DysplasiaType, Patolog-*

```
<map>
<type>2Class</type>
    <class>
        <id>
          http://www.coode.org/oppl/variablemansyntax#?histopathologyReport
        </id>
    </class>
    <entity>
        <nodes>
            <node id="1">
                <id>OBSERVATION[@archetype_node_id="at0000.1.1"]</id>
            </node>
        </nodes>
    </entity>
</map>
```

Figura 10.8: Regla resultado de definir la correspondencia de la variables ?histopathologyReport en SWIT

yAnatomyResult y *ConfigurationEndoscopy* se definen a través de su campo *code*. La regla de identidad para una instancia de tipo *DysplasiaType* se muestra en la figura 10.9.

```
<condition>
    <class><id>http://miuras.inf.um.es/ontologies/precol.owl#DysplasiaType</id></class>
    <requirement>
        <and>
            <requirement>
                <scope>ALL</scope>
                <dataproperty>
                    http://miuras.inf.um.es/ontologies/precol.owl#code
                </dataproperty>
                <value>EQUALS</value>
            <requirement>
        </and>
    </requierement>
</condition>
```

Figura 10.9: Regla de identidad para “DysplasiaType”

Si los extractos clínicos de un paciente X tienen los datos mostrados en la tabla 10.6, el resultado de aplicar las reglas de transformación a este paciente creará las instancias en la ontología que se muestran en la figura 10.10.

Los hallazgos número 1 y 2 no se clasifican como adenomas debido a sus características, por lo que en total, el informe histopatológico del paciente sólo contiene un adenoma, clasificado como un adenoma normal (*NormalAdenoma*). Esta clasificación parcial puede ser añadida como anotaciones al

Tabla 10.6: Datos clínicos del paciente X

# hallazgo	Configuración endoscópica	Tipo de displasia	Patología anatómica	Tamaño máximo
1	11 (sésil)	275 (desconocida)	185 (hiperplasia)	2
2	11 (sésil)	275 (desconocida)	185 (hiperplasia)	2
3	13 (no sésil)	283 (baja)	181 (tubular)	5

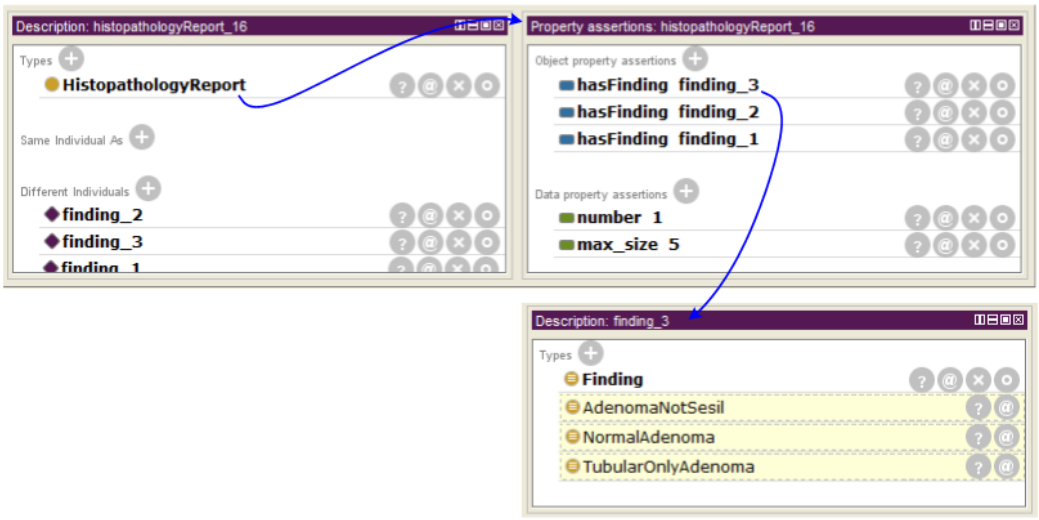

Figura 10.10: Representación de datos de hallazgos para el paciente X en la ontología del dominio

extracto clínico del paciente, así, además de las anotaciones obtenidas de sus arquetipos, se añaden anotaciones como *"NormalAdenoma"*, *"TubularOnlyAdenoma"* a partir de la clasificación de su adenoma 3 y 4 o *Hyperplasia* a partir de los valores de sus hallazgos número 1 y 2.

10.1.4 Clasificación de pacientes

Clasificar los pacientes según su riesgo de desarrollar cáncer de colon de acuerdo a las guías clínicas europea y americana requiere crear una ontología de clasificación con tantas reglas como grupos de clasificación existan. La tabla 10.7 muestra cada una de las reglas definidas en la ontología de clasificación para los grupos definidos en la tabla 10.3.

Si se aplican las reglas de la tabla 10.7 al paciente X cuya instancia de

Tabla 10.7: Reglas equivalentTo de clasificación según guías clínicas europea y americana

Nivel de riesgo	Guía clínica europea	Guía clínica americana
Alto	HighRiskEuropeanProtocol EquivalentTo (HistopathologyReport and ((max_size some integer[≥20)]) or (number some integer [≥5)])))	HighRiskAmericanProtocol EquivalentTo (HistopathologyReport and ((hasAdenoma some AdvancedAdenoma) or (number some integer[≥3)]))) or (HistopathologyReport and (max_size some integer [≥20)]))
Intermedio	IntermediateRiskEuropeanProtocol EquivalentTo HistopathologyReport and (((hasAdenoma some AdvancedAdenoma) and (max_size some integer[<20)]) and (number some integer [<5)])) or ((hasAdenoma only NormalAdenoma) and (number some integer [>2)]) and (number some integer [<5)])))	
Bajo	LowRiskEuropeanProtocol EquivalentTo HistopathologyReport and (hasAdenoma only NormalAdenoma) and (number some integer [<3)])	LowRiskAmericanProtocol EquivalentTo HistopathologyReport and (hasAdenoma only NormalAdenoma) and (number some integer [<3)])

HistopathologyReport aparece en la figura 10.10, el paciente será clasificado como de bajo riesgo según las guías europeas y americanas y las anotaciones *LowRiskEuropeanProtocol* y *LowRiskAmericanProtocol* pasarán a formar parte de su perfil.

10.1.5 Gestión de datos clínicos

ArchMS ofrece una interfaz de gestión tanto para los pacientes como para los médicos. Un médico puede ver todos los historiales clínicos de sus pacientes, mientras que un paciente puede visualizar su propio historial. La figura 10.11 muestra la vista de un paciente, dónde puede visualizar su extracto clínico sobre su informe histopatológico y los arquetipos utilizados para la recogida de datos.

Los pacientes pueden obtener recomendaciones de recursos de aprendizaje a partir de un extracto clínico o a partir de los arquetipos implicados en su historial clínico. En el caso del paciente X visto en las secciones anteriores, la información utilizada para obtener documentos de aprendizaje será:

- Si se obtienen a partir un arquetipo, ArchMS utiliza el perfil semántico del arquetipo, es decir, el conjunto de anotaciones y enlaces terminológicos.

- Si se obtienen a partir de un extracto, ArchMS utiliza:

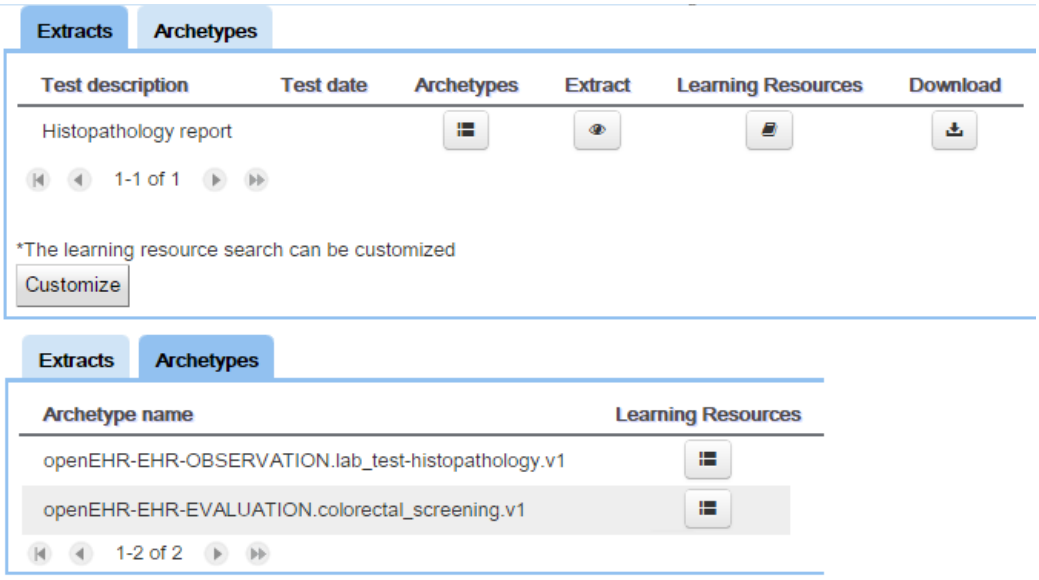

Figura 10.11: Interfaz ArchMS para el paciente X

- El perfil semántico de cada uno de los arquetipos utilizados para recopilar el extracto clínico.
- Si el especialista ha realizado operaciones de clasificación sobre el paciente, ArchMS también utiliza las nuevas anotaciones resultado de dicha clasificación.

Supongamos que en la plataforma hay disponibles dos recursos de aprendizaje, *"What you need to know about cancer of Colon and Rectum"* y *"Chemotherapy and You: Support for People With Cancer"*, cuyas anotaciones se muestran en las tablas 10.8 y 10.9.

Mediante la exploración de las anotaciones dadas a cada uno de los recursos, se deduce que el primero de ellos (*"What you need to know about cancer of Colon and Rectum"*) está dirigido a todos los pacientes con cáncer de colon y recto, incluyendo a todos los pacientes del programa de cribado, mientras que el segundo (*"Chemotherapy and You: Support for People With Cancer"*) está dirigido a pacientes a los que se les está aplicando quimioterapia (sin especificar tipo de cáncer) y a los pacientes del programa de cribado clasificados como alto riesgo. El paciente X, por su clasificación debería obtener la recomendación del primero de ellos, pero no del segundo, sin embargo esto depende de los parámetros dados al métodos de recomendación. El método de recomendación aplica el método de similitud, que depende de cuatro paráme-

Tabla 10.8: Anotaciones para "What you need to know about cancer of Colon and Rectum"

Código MeSH	**Etiqueta**	**Anotación de nivel de riesgo**
D003110	Colonic Neoplasms	LowRiskEuropeanProtocol
D009369	Neoplasm	LowRiskAmericanProtocol
D000236	Adenoma	HighRiskAmericanProtocol
D003106	Colon	HighRiskEuropeanProtocol
D012007	Rectum	IntermediateRiskEuropeanProtocol
D008403	Mass Screening	
D015179	Colorectal Neoplasm	
D011127	Polyps	

Tabla 10.9: Anotaciones para "Chemotherapy and You: Support for People With Cancer"

Código MeSH	**Etiqueta**	**Anotación de nivel de riesgo**
D003110	Drug Therapy	HighRiskAmericanProtocol
D009369	Neoplasm	HighRiskEuropeanProtocol
		AdvancedAdenoma

tros, los pesos dados a la distancia taxonómica, a la similitud de propiedades y a la similitud lingüística y el umbral de similitud. Para elegir los pesos y el umbral más adecuado nos fijamos en el origen de las anotaciones. Por un lado, los recursos formativos tienen anotaciones provenientes de MeSH, una terminología que se presenta como una taxonomía, por lo que la distancia taxonómica es el valor más importante y que carece de propiedades. Por otro lado, tienen anotaciones que corresponden con las categorías de clasificación. Estas categorías tienen la misma distancia taxonómica, y sus diferencias importantes están en las propiedades que contienen. La similitud lingüística se puede tener en cuenta pero su relevancia es mínima, por ejemplo HighRiskAmericanProtocol y LowRiskAmericanProtocol son lingüísticamente similares pero semánticamente muy distintos.

Para tener un equilibrio entre las anotaciones provenientes de MeSH y las anotaciones provenientes de la ontología de clasificación, se obtienen recomendaciones para el paciente X con un valor 0,6 para el peso de la distancia taxonómica, de 0,3 para la similitud de propiedades y de 0,1 para la similitud

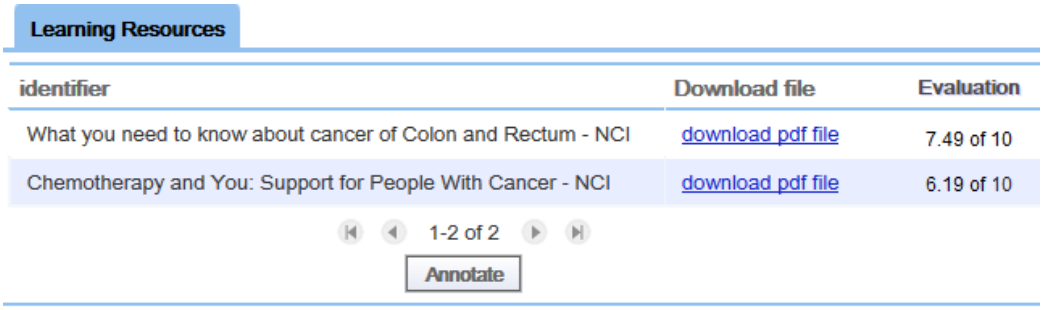

Figura 10.12: Resultados recomendación de recursos de aprendizaje para el paciente X

lingüística. En una primera ejecución se escoge un umbral bajo de 0,5 para ver los valores de similitud asignados a cada uno de los recursos formativos.

Con estos parámetros, los resultados para el paciente X aparecen en la figura 10.12. Mientras que el primer recurso obtiene una puntuación de similitud de 0,749 (7,49 sobre 10), el segundo obtiene una puntuación de 0,619 (6,19 sobre 10). La diferencia entre ambos no es muy significativa, esto se debe a los valores que tienen los pesos. La similitud con más peso es la distancia taxonómica, esto funciona bien para las anotaciones provenientes de MesH pero no para las que provienen de la ontología de clasificación.

La tabla 10.10 muestra los valores de similitud para cada uno de los tipos de anotación por separado. Teniendo en cuenta las anotaciones MeSH, la diferencia entre los dos recursos es muy significativa, mientras que teniendo en cuenta las anotaciones de la ontología de clasificación, la similitud es prácticamente la misma. Esto se debe a que las clases que definen las categorías de clasificación tienen la misma distancia taxonómica en la ontología.

Tabla 10.10: Valores de similitud para las anotaciones MeSH y para las anotaciones provenientes de la ontología de clasificación

	MeSH	**Ontología de clasificación**
What you need to know about cancer of Colon and Rectum	0,846	0,652
Chemotherapy and You: Support for People With Cancer	0,591	0,647

Este ejemplo muestra que sería interesante mejorar el método de reco-

mendación permitiendo definir distintos parámetros de peso para cada tipo de anotación. Es decir, dar más importante a la distancia taxonómica en las anotaciones MeSH y dar más importancia a la similitud de propiedades en las anotaciones de la ontología de clasificación.

10.2 Transformación de modelos clínicos: CEM a arquetipos openEHR

El objetivo principal de este escenario de aplicación es la reutilización de los modelos clínicos disponibles para los sistemas de Historia Clínica Electrónica (HCE) basados en CEM para su uso en sistemas basados en el estándar openEHR. Para ello se utilizan los métodos de transformación de modelos explicados en esta tesis.

La transformación entre modelos utiliza la representación OWL disponible para cada uno de los modelos de información y clínicos de CEM y openEHR. Para este caso de uso se utiliza la representación OWL de CEM creada dentro del proyecto "Strategic Heatlh IT Advanced Research Project, secondary use of EHR" (SHARP) que se presenta en la sección 3.5.2 y la representación OWL de openEHR desarrollada en el contexto del proyecto Archeck, presentada en la sección 3.5.1.2. El uso de este formalismo común en la transformación de ambos modelos, el cual soporta actividades como razonamiento automático, permite asegurar que solo contenido lógicamente consistente es transformado. En ambas aproximaciones, la construcción de la representación OWL sigue una metodología similar, los conceptos principales de los modelos de información se representan como clases OWL y los modelos clínicos como subclases de dichas clases, mientras que cada entidad restringida se define como una clase OWL con restricciones definidas.

En este escenario se diseña la transformación unidireccional de modelos clínicos basados en CEM a arquetipos openEHR. Cada modelo CEM se clasifica en una categoría estructural básica que captura los atributos comunes de un modelo perteneciente a una clase específica. Por lo tanto, los CEM de la misma categoría estructural siguen una estructura similar y se les puede aplicar la misma metodología de transformación. Por ejemplo, cada modelo CEM que sea de tipo *Panel* sigue el mismo conjunto de transformaciones para obtener su representación openEHR. Debido a esto, se propuso la creación de plantillas OWL que definen la representación tipo de una categoría CEM en openEHR. La figura 10.13 muestra parte de la plantilla openEHR definida para representar modelos CEM de tipo *Panel*. A la izquierda aparece el *Panel* con una propiedad *item*. A la derecha, las entidades en gris representan las

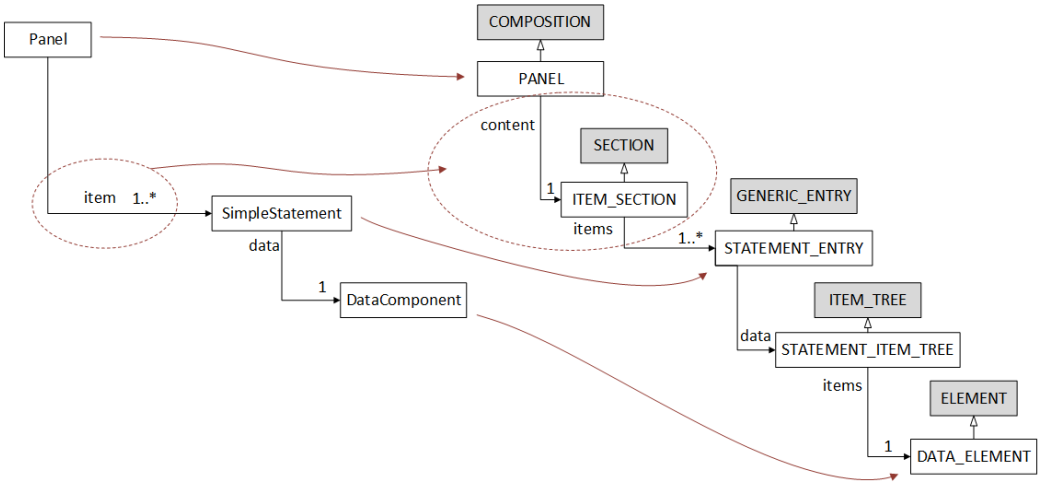

Figura 10.13: Transformación de un Panel CEM a un arquetipo openEHR

clases del modelo de información de openEHR, mientras que las entidades en blanco representan las entidades equivalentes al Panel CEM que forman la plantilla definida.

La figura 10.13 también muestra las correspondencias entre el modelo CEM y la plantilla equivalente. El estudio de las representaciones OWL de ambas especificaciones dio lugar a la identificación de dos tipos de correspondencia:

- Correspondencia clase a clase: Realiza la correspondencia entre dos clases. En este caso se aplica la regla de transformación para crear una clase OWL a partir de otra clase OWL existente presentada en la sección 7.2.3. La correspondencia entre una clase A de CEM y una clase B de openEHR crea una nueva clase C congruente a A y subclase de B. En la figura 10.13 la clase *openEHR:PANEL* de la plantilla openEHR es congruente a la clase *cem:Panel* de CEM. Por lo tanto:

$$Regla\_clase\_clase(cem : Panel, openEHR : COMPOSITION) : \exists openEHR : PANEL \sqsubseteq openEHR : COMPOSITION \mid openEHR : PANEL \cong cem : Panel \quad (10.1)$$

  Es decir, todo modelo CEM definido como subclase de *cem:Panel* será definido en openEHR como subclase de *openEHR:Panel*.

- Correspondencia propiedad a estructura: Realiza la correspondencia de un axioma de propiedad de un modelo CEM a una estructura ontológi-

ca en openEHR. Por ejemplo, en el ejemplo de la figura 10.13 el axioma de *cem:Panel* definido como *cem:Panel subClassOf cem:item min 1 cem:SimpleStatement* se corresponde con la estructura en el arquetipo openEHR *openEHR:PANEL subClassOf openEHR:content exactly 1 (openEHR:ITEM_SECTION and openEHR:items min 1 openEHR-:STATEMENT_ENTRY)*, por lo que su correspondencia es con una estructura de clases y propiedades.

La correspondencia propiedad a estructura no puede realizarse con una regla simple, por lo tanto, el uso de patrones para este caso no sólo es recomendable siguiendo las buenas prácticas de reutilización en construcción de ontologías, sino que se hace necesario para definir reglas de correspondencia más complejas. La existencia de plantillas para cada una de las categorías estructurales de CEM permite la creación de patrones de transformación para cada una ellas, los cuales crean una vista sobre cada una de las plantillas.

La figura 10.14 muestra los patrones que permiten transformar un *Panel* CEM en un arquetipo openEHR de tipo *COMPOSITION*. En los patrones, las variables *?panel* y *?statement* se hacen corresponder con entidades de un modelo CEM de entrada, mientras que el resto de variables, *?panelOPENEHR, ?itemSection, ?statementEntry, ?statementItemTree, ?statementDataElement* parametrizan las entidades del arquetipo openEHR de salida y se definen en el patrón a partir de las variables *?panel* y *?statement* de entrada.

Los dos primeros patrones de la figura 10.14 crean la estructura principal de un *Panel* compuesto por un *SimpleStatement* en su relación *item* en openEHR. El tercer patrón crea el resto de la estructura de un *SimpleStatement* con un *DataComponent*.

Debido a la existencia de plantillas para cada una de las categorías estructurales de CEM, las correspondencias entre un modelo CEM y su equivalente openEHR están predefinidas para cualquier modelo CEM de entrada. Los patrones de la figura 10.14 se utilizan sobre una plantilla openEHR que define el arquetipo equivalente a un *Panel* CEM, por lo que dado un modelo CEM de entrada de tipo *Panel*, las reglas de correspondencia predefinidas enlazan la variable *?panel* con la entidad subclase de *Panel* en el CEM y la variable *?statement* con la entidad subclase de *SimpleStatmenet* en el CEM. De esta manera, el proceso de transformación se puede automatizar en los siguiente pasos:

1. Identificación de plantillas: Dado un modelo CEM se identifican las plantillas openEHR que le corresponden dependiendo de su categoría estructural.

```
?panel:CLASS,
?panelOPENEHR:CLASS = create(?panel.RENDERING+"_PANEL" ),
?itemSection:CLASS = create(?panel.RENDERING+"_ITEM_SECTION")
BEGIN
   ADD ?panelOPENEHR SubClassOf PANEL,
   ADD ?itemSection SubClassOf ITEM_SECTION,
   ADD ?panelOPENEHR SubClassOf (COMPOSITION_content exactly 1 ?itemSection)
END;
```

```
?panel:CLASS,
?statement:CLASS,
?itemSection:CLASS = create(?panel.RENDERING+"_ITEM_SECTION"),
?statementEntry:CLASS = create(?statement.RENDERING+"_STATEMENT_ENTRY")
BEGIN
   ADD ?statementEntry SubClassOf STATEMENT_ENTRY,
   ADD ?itemSection SubClassOf (SECTION_items max 1 ?statementEntry)
END;
```

```
?statement:CLASS,
?statementItemTree:CLASS = create(?statement.RENDERING+"_STATEMENT_ITEM_TREE"),
?statementDataElement:CLASS = create(?statement.RENDERING+"_DATA_ELEMENT")
BEGIN
    ADD ?statementDataElement SubClassOf DATA_ELEMENT,
    ADD ?statementItemTree SubClassOf (ITEM_TREE_items max 1 ?statementDataElement)

END;
```

Figura 10.14: Patrones de transformación de un Panel CEM a un arquetipo openEHR

2. Instanciación de patrones: Los patrones asociados a las plantillas se instancian con las entidades del modelo CEM de entrada.

3. Ejecución de la transformación: Los patrones instanciados se ejecutan sobre la plantilla seleccionada y se obtiene el arquetipo openEHR equivalente.

Si se va a transformar el modelo CEM *BloodPressurePanel* que se define en sintaxis OWL Manchester de la siguiente forma:

```
Class: BloodPressurePanel
    SubClassOf:
        item max 1 DiastolicBloodPressure,
        item max 1 SystolicBloodPressure,
        Panel
...

Class: DiastolicBloodPressure
```

```
    SubClassOf:
        data max 1 PQ,
        SimpleStatement
...

Class: SystolicBloodPressure
    SubClassOf:
        data max 1 PQ,
        SimpleStatement
...
```

La aplicación del proceso de transformación al modelo *BloodPressurePanel* lo identifica como un *Panel* compuesto por *SimpleStatement* y selecciona la plantilla y los patrones de las figuras 10.13 y 10.14. Las reglas de correspondencia predefinidas enlazan *BloodPressurePanel*→*?panel* y *DiastolicBloodPressure*→*?statement*, *SystolicBloodPressure*→*?statement*. El resultado de ejecutar los patrones es el arquetipo openEHR OWL descrito a continuación:

```
Class: BloodPressurePanel_PANEL
    SubClassOf:
        content exactly 1 BloodPressurePanel_ITEM_SECTION,
        PANEL

Class: BloodPressurePanel_ITEM_SECTION
    SubClassOf:
        ITEM_SECTION

Class: BloodPressurePanel_ITEM_SECTION
    SubClassOf:
        ITEM_SECTION,
        items max 1 SystolicBloodPressure_STATEMENT_ENTRY,
        items max 1 DiastolicBloodPressure_STATEMENT_ENTRY

Class: DiastolicBloodPressure_STATEMENT_ENTRY
    SubClassOf:
        STATEMENT_ENTRY,
        data max 1 DiastolicBloodPressure_STATEMENT_ITEM_TREE

Class: SystolicBloodPressure_STATEMENT_ENTRY
    SubClassOf:
        STATEMENT_ENTRY,
        data max 1 SystolicBloodPressure_STATEMENT_ITEM_TREE
```

El resultado final de este trabajo es el conjunto de plantillas openEHR OWL para las categorías estructurales más comunes de CEM, *Panel* (compuesto por *SimpleStatement, CompoundStatement* u otros *Panel*), *SimpleStatement, CompoundStatement, Attribution, Modifier* y *Qualifier*, el conjunto de patrones OPPL2 que permiten realizar la transformación y una herramienta [246] que implementa los tres pasos (identificación de plantillas, instanciación de patrones y ejecución de la transformación) de la transformación CEM a openEHR.

Se realizó una evaluación técnica de las ejecución de la transformación para una serie de modelos CEM, verificando que los resultados obtenidos coincidían con la salida esperada según la ontología OWL de los arquetipos openEHR utilizada y las plantillas openEHR OWL diseñadas. El tiempo promedio para transformar un modelo CEM a un arquetipo openEHR en OWL es de 1,587 segundos en un servidor de 2.13 GHz, procesador de 8 núcleos, utilizando 6 GB de máquina virtual Java, razonador Hermit 1.3.5, OWLAPI 3.4.5 y OPPL2. Existe una correlación positiva entre el tiempo total y el número de *Components* que contiene un CEM (coeficiente de correlación de Pearson = 0,964). Un modelo de regresión lineal podría explicar la relación (ver figura 10.15) entre el número de *Components* en el modelo CEM y el tiempo (R = 0,996; P=0,0 para la constante y el número de componentes), que es un buen valor en términos de escalabilidad.

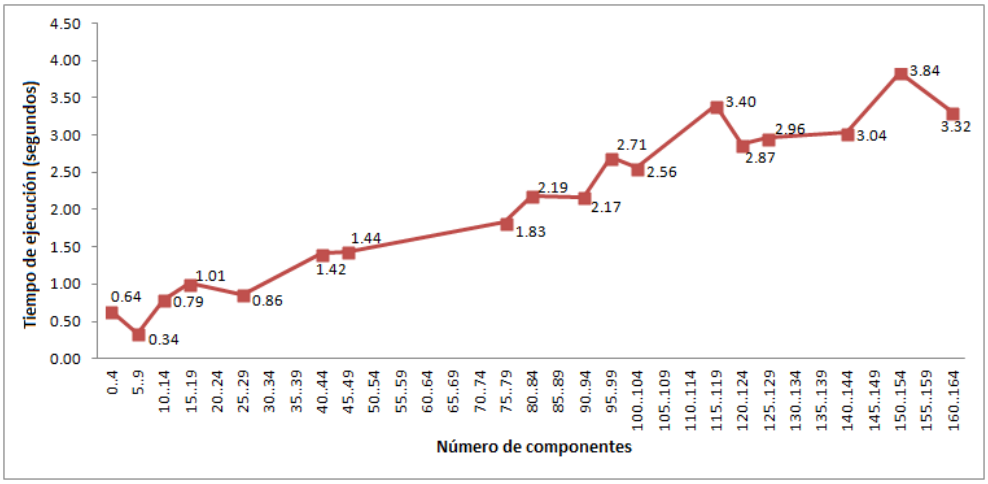

Figura 10.15: Tiempo de ejecución del proceso de transformación respecto al número de componentes de CEM

10.3 Datos ortólogos, enfermedades genéticas y anotación de secuencias genómicas

La herramienta SWIT ha sido utilizada para crear un repositorio integrado de recursos de información sobre genes ortólogos, enfermedades genéticas humanas e información derivada de procesos de anotación de genomas. Los genes ortólogos son copias diferentes de la misma secuencia genética, presente en especies diferentes y resultado de la divergencia evolutiva. Los repositorios de genes ortólogos proporcionan conjuntos de genes ortólogos, obtenidos utilizando distintos métodos y que proporcionan distintos detalles. Para la investigación científica es interesante el acceso a estos recursos de forma integrada, como se demuestra por iniciativas como Quest for Orthologs [247], que tiene como objetivo mejorar la estandarización de la información sobre ortólogos para el uso compartido de los conjuntos de datos. Entre sus propuestas se encuentran los formatos basados en XML para la representación de datos sobre ortólogos y secuencias de datos, OrthoXML y SeqXML. La unión de genes ortólogos, junto a enfermedades genéticas y la información derivada de los procesos de anotación de secuencias genómicas resulta de interés para estudiar las relaciones de ortología de los genes y su influencia en las enfermedades genéticas durante los procesos de anotación de genomas.

El repositorio creado en este proceso, llamado OGO [165], en su primera versión proporcionaba un recurso integrado de genes ortólogos y enfermedades relacionadas. Para ello integró recursos provenientes de las bases de datos KOG, Inparanoid, OrthoMCL, Homologene y OMIM. La integración se realizó utilizando como modelo de salida global una ontología, OGO, que proporciona una representación formal del dominio de los ortólogos y las enfermedades genéticas. Este repositorio contenía más de 50000 grupos de ortólogos, más de un millón de genes y proteínas y alrededor de 18000 enfermedades genéticas humanas.

Para generar el contenido de la última versión del repositorio, no sólo se utilizó la información de ortólogos contenida en esquemas relacionales, sino también los datos en ficheros XML siguiendo el formato OrthoXML disponible para OrthoMCL e Inparanoid. La figura 10.16 muestra un ejemplo de ficheros OrthoXML. En este formato se muestran primero los genes (`<genes>`) por especie (`<species>`) a la que pertenecen y aporta información del recurso del que se obtiene la información (`<database>`) y de la proteína a la que se traduce el gen (`<proteinId>`). La segunda parte del fichero muestra los grupos de genes (`<groups>`), que pueden ser ortólogos (`<orthologGroup>`) o parálogos (`<paralogGroup>`). El esquema OrthoXML permite que los grupos contengan genes y otros grupos anidados.

```
<?xml version="1.0" encoding="utf-8"?>
<orthoXML>
  <species name="Caenorhabditis elegans" NCBITaxId="6239">
    <database name="WormBase" version="Caenorhabditis-elegans_WormBase_WS199_protein-all.fa">
      <genes>
        <gene id="1" geneId="WBGene00000962" protId="CE23997" />
        <gene id="5" geneId="WBGene00006801" protId="CE43332" />
      </genes>
    </database>
  </species>
  <species name="Homo Sapiens" NCBITaxId="9606">
    <database name="Ensembl" version="Homo_sapiens.NCBI36.52.pep.all.fa">
      <genes>
        <gene id="2" geneId="ENSG00000197102" protId="ENSP00000348965" />
        <gene id="6" geneId="ENSG00000198626" protId="ENSP00000355533" />
      </genes>
    </database>
  </species>
  <scores>
    <scoreDef id="bit" desc="BLAST score in bits of seed orthologs" />
    <scoreDef id="inparalog" desc="Distance between edge seed ortholog" />
    <scoreDef id="bootstrap" desc="Reliability of seed orthologs" />
  </scores>
  <groups>
    <orthologGroup id="1">
      <score id="bit" value="5093" />
      <geneRef id="1">
        <score id="inparalog" value="1" />
        <score id="bootstrap" value="1.00" />
      </geneRef>
      <geneRef id="2">
        <score id="inparalog" value="1" />
        <score id="bootstrap" value="1.00" />
      </geneRef>
    </orthologGroup>
    <orthologGroup id="3">
      <score id="bit" value="3795" />
      <geneRef id="5">
        <score id="inparalog" value="1" />
        <score id="bootstrap" value="1.00" />
      </geneRef>
      <geneRef id="6">
        <score id="inparalog" value="1" />
        <score id="bootstrap" value="1.00" />
      </geneRef>
    </orthologGroup>
  </groups>
</orthoXML>
```

Figura 10.16: Ejemplo de información sobre genes ortólogos representada en formato OrthoXML

La figura 10.17 muestra un extracto de la ontología OGO utilizado para realizar la transformación de los contenidos sobre genes ortólogos en OrthoXML. Para ello hay que definir correspondencias entre el esquema OrthoXML y la ontología.

Una de las correspondencias está entre la entidad `species` de OrthoXML y la clase *organisms* de la ontología OGO. La ontología OGO contiene un individuo para cada una de las especies de la taxonomía del NCBI, por lo que la transformación no debería crear nuevos individuos de especies que ya existan en la ontología. En esta transformación, el interés de las especies está

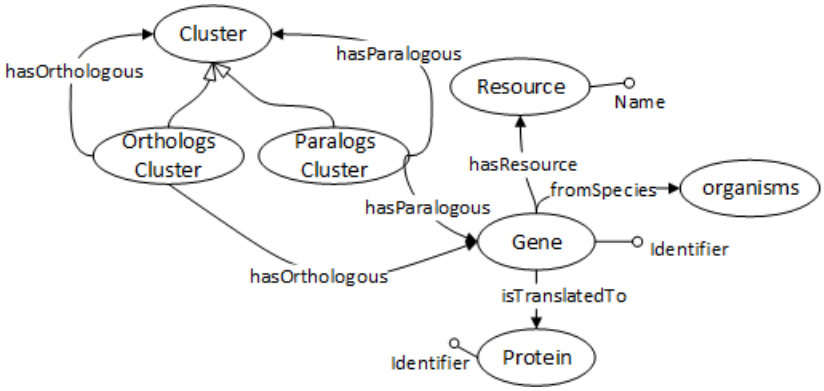

Figura 10.17: Diagrama de un extracto de la ontología OGO

en su relación con los genes, por lo que esta correspondencia se puede definir a través del siguiente patrón:

```
?gene : INDIVIDUAL,
?species : INDIVIDUAL[instanceOf organisms],
?speciesFinal : INDIVIDUAL=create("NCBI_"+?species.RENDERING),
?id : CONSTANT
BEGIN
    ADD ?gene fromSpecies ?speciesFinal,
    ADD ?gene Identifier ?id,
    ADD ?gene instanceOf Gene
END;
```

Los individuos de tipo *organisms* en la ontología OGO tienen URIs con el formato *"http://miuras.inf.um.es/ontologies/swit/ncbi.owl#NCBI_"+TaxonomyId*, donde *TaxonomyId* es el identificador de la especie en la taxonomía NCBI. En OrthoXML, una especie se identifica a través del elemento `species` y los atributos *@name* y *@NCBITaxId*. Para asociar instancias de `species` con individuos ya existentes en la ontología, la variable *?species* se define para crear un individuo temporal a partir de la información *@NCBITaxId* del elemento `species`. El valor que toma esta variable se utiliza para construir el individuo parametrizado por la variable *?speciesFinal*, que se construye siguiendo el formato de URI de las especies en la ontología OGO.

La asociación de las variables del patrón con entidades del recurso OrthoXML de entrada instancia las siguientes plantillas de reglas:

La plantilla 10.2 se utiliza para crear instancias de *Gene*. La instanciación de esta plantilla genera la regla de clase que se muestran en la figura 10.18.

$$Plantilla\_regla\_clase(C_1, ?gene)\{ \\ ?gene \in Gene \wedge \forall i_1 \in C_1, \exists i_2 \subseteq ?gene \mid i_2 \cong i_1\} \quad (10.2)$$

```
<map>
    <type>2Class</type>
    <class><id>http://www.coode.org/oppl/variablemansyntax#?gene</id></class>
    <entity>
        <nodes>
            <node id="1">
                <id>gene</id><base>/orthoXML/species/database/genes</base>
            </node>
        </nodes>
        <infos>
            <info><id>@geneId</id><base nodeRef="1"/></info>
            <info><id>@NCBITaxId</id>
                  <base>/orthoXML/species</base><base nodeRef="1"/>
            </info>
        </infos>
    </entity>
</map>
```

Figura 10.18: Regla de clase para crear instancias de *Gene*

La plantilla 10.3 se utiliza para asociar la propiedad *Identifier* a las instancias de *Gene*. La instanciación de esta plantilla genera la regla de propiedad que se muestran en la figura 10.19.

$$Plantilla\_regla\_propiedad((C_1, A_1), (?gene, ?id))\{ \\ ?gene \in Gene \wedge \forall i_1 \in C_1, \exists i_2 \subseteq ?gene \mid value(Gene, Identifier, i_2) \subseteq ?id \wedge \\ value(C_1, A_1, i_1) = value(Gene, Identifier, i_2)\} \quad (10.3)$$

```
<map>
    <type>2Prop</type>
    <domain>
        <class>
            <id>http://www.coode.org/oppl/variablemansyntax#?gene</id>
        </class>
        <entity>
            <nodes>
                <node id="1">
                    <id>gene</id><base>/orthoXML/species/database/genes</base>
                </node>
            </nodes>
            <infos>
                <info><id>@geneId</id><base nodeRef="1"/></info>
                <info><id>@NCBITaxId</id>
                    <base>/orthoXML/species</base><base nodeRef="1"/>
                </info>
            </infos>
        </entity>
    </domain>
    <predicate>
        <id>http://miuras.inf.um.es/ontologies/OGO.owl#Identifier</id>
    </predicate>
    <range>
        <class><id>http://www.coode.org/oppl/variablemansyntax#?id</id></class>
        <value>
            <nodes><node id="2"><id>@geneId</id><base nodeRef="1"/></node></nodes>
        </value>
    </range>
</map>
```

Figura 10.19: Regla de propiedad para asignar la propiedad *Identifier* a las instancias de *Gene*

La variable *?speciesFinal* se crea a partir de *?species* y depende de *?gene* a través de la relación *fromSpecies*. Por ello, se genera la plantilla 10.4, que relaciona las instancias de *Gene* con las instancias de *organisms* utilizando la variable *?species*, que resultará en que la variable *?speciesFinal* tome el valor adecuado. La instanciación de esta plantilla genera la regla de relación que se muestran en la figura 10.20.

$$\begin{aligned} Plantilla\_regla\_relacion((C_1, R_1, C_2), (?gene, ?species))\{ \\ ?gene \in Gene \wedge ?species \in organisms \wedge \\ Plantilla\_regla\_clase(C_1, ?gene) \wedge \\ Plantilla\_regla\_clase(C_2, ?species) \wedge \\ \{\forall i_1 \in C_1, i_2 \in C_2, \exists i_3 \subseteq ?gene, i_4 \subseteq ?species \mid \\ relacion(i_1, R_1, i_2) \wedge relacion(i_3, fromSpecies, i_4)\} \\ \Rightarrow R_1 \cong fromSpecies\} \end{aligned} \quad (10.4)$$

```
<map>
    <type>2Rel</type>
    <domain>
        <class>
            <id>http://www.coode.org/oppl/variablemansyntax#?gene</id>
        </class>
        <entity>
            <nodes>
                <node id="1">
                    <id>gene</id><base>/orthoXML/species/database/genes</base>
                </node>
            </nodes>
            <infos>
                <info><id>@geneId</id><base nodeRef="1"/></info>
                <info><id>@NCBITaxId</id>
                      <base>/orthoXML/species</base><base nodeRef="1"/>
                </info>
            </infos>
        </entity>
    </domain>
    <predicate>
        <id>http://miuras.inf.um.es/ontologies/OGO.owl#fromSpecies</id>
    </predicate>
    <range>
        <class>
            <id>http://www.coode.org/oppl/variablemansyntax#?species</id>
        </class>
        <entity>
            <nodes><node id="2"><id>species</id><base nodeRef="1"/></nodes>
            <infos>
                <info><id>@NCBITaxId</id><base nodeRef="2"/></info>
            </infos>
        </entity>
    </range>
</map>
```

Figura 10.20: Regla de relación para asociar las instancias de *Gene* las instancias de *organisms* a la que pertenecen

La variable *?speciesFinal* no genera plantillas de regla, pues se obtiene a partir de la variable *?species*. Mientras que la variable *?species* solo aparece en una plantilla de regla de relación, pues sólo aparece en el cuerpo del patrón en relación a *?gene*.

En el esquema OrthoXML, la referencia a los genes que forman los grupos de homólogos se hace a través del elemento `<geneRef>` y su atributo *@id* que equivale al atributo *@id* del elemento `<gene>`. La creación de las instancias de *Gene* se hacen según el elemento `<gene>` y su atributo *@geneId*. Por lo que a la hora de definir la relación *"OrthologsCluster hasOrthologous Gene"*, la correspondencia se debe hacer tanto con el elemento `<gene>` como con el elemento `<geneRef>`. La figura 10.21 muestra el resultado de esta regla de relación. En la sección `<range>` se define la correspondencia para las ins-

```
<map>
    <type>2Rel</type>
    <domain>
        <class>
            <id>http://miuras.inf.um.es/ontologies/OGO.owl#OrthologsCluster</id>
        </class>
        <entity>
            <nodes>
                <node id="1">
                    <id>orthologGroup</id><base>/orthoXML/groups</base>
                </node>
            </nodes>
        </entity>
    </domain>
    <predicate>
        <id>http://miuras.inf.um.es/ontologies/OGO.owl#hasOrthologous</id>
    </predicate>
    <range>
        <class><id>http://miuras.inf.um.es/ontologies/OGO.owl#Gene</id></class>
        <entity>
            <nodes>
                <node id="2"><id>geneRef</id><base nodeRef="1"/></node>
                <node id="3">
                    <id>gene</id><base>/orthoXML/species/database/genes</base>
                </node>
            </nodes>
            <option>
                <key nodeRef="2">@id</key><value nodeRef="3">@id</value>
            </option>
            <infos><info><id>@geneId</id><base nodeRef="3"/></info></infos>
        </entity>
    </range>
</map>
```

Figura 10.21: Regla de relación que asocia *OrthologsCluster* con *Gene*

tancias de *Gene*, dónde se corresponden tanto con el elemento `<gene>` como `<geneRef>` y en la sección `<option>` se igualan sus valores de *@id*.

La integración de varios recursos requiere la definición de reglas de identidad para detectar redundancias. La figura 10.22 muestra la regla de identidad para la clase *Gene*, que se define a partir de la especie a la que pertenece y su identificador. Es decir, dos instancias de *Gene* que pertenezcan a la misma especie y que tengan alguno de sus identificadores coincidente, serán consideradas una misma instancia.

En un versión posterior del repositorio OGO, la ontología global OGO se actualizó para incluir información derivada de la anotación de secuencias genómicas y se incluyó información sobre tres organismos de la familia *Mucoraceae*, de interés para la producción de biocombustibles. Este repositorio ha

```
<condition>
    <class><id>http://miuras.inf.um.es/ontologies/OGO.owl#Gene</id></class>
    <requirement>
        <and>
            <requirement>
                <scope>ALL</scope>
                <objectproperty>
                    http://miuras.inf.um.es/ontologies/OGO.owl#fromSpecies
                </objectproperty>
                <value>EQUALS</value>
                <class>
                    http://miuras.inf.um.es/ontologies/swit/ncbi.owl#organisms
                </class>
            <requirement>
            <requirement>
                <scope>SOME</scope>
                <dataproperty>
                    http://miuras.inf.um.es/ontologies/OGO.owl#Identifier
                </dataproperty>
                <value>EQUALS IGNORE CASE</value>
            </requirement>
        </and>
    </requierement>
</condition>
```

Figura 10.22: Regla de identidad para las instancias de *Gene*

sido publicado en la Web de Datos siguiendo los principios de Linked Data como el repositorio OGOLOD, que contiene 38035102 tripletas.

10.4 Componentes químicos

El cribado virtual es una técnica de cálculo utilizado en el descubrimiento de fármacos. Consiste en buscar en bibliotecas de pequeñas moléculas con el fin de identificar aquellas estructuras más prometedoras para enlazar con dianas terapéuticas. En este escenario se obtiene una representación semántica de una de estas librerías de moléculas, ZINC [248], con el objetivo de mejorar los métodos de selección de componentes. La librería ZINC puede ser descargada en formato XML. Para obtener su representación semántica se construyó una ontología OWL para este caso de uso y el esquema XSD correspondiente al XML descargado.

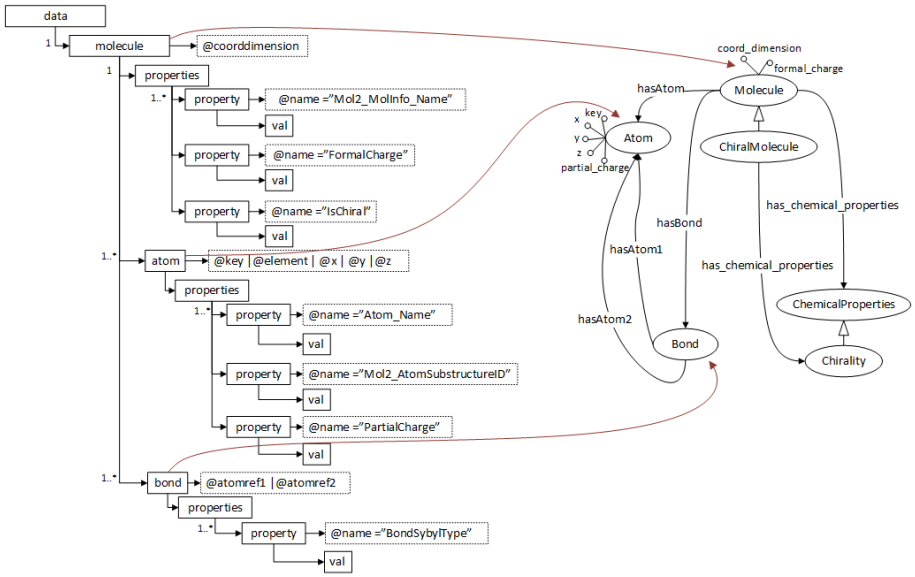

Figura 10.23: Esquema XSD para las librerías de moléculas (izqda.) y correspondencias con la ontología OWL del dominio (dcha.)

La figura 10.23 muestra la representación del esquema XSD y de la ontología OWL construidos y parte de las correspondencias definidas para realizar la transformación.

La transformación de este caso de uso se puede hacer por completo utilizan reglas básicas de transformación exceptuando la propiedad química de quiralidad de las moléculas, para la que es necesaria un patrón que enlace aquellas moléculas cuya propiedad *IsChiral* sea igual a 1 con el individuo existente en la ontología *chirality*, como se explicó en el capítulo 7 (sección 7.2.1). Sin embargo, se pueden definir patrones para cada una de las entidades principales de la ontología, *Molecule*, *Bond* y *Atom*, para agilizar la definición de correspondencias.

La construcción del patrón debe estar guiada por la definición de las entidades en la ontología. Para *Atom*, la figura 10.24 izquierda muestra los axiomas que definen a la clase en la ontología. A partir de dicha definición se puede construir el patrón mostrado en la parte derecha de la figura.

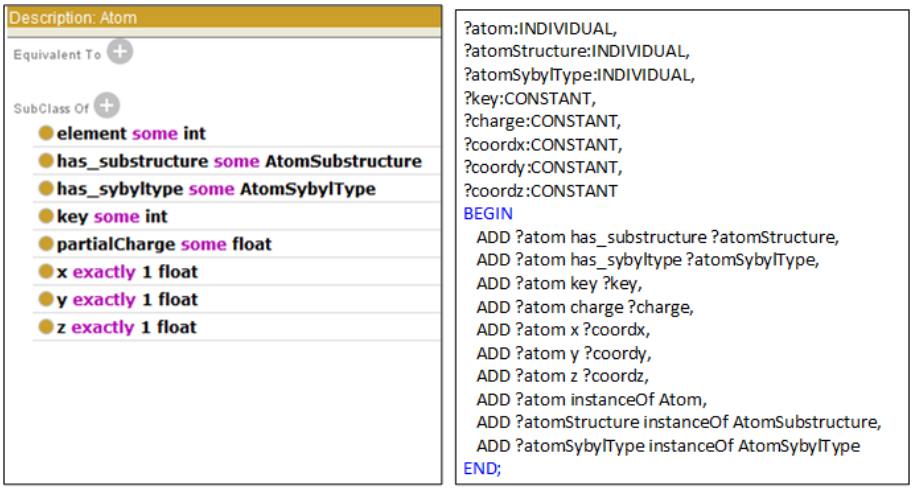

Figura 10.24: Axiomas de clase y patrón para la entidad *Atom*

La figura 10.25 muestra el patrón construido (derecha) a partir de la definición de *Bond* en la ontología (izquierda). Todos los patrones tienen asociadas las plantillas de reglas que crean, para este caso una plantilla de regla de clase para *?bond*, otra para *?sybyl* y tres plantillas de reglas de relación para *?bond* con *?atom1*, *?atom2* y *?sybyl*. No es necesario definir una regla de clase para *?atom1* y *?atom2*, pues son instancias de *Atom* y ya se crean con su propio patrón.

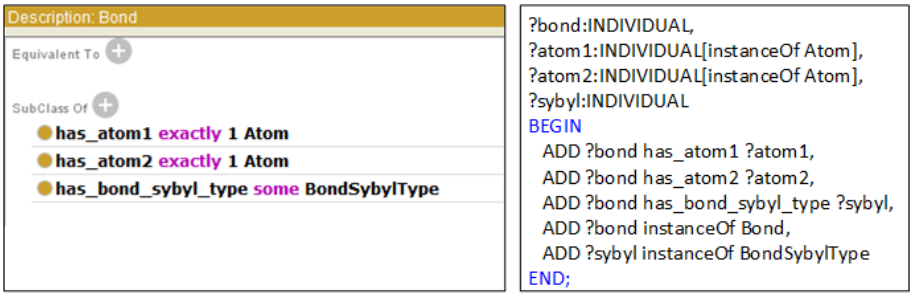

Figura 10.25: Axiomas de clase y patrón para la entidad *Bond*

La figura 10.26 muestras la reglas de relación generada al enlazar las variables *?bond* y *?atom1*. La primera clase define una regla de clase para la variable *?bond*. En el apartado `<infos>` se definen los campos que han de ser usados para crear la URI de las nuevas instancias. En este caso, las URI de las instancias de *Bond* se crean con la combinación de los identificadores de

los dos átomos que enlazan y el nombre de la molécula a la que pertenecen.

```
<map>
    <type>2Rel</type>
    <domain>
        <class>
            <id>http://www.coode.org/oppl/variablemansyntax#?bond</id>
        </class>
        <entity>
            <nodes>
                <node id="1">
                    <id>bond</id><base>/datadoc/data/molecule</base>
                </node>
            </nodes>
            <infos>
                <info><id>@atomRef1</id><base nodeRef="1"/></info>
                <info><id>@atomRef2</id><base nodeRef="1"/></info>
                <info>
                    <id>val</id>
                    <base>
                    /datadoc/data/molecule/properties/property[@name="Mol2_MolInfo_Name"]
                    </base>
                    <base nodeRef="1"/>
                </info>
            </infos>
        </entity>
    </domain>
    <predicate>
        <id>http://miuras.inf.um.es/ontologies/molecule.owl#has_atom1</id>
    </predicate>
    <range>
        <class>
            <id>http://www.coode.org/oppl/variablemansyntax#?atom1</id>
        </class>
        <entity>
            <nodes><node id="2"><id>@atomref1</id><base nodeRef="1"/></nodes>
            <infos>
                <info><id>.</id><base nodeRef="2"/></info>
                <info>
                    <id>val</id>
                    <base>
                    /datadoc/data/molecule/properties/property[@name="Mol2_MolInfo_Name"]
                    </base>
                    <base nodeRef="2"/>
                </info>
            </infos>
        </entity>
    </range>
</map>
```

Figura 10.26: Regla de clase para la relación entre *Bond* y uno de sus *Atom*

La figura 10.27 muestra a la izquierda la definición de *Molecule* en la ontología, y a la derecha el patrón diseñado para crear nuevas instancias de *Molecule* durante el proceso de transformación. El patrón resultado no incluye referencias a las propiedades químicas (*ChemicalProperties*), pues es información que se añade posteriormente con otros patrones, como puede ser el de la quiralidad. Tampoco incluye referencias a las entidades *Bond* y *Atom*, en este caso, añadirlas o no es una decisión de diseño y se decide no hacerlo

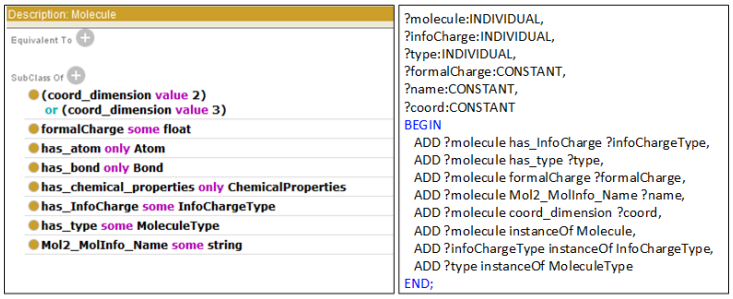

Figura 10.27: Axiomas de clase y patrón para la entidad *Molecule*

por aportar claridad al patrón. Crear la relación de ambas con *Molecule* sólo requiere añadir dos reglas básicas de relación en el proceso de transformación. La figura 10.28 muestra la regla básica que relaciona *Molecule* con *Bond*.

```
<map>
    <type>2Rel</type>
    <domain>
        <class><id>http://miuras.inf.um.es/ontologies/molecule.owl#Molecule</id></class>
        <entity>
            <nodes><node id="1"><id>molecule</id><base>/datadoc/data</base></node></nodes>
            <infos>
            <info>
                <id>val</id>
                <base>
                    /datadoc/data/molecule/properties/property[@name="Mol2_MolInfo_Name"]
                </base>
                <base nodeRef="1"/>
            </info>
        </infos>
        </entity>
    </domain>
    <predicate><id>http://miuras.inf.um.es/ontologies/molecule.owl#has_bond</id></predicate>
    <range>
        <class><id>http://miuras.inf.um.es/ontologies/molecule.owl#Bond</id></class>
        <entity>
            <nodes><node id="2"><id>bond</id><base nodeRef="1"/></node></nodes>
            <infos>
                <info><id>@atomRef1</id><base nodeRef="2"/></info>
                <info><id>@atomRef2</id><base nodeRef="2"/></info>
                <info>
                    <id>val</id>
                    <base>
                        /datadoc/data/molecule/properties/property[@name="Mol2_MolInfo_Name"]
                    </base>
                    <base nodeRef="2"/>
                </info>
            </infos>
        </entity>
    </range>
</map>
```

Figura 10.28: Regla de relación*Molecule* - *Bond*

Bloque III

Discusión, conclusiones y trabajo futuro

Capítulo 11

Discusión y conclusiones

11.1 Discusión y Trabajo Futuro

Las características de la información biomédica y sus necesidades de explotación requieren mecanismos para facilitar su acceso de forma integrada. Hoy en día, se considera que la Web Semántica proporciona un espacio natural para la integración y explotación de datos biomédicos [17]. Entre las iniciativas actuales de la Web Semántica, los esfuerzos de Linked Open Data [20] persiguen la publicación y compartición de conjuntos de datos biomédicos abiertos utilizando formatos semánticos. Berners-Lee [110] sugirió un esquema de desarrollo de cinco estrellas para datos abiertos (Open Data), donde los niveles más altos se consiguen haciendo uso de tecnologías de la web semántica. La construcción de este tipo de conjuntos de datos se ve obstaculizada por la gran cantidad de datos biomédicos disponibles y su heterogeneidad, por lo que se requieren técnicas que proporcionen asistencia en los procesos de creación.

Para llevar al espacio tecnológico de la Web Semántica los conjuntos de datos biomédicos, se requieren métodos de transformación que se apliquen sobre los distintos repositorios biomédicos existentes. Estos métodos deben ser genéricos para que puedan aplicarse a los distintos tipos de repositorios existentes. En esta tesis se diseña un proceso de transformación basado en la definición de reglas de correspondencia entre las estructuras genéricas de entrada y salida de los datos de entrada. En el modelo de transformación aquí planteado, tanto la representación de entrada como las representaciones de salida aceptan esquemas semi-estructurados o estructurados, en los que entidades, relaciones, atributos y asociaciones son fácilmente identificables.

Si se instancia el modelo de transformación a un modelo de salida dado por una ontología OWL, surge también la posibilidad de transformar datos

como clases OWL en lugar de como instancias OWL. La utilización de un modo u otro depende del caso de uso y de la explotación de los datos, por lo que la flexibilidad del modelo de transformación es importante para obtener el repositorio semántico deseado.

La definición de correspondencias entre esquemas de entrada y de salida puede ser una tarea compleja. El modelo de transformación permite definir patrones de diseño que encapsulan la definición de instancias en el modelo de salida, favorecen la reutilización de correspondencias y reducen el esfuerzo de definirlas, mientras que la herramienta SWIT, que implementa el modelo de transformación para la creación de repositorios OWL/RDF, trata de ofrecer una interfaz amigable que ayude en su definición. Sin embargo, queda la cuestión de quién define las correspondencias y los patrones. De la correcta definición de estos depende la obtención de un resultado satisfactorio en los métodos de transformación e integración, y la resolución de los problemas por inconsistencias en los datos de entrada. La formación de los administradores de datos en informática de la salud debe aumentar para explotar lo mejor de las tecnologías semánticas abiertas, sin embargo, un usuario experto puede encontrarse con el problema de que el número de correspondencias a definir es muy elevado. Queda como trabajo futuro el uso de técnicas de definición de correspondencias automáticas que ayuden al usuario y reduzcan el tiempo necesario para esta actividad.

Las reglas de identidad controlan la redundancia de las nuevas instancias creadas y son una pieza clave de los procesos de integración. La definición de las reglas requiere la identificación de los atributos y relaciones de una entidad que describan cómo se distinguen unas instancias de otras. Por lo tanto, la correcta definición de las reglas requiere la existencia de atributos y relaciones que le den a la entidad su cualidad de identidad en el mismo sentido que se define en las ontologías formales [249]. Las propiedades que dan la identidad no tienen nada que ver con la pertenencia a clase, por lo que las reglas de identidad no se modelan como los criterios de necesario y suficiente, sino que son aquellas propiedades únicas para la instancia que permiten distinguir una instancia del resto de instancias de la clase. En una ontología OWL, las propiedades utilizadas en la regla de identidad de una clase son aquellas que se utilizarían en una axioma *Key* [250]. Estos axiomas asocian a una clase un conjunto de `owl:ObjectProperty` y `owl:DatatypeProperty`, de manera que cada instancia de dicha clases se identifica de forma única por los valores de dicho conjunto de `owl:ObjectProperty` y `owl:DatatypeProperty`, de manera que si dos instancias de la clase coinciden en los valores de todas las propiedades de su axioma *Key*, son consideradas la misma instancia. Los axiomas *Key* no se aplican sobre individuos que no han sido explícitamente

nombrados, por lo tanto estos axiomas no afectan para los casos en los que la pertenencia a una clase es inferida. La limitación de estos axiomas en los procesos de inferencia hace necesario acudir a otros métodos para identificar instancias equivalentes.

La implementación del modelo de transformación en SWIT comprueba todos los aspectos formales que garantizan la generación de contenido consistente, independientemente del caso de uso y de la explotación posterior de los datos. Como consecuencia, el número de operaciones y axiomas que deben crearse por cada nueva instancia puede hacer que el tiempo de transformación sea muy largo para conjuntos de datos de tamaño medio y grande. Sin embargo, dependiendo de la explotación que se vaya a realizar de los datos transformados, se pueden relajar algunas de las condiciones del proceso de transformación. Por ejemplo, para conjuntos de datos independientes o para aquellos casos que la identidad de la instancia queda garantizada por la construcción de su URI, las reglas de identidad pueden no ser necesarias o en caso de que no se requiera razonamiento automático, algunos axiomas no necesitan generarse. Por ejemplo, puede no ser necesario añadir el axioma `owl:differentFrom` a cada una de las instancias creadas con respecto al resto, y por lo tanto, el tiempo de ejecución se ve reducido, según el estudio de complejidad realizado en el capítulo 7 (sección 7.2.2), en $O(i_t)^2$, dónde i_t es el máximo número de nuevas instancias creadas.

La instanciación del modelo de transformación a un modelo de salida definido por una arquitectura ontológica, formada por una ontología OWL y patrones de diseño de contenido ontológico, permite crear un modelo de integración guiado por la transformación de recursos heterogéneos a un modelo común, es decir por el dominio de salida, e independiente de la estructura de los recursos de entrada.

El problema del tamaño también aparece con la integración de repositorios. A pesar de que el modelo de integración aquí presentado no se orienta a la transformación e integración completa de repositorios de entrada, sino a una integración guiada por un dominio de aplicación final, que selecciona sólo aquellos datos de entrada que pertenecen al dominio de salida, la integración se realiza en un repositorio físico común y se pueden generar repositorios finales muy grandes que pueden llevar a problemas de eficiencia en el procesado de los mismos, por ejemplo, limitaciones en los procesos de inferencia. Estas limitaciones de tamaño vienen dadas por las prestaciones de los sistemas que alojan los repositorios finales.

Finalmente, la disponibilidad de representaciones OWL para modelos clínicos de HCE, junto a métodos semánticos de gestión de contenido y los modelos de transformación e integración diseñados, permiten la creación de

una plataforma integrada donde los modelos clínicos pueden ser gestionados junto a los datos clínicos, y la semántica de ambos puede ser explotada junto a recursos biomédicos externos. Esta plataforma se implementa en la herramienta ArchMS.

El sistema gestor ArchMS, es un gestor de modelos y datos clínicos que incluye los modelos de transformación e integración aquí presentados para la explotación del conocimiento clínico junto a otros recursos biomédicos. Al contrario que otros gestores de modelos clínicos, hace uso de la Web Semántica. Otros sistemas, como CKM, LinkEHR o CIMM, se basan en la tecnología ADL y se orientan al soporte de la construcción y publicación de arquetipos. La especificación del modelo de arquetipos no ha sido pensada considerando la web semántica, prueba de ello es que los arquetipos no tienen URI, que hacen de identificadores de recursos en la Web Semántica. En ArchMS se genera una URI para cada arquetipo representado en OWL, además, no está orientado a la creación de arquetipos, si no a su explotación junto a extractos clínicos de datos. La principal ventaja de ArchMS con respecto a otros sistemas de gestión es el uso de tecnologías OWL, que permiten la combinación del modelo de información, modelos clínicos y terminologías y la explotación integrada de los distintos recursos clínicos.

ArchMS utiliza las ontologías de diferente forma: como vocabulario controlado, esquema de conocimiento, búsqueda consistente, clasificación de instancias, reutilización e inferencia, estando estos usos entre las mayores aplicaciones de las ontologías según [51]. Uno de los mayores usos de ontologías en el dominio biomédico es la anotación, siendo la Gene Ontology (GO) una de las más populares. En este uso, las ontologías son explotadas como vocabularios controlados, ya que las clases de la ontología se utilizan principalmente como entidades de anotación. Los enlaces terminológicos de los arquetipos no deben ser confundidos con las anotaciones proporcionadas por el sistema. Los enlaces terminológicos se añaden a los términos o elementos del arquetipo durante su construcción, mientras que ArchMS no pretende dar soporte al diseño y desarrollo de arquetipos. Las anotaciones proporcionadas por el sistema deben entenderse como metadatos del arquetipo, que están asociadas al arquetipo como un todo y no con términos individuales. ArchMS anota los arquetipos de dos formas diferentes: (1) búsqueda textual; (2) similitud de arquetipos.

Por un lado, la descripción textual del arquetipo se procesa y se utiliza para obtener coincidencias entre los términos de terminologías biomédicas. Esta aproximación es de utilidad y permite obtener una primera versión del arquetipo anotado, sin embargo, mucho peso de la anotación recae en el usuario y existen métodos específicos de anotación automática de arquetipos

[251; 252] cuya integración en ArchMS puede ser interesante.

Por otro lado, las anotaciones se basan en la similitud semántica de arquetipos calculada aplicando funciones de similitud semántica. Dicha medida de similitud es un ejemplo del uso de ontologías como esquema del dominio, ya que las clases y propiedades de las ontologías del modelo de información y de arquetipos (ontologías utilizadas por PoseacleConverter) se utilizan en el cálculo. Sin embargo, esta medida no requiere utilizar razonamiento automático. Las funciones de similitud semántica son investigadas por el usuario especificando los valores de umbral y peso, pero no hay forma automática o estándar para determinar los mejores valores para los pesos. En general, un valor mayor del peso significa que se da más importancia a dicho factor sobre los otros. Se considera que el peso de la similitud lingüística debería ser el más pequeño porque no proporciona realmente información de la estructura particular o significado de la entrada de conocimiento. Debido a que se están comparando clases en una ontología, la distancia taxonómica debería ser considerada la más importante. Sin embargo, de la aplicación de este método, también utilizado en la recomendación de recursos formativos, en los casos de uso se hace evidente que el valor de los pesos depende del origen de las anotaciones, y la posibilidad de definir distintos pesos a las anotaciones de una misma entidad resulta la opción más eficiente, es decir, se deben hacer decisiones locales según la naturaleza local del conocimiento. Este mecanismo basado en pesos y umbrales permite a cada grupo de usuarios en ArchMS obtener los resultados que se ajusten a su noción de similitud, sin embargo, requiere un conocimiento adecuado de las tecnologías semánticas por parte del usuario para la selección de los mejores parámetros. Una línea de trabajo futuro sería el aprendizaje de los valores óptimos de parámetros dependiendo de las propiedades de los arquetipos comparados y el tamaño de las ontologías utilizadas en las anotaciones del arquetipo.

El razonamiento con ontologías es explotado en ArchMS con arquetipos y datos. El servicio de validación de arquetipos incluido en ArchMS, llamado Archeck, comprueba la corrección de los arquetipos especializados aplicando razonamiento automático sobre la representación OWL de los arquetipos. Archeck utiliza una representación basada en clases OWL de los arquetipos, mientras que PoseacleConverter los representa como individuos OWL debido a que realizan tareas con distintos propósitos. El propósito original de la representación OWL de arquetipos en PoseacleConverter era soportar la transformación entre estándares, para la que está demostrado que es efectiva. Ninguna de las propuestas de representación OWL se presentan como estándares, si no como soluciones tecnológicas apropiadas para las distintas actividades semánticas ejecutadas en el sistema.

Por otro lado, el razonamiento automático es utilizado para la clasificación de pacientes. Dicha actividad se ejecuta sobre los datos de los pacientes importados a ArchMS como extractos XML y transformados a RDF/OWL utilizando el método de transformación implementado en SWIT. Una vez que los extractos de HCE se transforman a instancias, se utiliza inferencia para clasificar dichas instancias. Por ejemplo, en el caso de uso de cribado de cáncer de colon y recto, los datos de los pacientes se clasificaban según el nivel de riesgo siguiendo los protocolos europeo y americano. Esta transformación de datos permite moverlos del espacio tecnológico de los arquetipos al espacio tecnológico de la Web Semántica. Actualmente, los datos transformados no conservan información sobre la estructura de los arquetipos, ya que la transformación está dirigida principalmente por la ontología de dominio. En el futuro se pretende realizar pruebas con la transformación del arquetipo para investigar que aproximación de transformación puede ser más apropiada según las diferentes tareas.

11.2 Verificación de la hipótesis

La hipótesis de esta tesis es que el uso de tecnologías de la Web Semántica permite generalizar la integración de información biomédica heterogénea y facilita la gestión de modelos y datos clínicos y la explotación de información. Esta hipótesis está formada por sub-hipótesis que han sido demostradas a través de la respuesta a las cuestiones en las que se descomponen y que se exponen a continuación.

11.2.1 Sub-hipótesis 1

Es posible la definición de un método de transformación de información biomédica guiado por el dominio de salida a través del uso de reglas de transformación y patrones de diseño.

¿Qué representaciones son las más comunes en los sistemas de información biomédica? En el capítulo 2 se exponen los formatos más utilizados en la representación de la información biomédica, distinguiendo entre información clínica y biológica. En los sistemas de información clínica, la información se almacena en el historial clínico del paciente. Entre los estándares y especificaciones disponibles para definir la arquitectura de los sistemas de historia clínica electrónica (HCE) destacan aquellos basados en el modelo dual, que distinguen entre el modelo de referencia, que estructura la información, y el modelo de conocimiento, para la representación del conocimiento clínico. En el capítulo se exponen los estándares y especificaciones

que se basan en el modelo dual CEN/ISO 13606, openEHR, HL7 y CEM, en todos ellos, los modelos clínicos se definen con lenguajes con orientación sintáctica, como ADL o CDL, y los extractos clínicos se almacenan en ficheros XML.

Los datos biológicos se presentan más comúnmente en ficheros de texto plano, ficheros estructurados como XML, bases de datos relacionales y bases de datos basadas en grafos.

Tanto en los repositorios clínicos como biológicos, el uso de terminologías biomédicas es común para la anotación de entidades, aportan estandarización de los términos utilizados y facilitan la reutilización de los datos.

La variedad de propuestas de representación y terminologías disponibles dificultan el acceso, gestión e interpretación de la información biomédica e impiden la consecución de la interoperabilidad semántica. Lenguajes como ADL, utilizados en la representación de arquetipos, o XML, muy extendido para la representación de recursos biológicos, tienen una orientación sintáctica y se muestran poco flexibles e insuficientes para realizar tareas que requieren la explotación de la semántica de la información.

¿Cuáles son los métodos de transformación de recursos a representación semántica y qué problemas tienen asociados? Distintas iniciativas y estudios proponen el uso de las tecnologías de la Web Semántica para la representación, gestión, integración de la información biomédica. Por un lado, las ontologías permiten crear completos modelos de conocimiento y se consideran claves en la consecución de la interoperabilidad semántica. Por otro lado, la Web de Datos, surgida a partir de la aplicación de los principios de Linked Data, permite publicar y compartir conjuntos de datos biomédicos.

En el capítulo 4 se exponen diferentes herramientas utilizadas para obtener una representación semántica de basada en ontologías OWL o en RDF de recursos de información. La metodología de transformación seguida por las herramientas sigue un esquema común, parten de un repositorio de datos fuente representados según un modelo de entrada, definen correspondencias entre dicho modelo de entrada y el modelo destino. Dichas correspondencias guían el proceso de extracción y transformación. En algunos casos el modelo semántico destino existe previamente y en otros se genera a partir del recurso de entrada.

Herramientas como D2RQ, Triplify, Virtuoso Linked Data Views o XS-2OWL generan una representación OWL o RDF a partir de bases de datos relacionales o archivos XML. La generación del repositorio de salida es automática o semi-automática y se trata de una transformación guiada por el esquema lógico de la representación origen del contenido. Otras herramientas como RDB2OWL o Karma realizan una transformación guiada por el

dominio, utilizando una ontología pre-existente en el proceso de definición de correspondencias, sin embargo, tienen problemas de complejidad en la definición de las correspondencias y están orientadas a un formato de representación específico.

¿Qué componentes definen un modelo de transformación genérico? En el capítulo 7 de esta tesis se expone la primera propuesta, un modelo genérico de transformación de recursos de información. El modelo de transformación se compone de: un modelo de entrada, que define la representación de las instancias en el recurso de entrada; un modelo de salida, que define la representación destino de las instancias; y reglas de transformación, que guían el proceso de transformación y se dividen en las reglas de correspondencia y las reglas de identidad.

Los modelos de entrada y salida se estructuran según un metamodelo que define los componentes mínimos que estos deben contener para ser utilizados en el modelo de transformación. Este modelo está dirigido a aceptar recursos de entrada con información estructurada, como bases de datos relacionales, ficheros XML o extractos clínicos basados en arquetipos.

Las reglas de correspondencia definen la asociación entre el modelo de entrada y el modelo de salida y permiten definir las instancias del modelo de entrada como instancias del modelo de salida. Para definir transformaciones de mayor complejidad, el modelo de transformación incorpora patrones de diseño que representan parte o la totalidad de la definición de una entidad en el modelo de salida, lo que crea una vista que aísla al usuario de la complejidad del modelo de salida.

Las reglas de identidad definen las propiedades y relaciones que distinguen a una instancia en el modelo de salida de cualquier otra, de manera que se utilizan para identificar instancias redundantes.

Como las reglas de transformación se definen utilizando los componentes de los modelos de entrada y salida, cualquier esquema cuyo modelo cumpla con el metamodelo definido puede ser utilizado en el modelo de transformación. Si se utiliza como modelo de salida una ontología OWL se realiza una transformación dirigida por la semántica del dominio de salida.

Como resultado se obtiene un modelo de transformación genérico para cualquier modelo de entrada y salida que cumplan las características del metamodelo definido, flexible en la definición de correspondencias que guían la transformación y con mecanismos que facilitan las transformaciones más complejas y controlan los problemas de redundancia.

¿Qué ventajas y facilidades traen el uso de reglas de transformación y patrones de diseño en el modelo de transformación? Por un lado, las reglas de transformación flexibilizan el proceso, pues permiten

realizar distintas transformaciones para una misma fuente dependiendo del dominio de salida. Los patrones, definidos en la implementación del modelo en el lenguaje OPPL 2, facilitan la creación de entidades complejas. El patrón contiene todos los detalles de propiedad y asociaciones que caracterizan a la entidad, mientras que parametriza aquellos valores que dependen del recurso de entrada, de esta manera el usuario no necesita conocer en profundidad el modelo de salida, que puede ser muy complejo, y puede definir las reglas de correspondencia de forma guiada por el patrón. Por otro lado, la definición de reglas de identidad, que se utilizan para definir las propiedades que distinguen a una instancia en el modelo de salida de cualquier otra, se utilizan para la identificación de instancias que se consideran redundantes por ser semánticamente idénticas.

11.2.2 Sub-hipótesis 2

La aplicación del modelo de transformación genérico a la transformación a una representación OWL permite definir un proceso de integración genérico para información proveniente de fuentes heterogéneas.

¿Qué técnicas de integración de recursos heterogéneos existen y cuáles son sus problemas asociados? En el capítulo 5 se presentan arquitecturas y sistemas de integración existentes. Las arquitecturas de integración más comunes se clasifican en almacenes de datos, sistemas basados en mediadores y sistemas basados en enlaces.

Con la llegada de las tecnologías de la Web Semántica se hizo muy común el uso de las ontologías en el modelado del esquema global de integración, además de que la aparición de los principios de Linked Open Data (LOD) fomentó la publicación de recursos en formato semántico siguiendo estos principios. Los sistemas de integración en el dominio biomédico expuestos en el capítulo 5 son soluciones propias creadas específicamente para los recursos concretos a integrar. Los recursos biomédicos son muy heterogéneos, por lo que los sistemas deben resolver los problemas de conflictos entre esquemas y datos que surgen en la realización de la integración. Existe una carencia de soluciones genéricas para la integración de recursos y que además no requieran de una intervención totalmente manual para resolver los problemas de conflictos que surgen entre las fuentes.

¿Cómo se pueden generalizar los procesos de integración para que sean aplicados a cualquier recurso de información? En el capítulo 8 se presenta el modelo de integración definido en esta tesis, basado en la transformación de los recursos heterogéneos a un modelo global basado en una arquitectura ontológica. Esta arquitectura está formada por una on-

tología OWL junto a patrones de diseño de contenido ontológico. Como el modelo de transformación está guiado por la definición de correspondencias sobre la ontología de salida y los patrones, se obtiene una integración basada en la transformación guiada por el dominio de aplicación e independiente de la estructura de los recursos de entrada.

Por lo tanto, la integración a través de la transformación permite obtener un modelo de integración genérico, aplicable a cualquier modelo de entrada que cumpla los requisitos del modelo de transformación, y flexible en la creación del repositorio integrado final, que depende del modelo de salida.

¿Cómo mejoran las tecnologías de la Web Semántica la integración de recursos heterogéneos? La ontología global utilizada como modelo de salida ofrece un vocabulario común de representación, por lo que la definición de las reglas de correspondencia que enlazan los esquemas de entrada con la ontología de salida permiten resolver los problemas de conflictos de nombrado. El uso conjunto de las reglas de transformación con los patrones de diseño permiten solventar con facilidad problemas de inconsistencias entre el esquema de entrada y salida. El uso de una arquitectura ontológica, proporciona la semántica explícita en la representación de las instancias que facilita la correcta definición de las reglas de identidad. De esta manera, se pueden identificar instancias redundantes entre las fuentes y localizar las inconsistencia entre las mismas.

La ontología aporta un formalismo de representación común para todos los recursos a integrar, y al guiar la transformación permite obtener la representación semántica del contenido origen, que favorece su posterior explotación. El modelo de integración expuesto utiliza ontologías OWL que permiten el uso de inferencia para comprobar la consistencia del recurso integrado final.

11.2.3 Sub-hipótesis 3

Mediante la aplicación del proceso de integración definido y métodos basados en tecnologías de la Web Semántica se facilita la explotación integrada del conocimiento incluido en los recursos biomédicos y el uso secundario de la información.

¿Qué tareas son clave en la gestión, explotación y uso secundario de la información biomédica? La investigación en las distintas disciplinas de la biomedicina requiere el acceso a información distribuida en distintos sistemas y representada en distintos formatos. La posibilidad de realizar un acceso integrado y homogéneo a la información es clave para dar apoyo a este tipo de investigaciones.

A nivel de sistemas HCE, los sistemas de gestión de arquetipos existen-

tes, algunos de ellos expuestos en el capítulo 2, se orientan a la creación y publicación de nuevos modelos clínicos en una estándar o especificación concreta y utilizan una representación sintáctica de los mismos, lo que dificulta la realización de actividades semánticas para la explotación de estos, como la validación de su consistencia o la comparación semántica de los mismos, que son claves para la interoperabilidad semántica y fomentar la compartición de modelos clínicos entre distintas instituciones.

La explotación completa de los recursos clínico debería permitir su integración con recursos externos para dar soporte a distintos estudios científicos y facilitar su acceso para actividades de identificación de grupos de estudio clínico y evaluación de la calidad de la asistencia sanitaria entre otros.

¿Cómo el uso de representación semántica basada en ontologías OWL facilita las tareas de gestión de información biomédica y su uso secundario? En el capítulo 9 se presenta la última propuesta de esta tesis, una plataforma de gestión de información biomédica relacionada con HCE junto a recursos biomédicos externos.

A través de diferentes representaciones OWL de arquetipos clínicos ADL, la plataforma proporciona nuevas actividades semánticas de gestión y explotación, que incluyen la transformación entre estándares CEN/ISO 13606 y openEHR, la validación de la consistencia de los arquetipos con respecto al modelo de referencia y al arquetipo padre, la comparación semántica entre arquetipos, la anotación con terminologías externas y la construcción de sus perfiles semánticos. Estos métodos permiten gestionar un repositorio de arquetipos de forma más avanzanda, facilitando la búsqueda y validación de arquetipos atendiendo a su semántica. La plataforma acepta extractos de datos clínicos, que pueden ser explotados junto a los arquetipos involucrados en su recogida.

Por medio de la inclusión de los modelos de transformación e integración en la plataforma se puede realizar una transformación dirigida por el dominio de los datos clínicos a una representación OWL, de manera que pueden ser explotados en nuevos dominios utilizando las técnicas de razonamiento proporcionadas por OWL y pueden ser integrados junto a otros recursos externos.

11.3 Contribuciones

En esta tesis se presentan soluciones para la transformación de la información biomédica a una representación semántica que favorezca su interoperabilidad, integración, gestión y explotación y dé soporte a la investigación para la

consecución de la medicina traslacional. Las soluciones aquí presentadas se han aplicado con éxito en varios escenarios de validación:

- Estudio sobre pacientes del programa de cribado de cáncer de colon y recto, cuyos datos clínicos son representados por medio de extractos clínicos definidos por arquetipos openEHR. Los extractos clínicos son transformados a una representación OWL definida por una ontología de dominio que permite aplicar técnicas de razonamiento y realizar clasificaciones de los pacientes según su riesgo de desarrollar cáncer de colon y recto.

- Transformación entre modelos clínicos CEM y arquetipos openEHR, para favorecer la reutilización de modelos clínicos.

- Integración de recursos heterogéneos sobre genes ortólogos, enfermedades genéticas e información sobre anotación de secuencias genómicas, almacenados en bases de datos relacionales y ficheros XML, mediante su transformación a una representación según una ontología OWL, con el propósito de construir una repositorio RDF integrado.

- Transformación de componentes químicos en ficheros XML a una representación OWL que favorezca la búsqueda de nuevas moléculas.

Las principales aportaciones que se pueden extraer del trabajo presentado son:

- Diseño de un modelo de transformación genérica de datos entre esquemas de representación estructurados. La definición de reglas de correspondencia entre un modelo de entrada y un modelo de salida permiten la transformación de instancias de entrada a una representación según el modelo de salida. La definición de reglas de identidad permite identificar las instancias redundantes. Los modelos de entrada y salida aceptados están definidos por un metamodelo y se incorporan patrones de diseño para realizar transformaciones más complejas.

  Mediante la inclusión de nuevas reglas de correspondencia, este modelo de transformación se adapta a la transformación y creación de entidades en un modelo de salida, en lugar de la transformación y creación de instancias. De la aplicación de esta adaptación a un escenario de validación para la obtención de arquetipos openEHR a partir de modelos clínicos CEM se obtuvo un conjunto de plantillas openEHR OWL y de patrones que permiten realizar esta transformación de forma automática.

- Diseño de un modelo de integración de información biomédica heterogénea. Mediante la instanciación del modelo de transformación a un modelo de salida definido por una arquitectura ontológica formada por una ontología OWL y patrones de diseño de contenido ontológico se integran distintos recursos heterogéneos. El modelo de salida se define de forma independiente a la estructura de los recursos fuente. El resultado es la integración de los recursos guiada por el dominio de aplicación, pues se basa en la transformación de los mismos a un modelo de salida global, independiente de los modelos de entrada.

- Diseño de una plataforma de integración, gestión y explotación de información biomédica. En la plataforma se seleccionan las representaciones OWL más adecuadas para modelos clínicos e incluye métodos semánticos de validación, anotación, comparación y búsqueda que permite realizar una gestión y compartición adecuada de los modelos. La plataforma también incluye los modelos de transformación e integración definidos, lo que permite explotar de forma conjunta los datos recogidos utilizando los modelos clínicos y recursos biomédicos externos.

- Implementación del modelo de transformación, integración y la plataforma de gestión y explotación a través de dos aplicaciones web. SWIT implementa el modelo de transformacion y asiste durante el proceso de transformación de recursos de entrada a repositorios OWL o RDF. ArchMS implementación la plataforma integrada, gestiona los modelos clínicos teniendo en cuenta su semántica, ya sea a través de sus enlaces terminológicos o sus anotaciones externas, y gestiona los datos clínicos junto a estos, incluye los métodos de transformación e integración que posibilitan la creación de repositorios semánticos a partir de los datos clínicos, que pueden ser explotados e integrados con otros recursos.

11.4 Conclusiones generales

La investigación traslacional requiere el acceso de forma integrada a recursos heterogéneos. En el ámbito clínico, distintas propuestas de estándares y especificaciones tratan de favorecer la interoperabilidad semántica de la información, mientras que propuestas como Linked Open Data fomentan la publicación y enlazado de los datos. Sin embargo, la naturaleza sintáctica de los lenguajes utilizados para la representación de modelos clínicos resulta insuficiente para su gestión, mientras que los métodos de publicación de datos almacenados en la Web de Datos realizan una transformación sintáctica del

contenido, guiada por el esquema lógico de la representación origen y existen problemas a la hora de generalizar dichos métodos.

Utilizar modelos globales basados en ontologías OWL para representar los recursos fuente permite definir una transformación dirigida por la semántica del dominio y a su vez utilizar esta semántica para explotar el repositorio final resultado. Una representación OWL permite validar y comparar el contenido atendiendo a su semántica, y facilita la integración de distintos recursos.

Las herramientas desarrolladas han demostrado ser efectivas en su utilización en distintos escenarios de validación, creando repositorios semánticos abiertos que contribuyen al desarrollo de la Web de Datos y permitiendo su explotación en el espacio tecnológico de la Web Semántica.

11.5 Publicaciones y contribuciones en congresos

11.5.1 Publicaciones JCR

- M. C. Legaz-García, M. Menárguez-Tortosa, J. T. Fernández-Breis, C. G. Chute, and C. Tao, "Transformation of standardized clinical models based on OWL technologies: from CEM to OpenEHR archetypes," *Journal of the American Medical Informatics Association: JAMIA*, vol. 22, no. 3, pp. 536-544, Feb. 2015.

- J. T. Fernández-Breis, J. A. Maldonado, M. Marcos, M. C. Legaz-García, D. Moner, J. Torres-Sospedra, A. Esteban-Gil, B. Martínez-Salvador, and M. Robles, "Leveraging electronic healthcare record standards and semantic web technologies for the identification of patient cohorts," *Journal of the American Medical Informatics Association: JAMIA*, vol. 20, no. e2, pp. e288-e296, Dec. 2013.

- M. C. Legaz-García, J. A. Miñarro-Giménez, M. Madrid, M. Menárguez-Tortosa, S. T. Martínez, and J. T. Fernández-Breis, "Linking Genome Annotation Projects with Genetic Disorders using Ontologies," *Journal of Medical Systems*, vol. 36, no. 1, pp. 11-23, Nov. 2012.

11.5.2 Congresos

- M. C. Legaz-García, J. A. Miñarro-Giménez, M. Menárguez-Tortosa and J. T. Fernández-Breis, "Lessons learned in the generation of biome-

dical research datasets using Semantic Open Data technologies," MIE 2015, Stud Health Technol Inform, vol. 210, pp. 165-169, 2015.

- J. T. Fernández-Breis, M. C. Legaz-García, H. Chiba, I. Uchiyama, "Efforts in the semantic standardization of orthology content," Quest For Orthologs meeting 2015.

- M. C. Legaz-García, C. Martínez-Costa, J. A. Miñarro-Giménez, J. T. Fernández-Breis, S. Schulz, and M. Menárguez-Tortosa, "Ontology patterns-based transformation of clinical information," MIE 2014, Stud Health Technol Inform, vol. 205, pp. 1018-1022, 2014.

- C. Martinez-Costa, M. C. Legaz-Garcia, S. Schulz, and J. T. Fernandez-Breis, "Ontology-based infrastructure for a meaningful EHR representation and use," in 2014 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), 2014, pp. 535-538.

- C. Martínez-Costa, D. Bosca, M. C. Legaz-García, C. Tao, J. T. Fernández Breis, S. Schulz, and C. G. Chute, "Isosemantic rendering of clinical information using formal ontologies and RDF," Stud Health Technol Inform, vol. 192, p. 1085, 2013.

- J. T. Fernández-Breis, J. A. Miñarro-Giménez, M. C. Legaz-García, M. Egaña-Aranguren, "Towards Orthology Linked Datasets," Quest For Orthologs meeting 2013.

- M. C. Legaz-García, C. Tao, M. Menárguez-Tortosa, J. T. Fernandez-Breis, and C. G. Chute, "An approach for the Mapping of CEM and OpenEHR Archetypes," presented at the AMIA CRI 2013, 2013.

- M. C. Legaz-García, C. Martínez-Costa, M. M. Tortosa, and J. T. Fernández-Breis, "Exploitation of ontologies for the management of clinical archetypes in ArchMS," in ICBO'12, 2012, p. 1-1.

- M. C. Legaz-García, C. Martínez-Costa, M. Menárguez-Tortosa, and J. T. Fernández-Breis, "Recommendation of standardized health learning contents using archetypes and semantic web technologies," MIE 2012, Stud Health Technol Inform, vol. 180, pp. 963-967, 2012.

- M. C. Legaz-García, C. Martínez-Costa, M. Menárguez-Tortosa, and J. T. Fernández-Breis, "Towards semantic platforms for archetype management: ArchMS," presented at the OpenHealth 2011, Barcelona, Spain, 2011.

- M. C. Legaz-García, J. A. Miñarro-Giménez, M. Madrid, S. T. Martínez, and J. T. Fernández-Breis, "Using Ontologies for Supporting Genomic Sequence Annotation Projects," in Proceedings of the 2Nd ACM Conference on Bioinformatics, Computational Biology and Biomedicine, New York, NY, USA, 2011, pp. 617-625.

- J. T. Fernandez-Breis, M. Madrid, J. A. Miñarro-Giménez, M. del C. Legaz-García, M. Egaña, and R. Stevens, "Exploitation of Semantic Gene Ontology Annotations for Schizosaccharomyces Pombe," presented at the 6th International Fission Yeast Meeting, 2011.

Bloque IV

English

Chapter 12

Summary

12.1 Introduction

Biomedical informatics is the interdisciplinary field that studies and pursues the effective application of biomedical data, information, and knowledge for scientific inquiry, problem solving, and decision making, driven by efforts to improve human health [1]. Biomedical informatics is divided in four research areas: bioinformatics, biomedical imaging informatics, clinical informatics and public health informatics. The intensive collaboration between those four areas are key for achieving translational medicine, defined as the effective translation of results generated by progressions in basic science research (bench-based experiments) to their clinical validation in bedside clinical trials, ultimately leading to new approaches in clinical practice and an efficient health system [2; 3].

The complete achievement of translational medicine has to face several challenges, which are grouped in three identified translational barriers: (1) translation of innovations from bench-based experiments to validation in bedside clinical trials, ultimately (2) leading to new approaches adopted in the community and potentially leading to (3) the establishment of new policies and better health system [4]. The combined use of approaches from the four areas of biomedical informatics allows crossing these translation barriers. The collaborations between the different areas can be organized in two categories: (1) translational bioinformatics and (2) clinical research informatics.

Translational bioinformatics uses and extends the concepts and methods from bioinformatics to facilitate the practice of translational medicine, i.e., the translation of biological discoveries from the laboratory into new findings in clinical care [5], that is, it is focused on crossing the first barrier. Clinical research informatics focuses on approaches from biomedical informatics that

allow crossing the second and third barriers, that is, making innovations from clinical trials available for a better care of patients and population, integrating them in an efficient healthcare system. It focuses on clinical research, including investigation of the mechanisms of human disease, therapeutic interventions, clinical trials, development of new technologies, epidemiology, behavioral studies, and outcomes and health services research [6].

Due to the broad scope of application of biomedical sciences, biomedical informatics has to deal with a vast amount of data. The number and size of biological databases grow at an increasing rate. Nowadays, according to the *215 Molecular Biology Database Collection* report [7], there are more than 1500 biological databases, which use different representation formats for their biological data. Therefore, the retrieval and management of data are not easy tasks for researchers, because they need to know: (1) which resources are available and have the required information; (2) how those resources can be accessed and searched; (3) the meaning of data types and fields used in each resource. On the medical and clinical side, the advent of electronic health records (EHRs) is also contributing on making more data available for computer processing and promotes the secondary use of the data, but it creates new problems. The secondary use includes activities like rapid cohort identification, quality of care assessment, comparative effectiveness research, data privacy and de-/re-identification research, phenotyping methodology and predictive modelling [8]. Some of these activities demand combining data that is normally spread around different clinical systems, requiring semantic interoperability between such systems, that is, a complete access to the data, communication and understanding of the information independent of its origin. The lack of semantic interoperability becomes a reason for inefficiency in healthcare systems [9; 10] and has a cost of one billion dollars in the United States annually [11].

Therefore, the collaboration between different disciplines of biomedicine requires a representation of biomedical information that allows researchers to exploit and combine the data efficiently.

In the last decades, many efforts have addressed the development of EHR standards and specifications, including HL7 [14], openEHR [13], ISO EN 13606 [12] and CEM [15]. Such standards and specifications are based on the dual model architecture, which distinguishes two modelling levels. On the one hand, the information model provides the generic building blocks to structure the EHR information. On the other hand, clinical models are used to specify clinical recording scenarios by constraining the information model structures. In both openEHR and ISO 13606, clinical models are named archetypes and they have been considered a promising way of sharing clinical

data in a formal and scalable way [9]. Their interest is reinforced by the commitment of the Clinical Information Modeling Initiative (CIMI) to use archetypes [89]. ADL has been proved useful as format for expressing and exchanging clinical models but, as described in works like [87], ADL has a syntactic orientation that makes the realization of the semantic activities required in semantic interoperability environments more difficult. For example, checking the correct semantic definition of the archetype (i.e., the constraints of an archetype must be compatible with the ones of its parent archetype) or detecting whether two archetypes are equivalent would require much effort. ADL permits to bind archetype content to terminologies and ontologies, but it does not facilitate their joint exploitation. Moreover, the variety of standards available provokes the use of different standards in different institutions, so we need additional solutions for enabling exchange between different standards.

On the technical side, the Semantic Web [16] describes a new form of Web content meaningful to computers and it has been proposed as a technological space in which biomedical data can be integrated and exploited [17]. There are different basic technologies for the success of the Semantic Web, among which the cornerstone technology is the ontology. An ontology represents a common, shareable and reusable view of a particular application domain [18]. The fact that machines know the meaning of content enables the use of automated reasoning in the Semantic Web, which permits to infer new information or to check the logical consistency of the content. Besides, the Semantic Web community wishes to achieve the Web of Data [19], which would semantically connect datasets distributed over the Internet. More concretely, the Linked Open Data effort [20] pursues the publication and sharing of biomedical datasets using semantic formats.

There are several efforts of using Semantic Web technologies for biomedical information representation and management. In recent years, ontologies have gained momentum in biomedical research, initiated by the usefulness of the Gene Ontology [21] for sharing functional annotations of gene products. Current initiatives like the OBO Foundry [22] publish guidelines for the creation of new ontologies, and repositories like Bioportal [23] contain more than 400 biomedical ontologies, controlled vocabularies and terminologies. Despite most biomedical ontologies have been used for annotating data, recent efforts are promoting the use of semantic technologies for representing biomedical data, for example, the EBI RDF platform [24] or Bio2RDF [25]. Approaches for the representation of clinical models and data from EHR standards have shown their usefulness for interoperability between those standards [26; 27] and the validation of clinical models [28]. Most of such efforts

for representing data in semantic formats have been solved by in-house solutions, implementing resource-specific transformation scripts. Following this approach has some drawbacks: (1) every database team should have knowledge of Semantic Web technologies and languages and take care of the whole transformation process; (2) each database team might be using semantic representation approaches and formats, which would make difficult the interoperability of the resulting semantic datasets. The semantic web community has developed a series of tools that facilitate the automatic generation of semantic content. Some of those tools mainly perform a syntactic transformation of the traditional formats, while those that provide a semi-automatic or manual transformation have a lack of flexibility, and are focused in specific representation formats.

Hence, in this thesis I propose solutions for the semantic transformation of biomedical datasets, guided by a semantic architecture, which will allow to obtain a precise semantic representation of the source dataset. The defined process is generic and does not depend on the formalism used for capturing the data. This process is part of an integration methodology that allows the integration of heterogeneous datasets for their combined exploitation, creating semantic repositories and open datasets following Linked Open Data principles. Finally, I define a platform for biomedical information management and integration, and exploitation of clinical models and data and external biomedical resources, that uses semantic representations and the transformation and integration methods.

12.2 Aims of the thesis

This thesis aims to assist translational researches by improving the standardization and integrated exploitation of biomedical information through the use of Semantic Web technologies. More specifically, the objectives of this thesis can be summarized as follow:

- Design and implementation of a generic data transformation model between structured representation schemata.
- Design and implementation of a heterogeneous biomedical information integration model.
- Design and implementation of a platform for integrating, managing and exploiting biomedical information, allowing integrated access to information from EHRs and external resources.

- Validation of the results through the transformation and integration of heterogeneous biomedical resources, their inclusion in the management platform and realization of activities of secondary use of biomedical information.

12.2.1 Research hypothesis

The main hypothesis of this thesis is that by using Semantic Web technologies, two main advantages can be achieved: the creation of a generic method for integrating heterogeneous biomedical information, and the reduction of the effort on managing biomedical information, including clinical models and data. This hypothesis is divided in the following sub-hypotheses:

- **A transformation model for biomedical information guided by the domain of the output representation can be designed by the definition of transformation rules and design patterns.** To prove this hypothesis requires answering the following questions:

  1. What formats of representation are more common in biomedical information systems?
  2. What methods of content transformation into a semantic representation are available and what are their associated problems?
  3. What components define a generic transformation model?
  4. What advantages brings the use of transformation rules and design patterns in the transformation model?

- **The application of the generic transformation model into an OWL representation allows to define a generic integration process for heterogeneous information sources.** To prove this hypothesis requires answering the following questions:

  5. What integration methods of heterogeneous resources are available and what are their associated problems?
  6. How can integration resources be generalized to be applied into any information resource?
  7. How the use of Semantic Web technologies improve the integration of heterogeneous resources?

- **The application of the defined integrated process and methods based on Semantic Web technologies allows the integrated exploitation of the knowledge included on biomedical resources**

and the secondary use of information. To prove this hypothesis requires answering the following questions:

8. What tasks are key for managing, exploiting and the secondary use of biomedical information?
9. How the use of a semantic representation based on OWL ontologies makes easier the task of management and secondary use of biomedical information?

12.2.2 Methodology

The methodology proposed is based on the analysis of the state of art, the formalization of the proposed methods, their implementation and their validation in an application domain.

- Analysis of the state of art:
  - Biomedical informatics: it involves studying the most common representation formats for biomedical information, including standards and specifications of EHR, focusing on those based on the dual model architecture, such as ISO 13606, openEHR, HL7 and CEM; tools for managing clinical models; common representation of biomedical datasets and biomedical terminologies.
  - Semantic Web: it includes the study of Semantic Web technologies, RDF and OWL languages, Linked Data proposals, and methods of ontology engineering for creating and reusing ontologies. It also involves studying biomedical ontologies, the existent proposals for their creation and management, and their applications to standards and specifications of EHRs and related tools.
  - Methods of content transformation to semantic representation: study of current proposals for obtaining RDF and OWL representations of data stored in repositories with no semantic orientation.
  - Information integration: study of existing proposals for integrating heterogeneous repositories, and the role of ontologies in integration.
- Formalization of the proposal:
  - Generic transformation of data: the transformation is guided by the definition of transformation mappings between input and output schemata and the use of ontological content design patterns.

- Heterogeneous resources integration process: by instantiating the transformation method, selected heterogeneous resources from a biomedical domain are integrated. The output model is defined by a global ontology and ontology design content patterns.
- Selection of suitable semantic representations of clinical models and methods for consistency validation, comparison and annotation.
- Implementation of the proposal with the creation of tools for transforming, integrating, managing and exploiting biomedical information.
- Validation of the proposal by defining several validation scenarios in the biomedical domain. The methods and tool designed will be applied to the following scenarios: study of clinical data from patients of a colorectal screening program for performing automatic classification of the patients; transformation between CEM and openEHR archetypes; creation of an integrated repository about orthologous genes, genetic disorders and information about genomic sequences annotations; transformation of a dataset of chemical components into an OWL representation.

12.3 State of art

The standards, specifications and formats of representation of biomedical information can be categorised according to the provenance of the information. On the one hand, clinical research informatics deals with information related to the health care of patients, represented by Electronic Healthcare Records (EHRs) standards and specifications. On the other hand, translational bioinformatics deals with information stored in biological databases.

The EHR is defined as a repository of patient data in digital form, stored and exchanged securely, and accessible by multiple authorized users [31]. The increasing use of EHRs in our globalized world leads to a situation where patient's health data is spread across different health systems. The need for accessing this data demands semantic interoperability of clinical information, that is, their meaningful communication across EHR systems. This situation has led to the emergence of many efforts that have addressed the development of EHR standards and specifications. Among all the standards and specifications, the most promising are those based on the dual model architecture.

Standards and specifications based on the dual model architecture distinguishes two modelling levels. On the one hand, the information model provides the generic building blocks to structure the EHR information. On the other hand, clinical models are used to specify clinical recording scenarios by constraining the information model structures. ISO EN 13606[12], openEHR [13], Health Level 7 (HL7) [14] and Clinical Element Model (CEM) [15] are examples of standards and specifications for EHR based on dual model. In openEHR and ISO EN 13606, clinical models are named archetypes and the knowledge model is also known as archetype model. Both openEHR and ISO EN 13606 share the archetype model, but differ on the reference model.

The languages normally used for defining clinical model have a syntactic orientation. Archetypes in openEHR and ISO EN 13606 are represented using the Archetype Definition Language (ADL) [41], which is a generic, formal language for representing constraint-based models, including archetypes. CEM models are defined using Constraint Definition Language (CDL), a language with proprietary syntax. Within HL7, CDA [37] is a standard that specifies the structure and semantics of clinical documents and encodes those documents in XML. Furthermore, FHIR [39] is a new generation of HL7 standards for the exchange of clinical information where any exchangeable content is a resource represented using XML or JSON. The problem with languages with syntactic orientation is that make more difficult the realization of the semantic activities required in semantic interoperability environments. For example, detecting whether two clinical models are equivalent or checking the correct semantic definition of an archetype (i.e., the constraints of an archetype must be compatible with the ones of its parent archetype) would require much effort.

Besides, the lack of appropriate tooling for applying and exploiting clinical models and data in semantic interoperability environments is considered a barrier to the adoption of dual-model architectures by the majority of vendors. LinkEHR [43], the Clinical Knowledge Manager (CKM) [42] and Clinical Information Model Manager (CIMM) [44] are examples of archetype-based tools. LinkEHR permits the edition of archetypes, the normalization of legacy data using archetypes and view of EHR extracts from CEN/ISO 13606, openEHR and HL7 standards, among others; CKM permits many archetype management activities with openEHR ADL archetypes and templates; CIMM has the purpose to serve as a a public reference place to publish and locate CEN/ISO 13606 archetypes.

In the bioinformatics side, biological data are characterised by: large volumes of data, produced by genome sequencing project at increasing rates; complexity, because a biological entity holds many relationships with other

ones; volatility, biological data is not static, its knowledge is continually changing and increasing; heterogeneity, the nomenclature use for naming biological entities is not uniform, so there is a widespread and deep issue of synonyms and homonyms; distribution, there is a vast amount of biological resources, which provokes that the information of a biological entity is spread around different databases.

Biological data is available in various formats. One common format is natural language as part of scientific publications. Focusing on more structured formats, a traditional one is tabular files, flat files in which each line corresponds to a record in the database and a concrete character provides the separation between the field/columns. XML files structure the content by means of tags, OrthoXML and SeqXML [61] are example of XML-based standards formats for representing orthology data. Relational and graph-based databases have gained popularity in the last years because they are effective for retrieving data through complex queries. Gene Ontology has a relational format representation [72], whereas Bio4j [75] is a graph-based repository including biomedical data from several databases.

In both clinical and biological sides, the use of biomedical terminologies is fundamental for the interoperability of information. Terminologies try to overcome the inherent heterogeneity of information, providing a standardized vocabulary used for the codification of concepts stored in EHRs and biological databases. On the one hand, in biological domains, terminologies are indispensable artifacts for annotating genomic sequences. On the other hand, clinical models in EHRs are already linked to clinical terminologies by terminological bindings that encoded local terms. A clinical terminology is defined as the standardized terms and their synonyms which record patient findings, circumstances, events, and interventions with sufficient detail to support clinical care, decision support, outcomes research, and quality improvement; and can be efficiently mapped to broader classifications for administrative, regulatory, oversight, and fiscal requirements [83]. One of the most important clinical terminologies is SNOMED-CT [86], the most comprehensive and precise clinical health terminology, developed by the International Health Terminology Standards Development Organisation (IHTSDO), that is now accepted as a common global language for health terms.

The standards, specifications, terminologies and formalisms proposed has the common objective of achieving the semantic interoperability of information. Several international initiatives have appeared for providing solutions for this problem. The Clinical Information Modeling Initiative (CIMI) [89] pursues the promotion of a common format for the interoperability of clinical models and has committed to use archetypes and ADL as the starting point,

together with SNOMED-CT as terminology. The SemanticHEALTH project [10] identified in its final report a roadmap for achieving semantic interoperability of EHR, which identified EHR standards, ontologies and terminologies as key players to achieve this objective. The EU network SemanticHealthNet (SHN) [90] aims to improve semantic interoperability and proposes the representation of the meaning of clinical data by ontologies and semantic content patterns. Such patterns aim at assisting in information modelling, preventing users from fully understanding the underlying, complex, ontological expressions.

The solutions proposed in this thesis are based on the use of Semantic Web technologies. The Semantic Web [16] describes a new form of Web content meaningful to computers. There are different basic technologies for the success of the Semantic Web, among which the cornerstone technology is the ontology. An ontology represents a common, shareable and reusable view of a particular application domain [18]. The Web Ontology Language (OWL) [102] is the *de facto* standard for the implementation of ontologies and enables the precise formalization of data meaning in a way that can be automatically exploited.

In the biomedical domain, more that 400 biomedical ontologies are available in OWL format in repositories like Bioportal [23] and more and more medical terminologies are becoming available in OWL. The study performed in [149] shows the multiple applications and importance of ontologies in biomedical research. Biomedical ontologies are frequently used as (1) source vocabularies to annotate biological datasets, what improves document or data retrieval and query. Such is the case of MeSH, a controlled vocabulary of the U.S. National Library of Medicine (NLM) used for indexing PubMed articles; (2) to exchange information in semantic interoperability and data integration scenarios. Their use within EHRs enables sharing the same medical domain vocabulary to describe clinical information, facilitating semantic interoperability; and (3) for decision support and reasoning, biomedical ontologies formally represent the biomedical domain by providing a set of axioms that define their concepts and how they relate to each other. Their formalization by using a Description Logics-based (DL) language allows performing reasoning and inferring additional information from the formalized one.

One of the most common uses of biomedical ontologies is to perform semantic annotation. Semantic annotation is defined as the process of adding semantic metadata to content. Annotations link the entities of the contents with their semantic descriptions [141]. The process of annotation [142] can be manual, if the user searches in a repository of semantic resources and search

for the terms for annotating; automatic, when a method makes a rule-based selection of terms for annotating without the intervention of the user; semi-automatic, when a method suggests terms for annotating and the user makes a selection among the suggestions.

The adoption of ontologies for annotation provides a means to compare entities based on aspects that would otherwise not be comparable [143]. There are two main approaches for semantic similarity [143; 145; 146]: edge-based approaches count the number of edges in the graph path between two classes (see for instance [145]). These approaches assume that all of the semantic links are equally weighted but, generally, the greater distance from the root is, the more specific the classes are; node-based approaches not only take into account the edges but also the properties of the classes involved (see for instance [147]).

It should be noted that the ontologies already available do not usually meet all our requirements. On the one hand, existing ontologies might not include all the concepts we need. On the other hand, many currently available biomedical ontologies have been designed for annotation purposes and therefore are not suitable for automated reasoning. Consequently, extensions and re-engineering of ontologies are likely to be required. In fact, best practices in ontology engineering recommend to reuse existing ontologies and to create modular ontologies [128]. Such recommendation means that the resulting ontology infrastructure will probably be a networked ontology. In [92] and [134] propose pattern-based design of ontologies as an approach to ontology development. They propose ontology design patterns, which are a reusable modeling solutions that encode modeling best practices. These approach have also been propose for developing axiomatically rich and rigorous bio-ontologies by biologists [136].

Besides, the Semantic Web community wishes to achieve the Web of Data, which would semantically connect datasets distributed over the Internet. More concretely, the Linked Open Data effort pursues the publication and sharing of biomedical datasets using semantic formats. Linked Open Data [110] is a type of open data in which datasets meet four requirements:

- Use URIs as names for things.
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using Semantic Web Standards like RDF and SPARQL.
- Include links to other URIs, so that more things can be discovered.

In recent years, ontologies have gained momentum in biomedical research, initiated by the usefulness of the Gene Ontology [21] for sharing functional annotations of gene products. Despite most biomedical ontologies have been used for annotating data, recent efforts are promoting the use of semantic technologies for representing biomedical data. Some examples are Uniprot RDF [160], the EBI RDF platform [24; 154] or Bio2RDF [25]. Semantic web technologies have also been applied in the EHR domain with different purposes: representation of clinical models and data [87; 178; 177; 176], interoperability of models and data [26; 27]; or checking the semantic consistency of clinical models [28].

It is widely accepted that without Semantic Web technologies in general, and ontologies in particular, is impossible to solve the problem of the semantic heterogeneity of information, and therefore, achieve the semantic interoperability of information [180]. Most of the efforts for representing data in semantic formats already presented have been solved by in-house solutions, implementing resource-specific transformation scripts. This method is inflexible in a domain with large volume of data generated by non experts in informatics. There is consequently a need for methods and tools that contribute to standardize the process of getting biomedical datasets in semantic formats. The Semantic Web community has developed a series of tools that facilitate the automatic generation of semantic content. The process of transformation is similar in all the tools, by defining mappings between the input model followed by the source data and the output model (preexisting or generated from the input). For example Triplify [193] reveals the semantic structures encoded in relational databases by making database content available in semantic format, Virtuoso [195] provides semantic views on different types of contents, and D2RQ [191] permit to exploit non-semantic databases as virtual semantic graphs. Other tools define mappings between relational databases and existent ontologies, for example, RDB2OWL [198] transforms the data from a relational database into a preexisting RDF graph or OWL ontology, and Karma [199] automatically defines the mapping model between a relational database and an ontology, that can be used for transformation process. Some of these tools mainly perform a syntactic transformation of the content, while those where the transformation is guided by an ontology, are inflexible in the definition of mappings between the output and input models, making their application in several scenarios difficult. The joint exploitation of heterogeneous biomedical data requires integration methodologies. These methodologies have to face several problems in the biomedical domain [179; 205; 206; 207]: broad domain of application of data, the autonomy of the biomedical resources, the use of their own representation and

nomenclature in each resource. The three main proposals are integration based on data warehouse, integration based on mediators and integration based on links.

Integration based on data warehouse [202] involves the extraction of content from the sources and its transformation and storage in a common repository following a global schema. This method improves the efficiency of the queries, since eliminates the need of accessing to each single source, but it requires a lot of computing in the integration process and to keep the warehouse updated.

Integration based on mediators [206] keeps the information in the source repositories. A global schema provides a single view over the underlying resources. The mediator receives the queries over the global schema and translates it into queries on the source repositories. Depending on the design of the global schema, there exist two approaches [211]: global-as-view (GAV), where the global schema is defined in terms of the data sources; local-as-view (LAV), where the global schema is defined regardless of the data source, and the mappings between the global schema and the sources are defined by characterizing each source as a view over the global schema.

Integration based on links emerged from the fact that an increasing number of sources are available on the web and require of users that manually browse through several web pages and data sources [179]. This methodology creates a graph in which the entities of different sources are connected by paths. The specific paths essentially constitute workflows in which the output of a source or tool is redirected to the input of the next source until the requested information is reached.

Semantic Web technologies provide a technological space for integrating and exploiting biomedical information [215] where ontologies are a key component. The integration architectures that make use of ontologies, usually follow one of three different approaches [216]:

- Single ontologies: one global ontology provides a shared vocabulary for the specification of the semantics of the domain. All information sources are related to one global ontology.

- Multiple ontologies: each information source is described by its own ontology. It requires the definition of alignment between ontologies to achieve an homogeneous access to the sources.

- Hybrid Approaches: the semantics of each source is described by its own ontology, but in order to make the source ontologies comparable to each other they are built upon one global shared vocabulary provided by a global ontology.

The use of ontologies in integration approaches requires in some case use of ontological alignment methodologies. Ontological alignment is defined as the creation a collection of binary relations between the vocabularies of two ontologies [217]. An alignment is a set of mappings, where each mapping defines the relation between two entities from two different ontologies. There are several languages for defining alignments, they can be defined using OWL, rule languages such as Semantic Web Rule Language [219] or own languages such as EDOAL [220].

12.4 Results

The main results of this thesis are a generic transformation model for generating semantic representation of schema-based resources, an integration methodology based on the transformation model, and a architecture for biomedical data managing and exploiting.

12.4.1 Transformation model

The transformation model takes an input data schema and a dataset represented using such schema, and generates a dataset according to an output data schema. The transformation of the data is driven by the definition of transformation rules. The transformation rules are divided on mapping and identity rules. Mapping rules are defined between the schemata and used for the extraction and transformation of data. Once defined the mapping rules, the transformation approach also takes into account the identity rules than can be defined over the output schema. These rules set the properties and attributes that make unique an entity, allowing to merge different data instances that refer to the same entity and preventing the creation of logically inconsistent content. Therefore, the transformation model is defined by an input and output models, and the transformation rules.

An input model defines how data are represented in the source, that is, which entities are used in order to represent the data instances. The transformation models uses input models defined according to the metamodel $\langle Entity, Attribute , Relation, Association \rangle$, where:

- Entities stand for the set of domain concepts. Such concepts are entities that can be unambiguously identified.
- Relations stand for the set of properties that link two entities.
- Attributes stand for the set of properties that do not link two entities.

- Associations define the links between Entities and Relations and Entities and Attributes. Associations are used to define the structure of the model.

The objective is to obtain a semantic representation of the input data. In the approach, an output model is defined by the metamodel $\langle Entity, Relation, Attribute, Association \rangle$, where each primitive has the same meaning as for input models. Therefore, generally speaking, the same models could be used as input or output models.

The data represented using a schema that corresponds to a valid input model is transformed into data represented using a schema that corresponds to a valid output model. For this transformation to be useful, both schemata domains should have not empty intersection, which should be a practical requirement for the approach. If the input domain is D_I and the output domain is D_O, then:

$$\forall D_I, D_O, D_I \cap D_O \neq \emptyset \tag{12.1}$$

The transformation rules play two major roles in our approach: (1) controlling that the information represented according to the input schema is correctly represented according to the output schema; and (2) preventing redundancy in the generated dataset. For this purpose, two major types of rules are defined in our approach, namely, mapping rules and identify rules.

Mapping rules define a relation of congruency between an entity a regarding an entity b, as the logic relation that guarantees that a can be obtained from b and a will be consistent regarding its model of representation and b. The mapping rules provide the sufficient information to allow the transformation engine to create the data instances in the output that are equivalent to the ones found in the input. The approach defines three types of basic mapping rules:

- Entity rules: linking entities of the input schema and entities of the output schema, creating instances of entities in the output dataset.

- Property rules: linking an attribute associated with an entity of the input schema with an attribute associated with an entity of the output one. This rule permits to assign values to attributes in the output dataset.

- Relation rules: link a relation associated with two entities of the input schema with a relation associated with two entities of the output one.

The transformation rules are, for some situations, not enough. On the one hand, it may happen that the transformation of data instances requires

to include information that is not made explicit in the input data in order to obtain an accurate semantic representation, this is related to the limitations in terms of semantics of formalisms like XML or relational databases. An example might be to add additional properties to all the instances of a given entity in the input schema in order to enrich their semantic definition. On the other hand, it may happen that a mapping rule requires creating instances under specific conditions. For example, relate instance a with instance b only when b is related with c.

To resolve this problem, the proposal adopts the approaches from software and ontology engineering, which use patterns in order to encapsulate best practices of modelling, facilitating and promoting the construction of reusable software and ontology re-using modules. The patterns used by the transformation model define a template over the output model about different modeling situations. Patterns represent part of the whole definition of a new entity in the output model and are used in the definition of mapping rules in replacement of the output model. Therefore, patterns create a view over the output model that prevents user from the complexity of the model and aids at the definition of complex rules. Patterns are designed using entities, attributes, relations and associations from the output model together with variables linked to specific entities, relations or attributes. The execution of a pattern creates new instances in the output model and it is the instantiation of a variable with different values what creates different instances using the same pattern.

Identity rules define the set of attributes and relations that permit to identify unequivocally each individual in the output dataset. These rules permit to prevent the creation of redundant content in the repository and also support the generation of a repository from multiple data sources, since such identity conditions would permit to identify which entities from different datasets correspond to the same entity in the output schema.

The transformation model defined is generic and can be applied for transforming data between two different models of representation. However, the goal is getting a semantic representation of data represented in traditional formats, so I will focus in the transformation of data into a semantic representation guided by and OWL ontology output model. Figure 12.1 shows the architecture of the designed transformation model.

The use of an OWL architecture as output models allows for new orientations of the transformation model. In an OWL representation of the knowledge, there is no clear boundary between schema and instance. The knowledge can be represented by means of classes or instances, and the nature of a concept as class or instance is a role acquired and depends on the

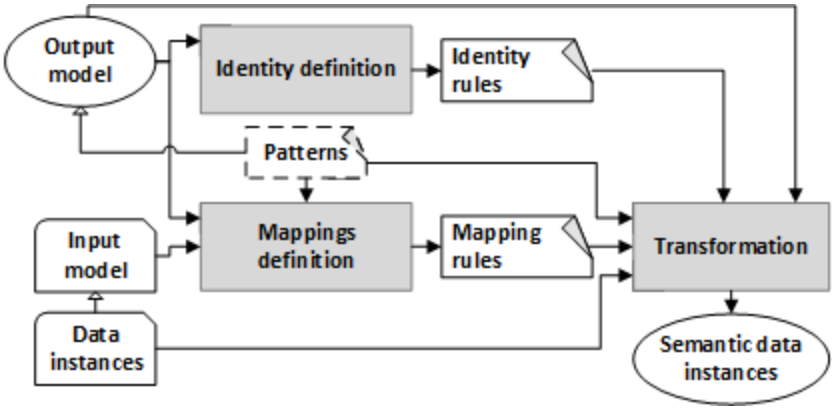

Figure 12.1: Architecture of the transformation model

expected exploitation of the knowledge [230]. Therefore, the transformation model can be adapted by redefinition of rules for creating a class-based representation of the domain instead of a instance-based one. In this way, an output model is reused for creating a new one that allows the representation of input resources, part of the transformation creates new classes in the output model while the rest of the transformation creates new instances.

Another approach of interest is obtaining a new output model from an input model. This approach has special interest in the clinical domain, for the creation of EHR clinical models from existent ones. This makes easier reusing clinical models from an EHR standard, making them available for other organizations that use different standards or specifications. The application of the transformation model to this approach will take advantage of the availability of OWL representations for EHR standards based on the dual model architecture.

12.4.2 Integration based of domain-guided transformation

The integration model allows for the homogeneous combination of heterogeneous resources, overcoming the problems associated with the difference in storage, representation structure, nomenclature and detail level. The integration model is defined by the tuple $\langle \langle s_1, s_2, \ldots, s_N \rangle , \langle m_1, m_2, \ldots, m_N \rangle , ri, O \rangle$, where $\langle s_1, s_2, \ldots, s_N \rangle$ is the set of input resources to be integrated, $\langle m_1, m_2, \ldots, m_N \rangle$ is the set of mapping rules defined for each input resource; ri stands for the identity rules, defined over the output model O.

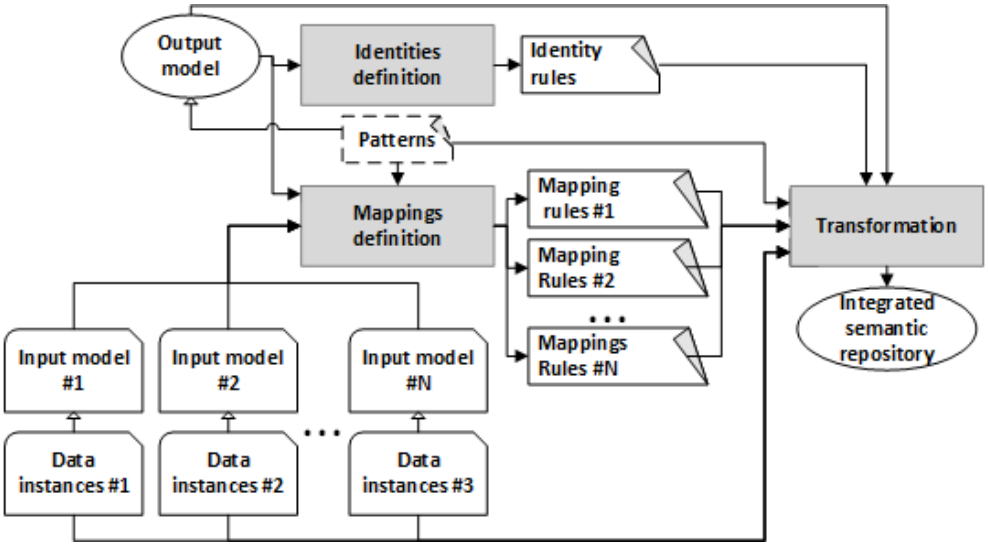

Figure 12.2: Architecture of integration

This integration model allows for combining heterogeneous information independently of the structure of representation in origin, by using the final output model designed, which contains the semantics of the application domain. Figure 12.2 shows the architecture of the integration model. The output model is defined independently of the input resources and contains all the semantics of the domain that guides the integration process. The construction of the output model does not require the alignment between source schemata and it is not the result of the integration of all of them in one. Depending on the final use of the source data, the global ontology can represent all the semantics from the source repositories, for example, when the objective is the complete publication of several repositories into the Web of Data; or can represent part of it, making a partial transformation of the source data that is of interest for a specific study.

The output model is an OWL ontology that can be complemented with ontology design content patterns. The use of patterns allows for the modularization of the output model and makes reusing mappings rules easier, since it is easier to found local matches between the representation of different schemata, rather than global ones.

The transformation of data is done sequentially, for each input resource, the mapping rules are applied in order to extract the source data and transform them into instances of the output model. The identity rules are checked before the integration process to avoid the redundancy of data.

The integration of heterogeneous resources have to overcome some prob-

lems due to the differences in content representation between resources:

- Nomenclature conflicts: this problem is due to the use of different terminologies in the input models for naming their entities. Since the global output model provides an homogeneous vocabulary, the definition of mappings between the input models and the output model resolve this problem.

- Data redundancy: this problem occurs when two or more instances in the input repositories describe the same concept in the output domain, and therefore, the should create the same instance in the resulted repository. This problem is resolved with the definition of identity rules that identify the value that make unique an instance in the output model.

- Inconsistencies due to incomplete data: this problem occurs when the source content does not contain all the information necessary to create an instance in the output global model with all its properties and relations. If the missing property is not a necessary one, the absence of information does not provoke an inconsistency. In those cases, the transformation and integration process run normally with the defined mappings. If the absence of information prevents from checking the identity rules or makes the final result inconsistent, the instance is excluded.

- Inconsistencies of the input models with the output model: this problem is due to the definition of the output model independently of the input models. In some cases, the differences between the models may difficult the definition of mappings rules. The use of patterns is useful in these cases, since the pattern creates a view over the output global model, this view can be adjusted to the specific characteristics of an input model.

- Inconsistencies between source repositories: this problem occurs when the same instance in the global output model is created from several instances of the input repositories, but those instances have different values on their properties. If the inconsistent properties are not those used in the identity rules, the equivalent instances are detected by the identity rules and the problem is handled as a redundancy problem. If the inconsistent properties are those used in the identity rules, these rules identify each instance as a different one, and both are created in the final repository.

12.4.3 Management of biomedical information

The transformation and integration model proposed are guided by the semantics of the domain. However, in the clinical domain, the management of clinical models based on dual model architecture also requires to pay attention on the structure of the models. Clinical models are defined, by constraining a reference model, as independent artifacts that can be shared between different institutions that work with EHRs. Not only the clinical information recorded by them is interested, but their structure that can be validated, compared and reused by different institutions. Hence, a repository of clinical models in a semantic representation resulting from a structure-based transformation processes is useful for performing semantic activities over the models. For example, the validation of their consistency, the comparison between them for finding similarities, and the annotation with biomedical terminologies.

Therefore, this thesis presents solutions for the management of biomedical information using OWL as common framework for the exploitation of clinical models, ontologies and terminologies, where a transformation of clinical models based on their structure is used for building a repository whereas the exploitation of the clinical information collected with those clinical models uses a domain-guided transformation of the data.

Two different OWL representations for archetypes for both ISO 13606 and openEHR are used. Both representations have in common that the information model is represented in OWL, but differ in the type of OWL entity used for representing archetypes:

- Archetypes are represented as OWL individuals [87] of the corresponding information model ontology (i.e., ISO 13606, openEHR). This representation is used for the transformation of archetypes between openEHR and ISO 13606 and for adding archetypes annotations (based on external ontologies/terminologies) which are exploited in activities like archetype comparison and search.

- Archetypes are represented as OWL classes [28] for tasks that require performing automated reasoning over their content, like validating the correctness of specialised archetypes.

Clinical data gathered using archetypes are called extracts and normally represented in XML. Due to the limitations of XML for semantic processing of data, the clinical extracts are transformed into a semantic representation based on OWL using the transformation process introduced above. The application of the transformation model is done by defining mappings between

the archetypes used for gathering the data and an OWL ontology that represents the semantics of the output domain.

12.4.3.1 Semantic annotation

An important activity for the clinical information is the semantic annotation of archetypes and clinical extracts. The annotations are provided by terminologies, ontologies and external semantic resources in the biomedical domain. Archetypes are already linked to terminologies by terminological bindings. However, these terminological bindings are not always defined and, depending on the specific use of the archetype, additional semantic content may be needed, for example to perform a personalized classification of patients. The annotation method is both manual and semi-automatic. Given a repository of ontologies, controlled vocabularies and terminologies in OWL format, the annotation method recommends annotations based on the textual content of the archetype and permits to retrieve exact or partial matches between the content of the archetype and the terms of the entities included in the repository. Besides, the user is also provided with a search facility, which would retrieve the corresponding terms from that repository. For this purpose, the representation of archetypes as OWL individuals is used. The annotations are represented in OWL format, so they can be exploited jointly with the content of the archetype. EHR data are indirectly annotated, mainly through the annotations of the archetypes used to capture the data.

12.4.3.2 Semantic profiles

The group of all the annotations are a representative generalization of the knowledge contained in the archetype and creates its semantic profile, useful for efficient, effective processing without needing to use the whole information about the archetype, using only such semantic interpretation. Using ontologies for such purpose permits to make decisions and recommendations based on a formal specification of the knowledge domain.

In terms of representation, a semantic profile is represented as a set of semantic annotations. The semantic profile of an archetype is the union set of all the annotations and terminological bindings of the archetype.

The semantic profile of an EHR extract is obtained from two sources, namely, archetype and data. First, the semantic profile of the EHR data is derived from the archetype to which the archetype-based data conform (e.g. the semantic profile of an extract about blood pressure will be the semantic profile of the blood pressure archetype). Second, the EHR data permits to define a more precise profile. For example, in case of having a low value

for the blood pressure, the semantic profile could include the annotation "hypotension".

12.4.3.3 Semantic similarity

The comparison between archetypes is important in order to create a non-redundant repository and encourage reusing available clinical models. In order to be able to identify archetypes expressing the same meaning, a semantic similarity method is included. The input to the similarity method is the two archetypes to be compared from the archetype repository, and the output is a score in the range [0,1]. Retrieving the archetypes from the repository enables to access the semantic profile of the archetype, which includes the semantic annotations and the terminology bindings. Besides, the ontologies of the information model, archetype model and the ones used in the semantic annotations of the semantic profile of the archetypes provide the semantic context for comparing the archetypes. The similarity approach is node-based since it exploits the terminological bindings, the semantic annotations and the hierarchical structure of ontologies. Basically, the method compares all the pairs of elements in the semantic profiles of the archetypes, obtaining a similarity score for each pair. This pairwise analysis returns the set of pairs obtained by the following steps:

- Compare all the pairs and select those with score higher than a given threshold.
- Get the set of pairs that maximize the sum of the similarity scores that include only one pair per element of the semantic profile of each archetype.

The pairwise similarity function uses the following factors:

- Taxonomic similarity (d): This distance measures the hierarchical distance between the classes associated with the two elements C_i and C_j , that is, through taxonomic links. This function uses both the union set of ancestors and the set of common ancestors of the classes. It should be noted that classes might present multiple inheritance, which would imply different taxonomic paths and, therefore, different taxonomic similarity scores. In such cases, the shortest distance is returned by the function.
- Properties similarity (ps): Similarity between the set of properties associated with the classes associated with the two elements.

- Linguistic similarity (ls): A string-based calculation of the terms associated with the ontological elements compared. If we are comparing two concepts from the OWL representation of two archetypes, this calculation uses the term definition of both concepts. When comparing two concepts from a terminology, it uses labels or the local name of the concepts compared. The current implementation uses the Levenshtein distance [233].

The sum of all the pairwise similarities between the selected pairs of elements of the semantic profiles returned by the pairwise analysis method constitutes the similarity of the semantic profiles of the archetypes. In addition to this, the similarity method includes another factor that takes into account the structural types of the archetypes compared in the context of the information model ontology. This factor, written structural similarity, assumes that two archetypes of the same type COMPOSITION are more similar than two archetypes of different types, for instance, COMPOSITION and SECTION. This score is obtained by applying the taxonomic similarity function to the types of both archetypes.

12.4.3.4 Secondary use of biomedical information

The availability of clinical data of patients in EHR systems promotes the secondary use of the data.

A major motivation for using OWL is its capability to perform sound and complete automated reasoning. OWL-DL classes have sets of axioms associated and two types of axioms are relevant for reasoning: (1) *subClassOf*; and (2) *equivalentClass*. The former one permits to define necessary conditions for an OWL individual to be a member of the OWL class, whereas the latter one permit to define sufficient conditions for an OWL individual to be classified as a member of the OWL class. Defining the inclusion/exclusion criteria as *equivalentClass* axioms permits the reasoner to automatically partition the clinical data into the groups of clinical interest.

This characteristic gives the possibility of applying OWL reasoning to classify patients according to the available EHR data. Patient classification means grouping patients in different categories attending to certain clinical criteria. Whereas the domain knowledge is usually modelled using *subClassOf* axioms, the classification rules are specified using *equivalentClass* ones. Ideally, the classification rules are implemented in a separated ontology that reuses the domain ontologies used for representing the clinical data. The ontology with the classification rules is called a classification ontology and it contains, at least, one class per group of interest.

Once this ontology is ready, an OWL-DL reasoner like Hermit [108] can be applied over the complete semantic dataset to infer all the possible information given the data. The result of such inference process will be the resulting classifications, which can be retrieved using semantic query languages like DL-query [234] or SPARQL [109], or through a programmatic API like OWLAPI [235].

In this proposal, two prerequisites have to be fulfilled, the classification categories can be specified in terms of rules expressed as OWL DL defined classes, and the EHR data is available in OWL. The classifications obtained for a given patient enrich its semantic profile associated, since they can be represented as new annotations associated with a given EHR extract.

The clinical data contained in the EHRs can also be used as source of learning resources recommendation methods. The recommendation of learning resources has the objective of providing training guided by clinical data from EHRs to patients and clinical experts. The recommendation proposed in this thesis requires is performance within an integrated architecture where it has access to semantic profiles and clinical data of the patients, clinical terminologies and distinction between clinicians and patients, together with a repository of learning resources enriched with semantic annotations, that constitute the semantic profile of the resource.

The recommendation of resources for a patient is done at extract level. Given the clinical extract of a patient, the method recovers its semantic profile associated and compares it with the semantic profile of the learning resources available. The comparison is done by applying the semantic similarity method explained above. The user sets a similarity threshold to accept or reject learning resources whose similarity score is above or below a certain value.

Since many of the resources are aimed to specialist users, the recommendation methods take into account another variable, namely the expert score. Specialist users, normally clinicians, assess the appropriateness of a learning resource to non-expert users. Depending on this score, the probabilities of a resource to be selected for a user increases (with a high expert score), or decreases (with a low expert score).

Lastly, the availability of integrated clinical data makes the assessment of the quality of care easier. This assessment is done using quality indicators, released by governmental institutions or other associations related to healthcare. Normally, the quality indicators are released in natural language and computed as an equation. For example, *"Number of examined lymph nodes after resection of a primary colon carcinoma"* is an indicator which numerator is *"Number of patients who had 10 or more lymph nodes examined*

after resection of a primary colon carcinoma", is the desired procedure to be followed, and has as denominator *"Number of patients who had lymph nodes examined after resection of a primary colon carcinoma"* all the procedures performed. The problem with those quality indicators is that they are normally computed manually, leading to problems of ambiguity and inefficiency. The CLIF method [236], developed by the medical informatics group from Academic Medical Center (AMC), formalizes quality indicators so they can be unambiguously defined and computed automatically over clinical data collected during the clinical care process. The formalization of the method has 8-steps that results in a query over the clinical data equivalent to the quality indicator. This 8-steps generic method has been instantiated to be used over a OWL-based representation of the clinical data, so that the result of the method is a SPARQL query that can be executed over an OWL representation of clinical data for computing a specific quality indicator.

12.5 Validation scenarios

Two main tools have been built in the development of this thesis. The transformation model has been implemented in the Semantic Web Integration Tool (SWIT) [231], that provides a web interface for guiding users through the steps of the transformation model:

1. The users selects input data and schema. Currently SWIT accepts data stored in a relational database, XML data following a XML schema and XML data following ADL archetypes. In this step, the user selects the output ontology and optionally, design patterns.

2. The user defines the mappings between the input schema and the output ontology/pattern. The interface assists in the process. Figure 12.3 shows SWIT interface for mapping definitions.

3. The user defines the identity rules over the output ontology.

4. The transformation process is executed.

The platform Archetype Management System [237] integrates the methods for managing and exploiting archetypes, clinical data and external resources. Figure 12.4 shows the general architecture of ArchMS.

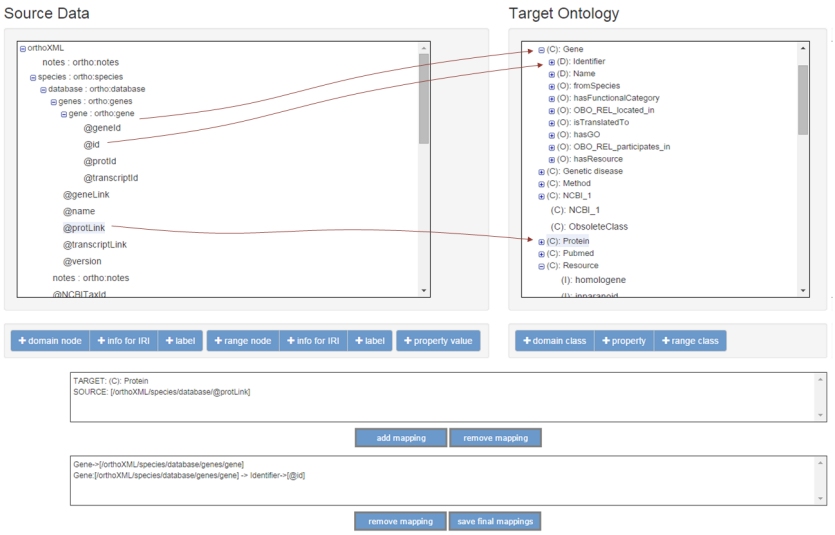

Figure 12.3: Mapping definition in SWIT

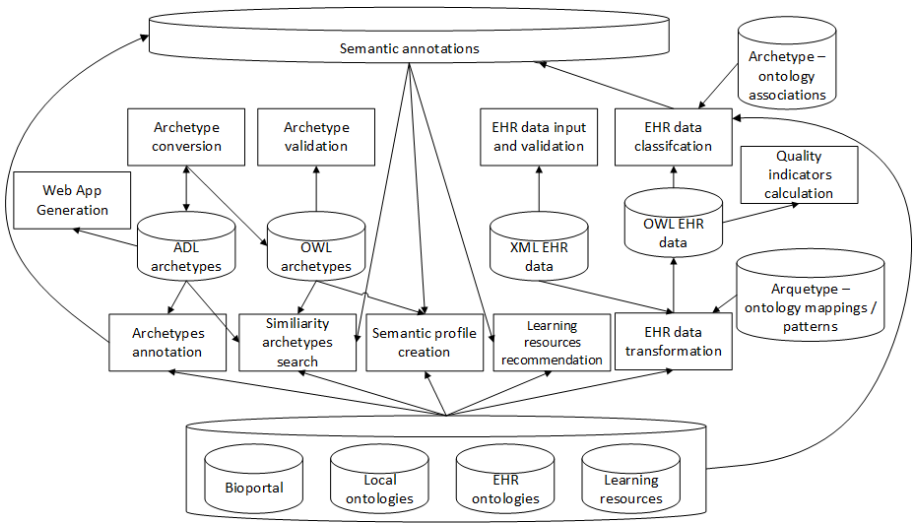

Figure 12.4: Architecture of ArchMS

The main activities that can be performed with archetypes are: conversion, validation, annotation, search for similar archetypes or for archetypes with concrete properties, and the generation of applications. Figure 12.5 shows ArchMS interface.

- Archetype management: The system allows importing ADL archetypes

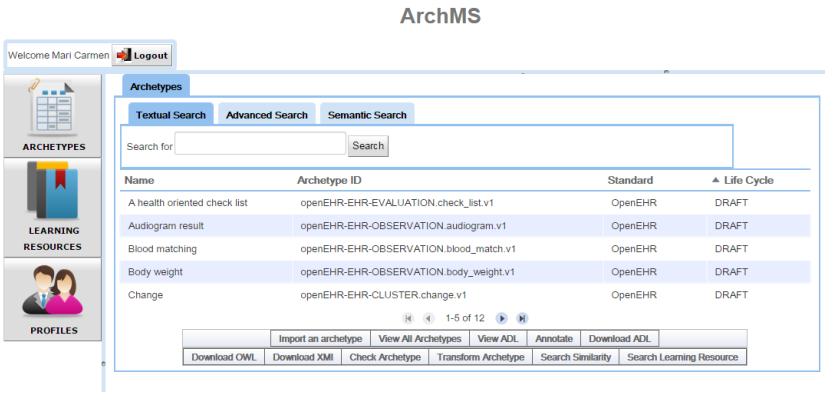

Figure 12.5: ArchMS interface

for both openEHR and ISO 13606 representations. It includes the functionality provided by previous developed tools by the group within this thesis has been developed, Archeck [175] and PoseacleConverter [173]. The first one allows checking the consistency of a specialized archetype regarding its parents. The PoseacleConverter allows transforming archetypes from openEHR into ISO 13606 and viceversa and representing them in OWL. ArchMS uses a MySQL relational database (ADL Archetypes) to store archetype metadata properties such as name, language, or purpose, as well as the ADL file and a semantic repository implemented using Jena [238] to store the OWL representation of the archetypes, allowing for issuing SPARQL queries. In order to speed-up queries, an archetype Lucene [239] index is obtained by parsing the ontology and keywords sections of the archetypes.

ArchMS integrates the generator of web applications based on archetypes created within the group [240] as a service, so the ArchMS administrator can generate ArchForms applications and make them available to the rest of users for downloading and further deployment.

- Archetype annotation: ArchMS implements manual and semi-automatic methods for the annotation of the archetypes. Bioportal is the main source of annotation resources for users.

- Archetype search and similarity: ArchMS provides different query options that exploit both the relational and semantic repositories:

    1. The textual search interface uses the Lucene index for finding archetypes that contain some textual description which matches

the textual description of the query.

2. The advanced search exploits the relational database for finding archetypes by metadata (e.g. language, archetype name, etc.) and by archetype annotations, in case they are available.
3. The semantic search executes a SPARQL query against the semantic repository of archetypes. This search facility exploits the representation of archetypes as OWL individuals.

ArchMS implements the semantic similarity function described in the Section Semantic Similarity, which permits to find which archetypes are similar. On the one hand, a user can decide to search for similar archetypes. On the other hand, once an archetype is successfully imported, ArchMS looks for similar archetypes using the default similarity threshold for checking whether an equivalent archetype already exists in the repository and recommending annotations associated with similar archetypes.

The main activities that can be performed with EHR data are to obtain the OWL representation of XML extracts, to visualize and input EHR data, to obtain the semantic profiles of EHR extracts and to classify EHR data.

- Data management: ArchMS processes XML EHR extracts from both ISO 13606 and openEHR specifications. The extracts imported into ArchMS can be generated using ArchForms or any other ISO 13606 or openEHR data management system.

- Data transformation and profiling: ArchMS enables to perform semantic activities on EHR data by transforming EHR extracts into OWL. This function uses the mappings between the archetypes and the ontologies, which have been previously uploaded to ArchMS. Such mappings can be created by invoking a SWIT service from ArchMS to automatically execute the data transformation once the extract has been imported into ArchMS. Once transformed into OWL, the semantic profile of the EHR data is automatically extracted.

- Data classification: ArchMS permits to classify EHR data according to rules that define the clinical status of the patients. For this purpose, the user has to select the classification ontology.

- Recommendation of learning resources: ArchMS makes a clinical data-based recommendation of learning resources. It gets the recommenda-

tion from PubMed, where resources are annotated using MesH (this resource is mainly for clinical experts) and from a repository of resources represented in the Shareable Content Object Model format (SCORM), which are annotated in the same way as archetypes and clinical extracts.

ArchMS manages three types of users in ArchMS, namely, administrator, user and physician. The administrator is in charge of the maintenance of the system and will be responsible for particular tasks like assigning the physician role to the corresponding users or generating the ArchForms applications. The administrator is also in charge of defining the semantic profile of the archetypes and SCORM learning resources, annotating them either using directly biomedical terminologies or recommendations from similar archetypes, and uploading the mappings between archetypes and ontologies. Patients have access to their clinical data, learning resources and additional knowledge based o their profiles, depending on classification resources. Physicians can access to the medical histories of all their patients, get additional knowledge from the clinical data of their patients and use advanced process for transforming the clinical data and creating repositories suitable to their research.

These tools have been used in the application of the following use cases:

- Orthology data and genomic sequence annotation: SWIT has been used for the creation of an integrating resource about genetic human disease, orthologous genes and information deriving from genome annotation processes. The OGOLOD repository [165] integrates information from orthology databases such as Inparanoid [63],KOG [166], Homologene [167] and OrthoMCL [168], with the OMIM database [169]. The content of the latest version of OGOLOD was generated using SWIT with the original set of selected orthologs databases but represented not only in relational schemata but also using the OrthoXML file format and integrated with information derived from sequencing projects. Consequently, the OGO ontology was updated and we used SWIT to define and execute the corresponding mapping rules between OrthoXML format and the OGO ontology and the sequence annotating projects relational database.

- Chemical components: SWIT has been used for generating semantic datasets of chemical components libraries. Virtual screening methods use libraries of small molecules to find the most promising structures that could bind with drug targets. Once of such libraries is ZINC [248], a free database of commercially-available compounds for virtual

screening. ZINC data can be downloaded in XML format, so the semantic dataset for ZINC was generated by creating a XML Schema and defining the mappings with a domain-specific ontology developed.

- EHR data: EHR data of colorectal cancer patients is stored in ArchMS together with the archetypes used for gathering the data. The transformation model implemented by SWIT was executed for getting a semantic representation of the data based on a domain ontology and, together with a classification ontology, performing automatic reasoning over the semantic content to determine the level of risk of developing cancer of each patient.

12.6 Discussion and future work

The query and exploitation of biomedical data requires methods that make easier their integrated access. The Semantic Web has been proposed as a technological space in which biomedical data can be integrated and exploited [17]. Besides, the Semantic Web community wishes to achieve the Web of Data, which would semantically connect datasets distributed over the Internet. The Linked Open Data effort [20] pursues the publication and sharing of biomedical datasets using semantic formats. Berners-Lee [110] suggested a five-star deployment scheme for Open Data, where each level impose additional conditions. The use of RDF and an appropriate use of URIs permit the achievement of the fourth star and the fifth one can be achieved by getting the dataset linked from an external one. The construction of these datasets is hampered by the heterogeneity of the available biomedical data, so the development of new methods that assist in the creation process is required.

The first step in bringing the biomedical datasets into the Semantic Web is the definition of methods of transformation. In this thesis, a transformation model is designed based on the definition of mapping rules between the input model that define the structure of the source content and the output model that defines the final representation. Together with the mapping rules, the model uses identity rules for identifying redundant transformed data.

The instantiation of the transformation model to an output model based on an OWL ontology gives the possibility of applying the transformation model to the transformation of data into OWL classes instead of OWL instances. The choice of one or another depends on the use case and the intended exploitation of the data, so the flexibility of the transformation model is key for getting the final intended repository.

The mapping definition in the transformation model may be a complex

task. The transformation model allows the definition of design patterns that make the mapping definition easier, while the tool SWIT, that implements the transformation model for creating OWL/RDF repositories, provides with a friendly interface that assists in the mapping definition. However, an important question is who should define the mappings. The lesson learned here is that training of health informatics data managers on semantic technologies has to be increased to exploit the best of the open semantic technologies. Besides, even ontology expert may have problems to define the mappings with complex schemata and ontologies. Automatic mapping techniques would have contributed to significantly reduce the mapping time, so efforts in this area are key to support the mapping process.

Identity rules control the redundancy of the new instances and are a key player in the integration process. The definition of the rules requires the identification of attributes and relations of an entity that describe how instances can be distinguished for one another. Therefore, the correct definition of rules requires the existence of properties that give the entity their identity quality in the same way as is defined in formal ontologies [249]. These identity properties are not related with the properties that define class membership, so identity rules are not defined using necessary criteria, but with unique properties that allows the differentiation of instances from each other. In an OWL ontology, the properties use by an identity rule are those that would be use in a Key axiom [250]. These axioms link a class with a set of `owl:ObjectProperty` and `owl:DatatypeProperty`, making it uniquely identified by those properties, that is, no two distinct instances of the class can coincide on the values of all the properties declared in the set. Key axioms apply only to individuals explicitly introduced in the ontology by name, and not to unnamed individuals. Thus, key axioms will not affect class-based inferences. These limitations makes necessary to use other methods for identifying equivalent instances.

The SWIT full mode checks all the formal aspects that guarantee the generation of consistent datasets, independently of the use case and the intended exploitation of the data. The consequence is that, for medium and large datasets, the transformation time is longer than expected because of the number of instances of axioms to be generated. This might not be a problem in case of stable datasets or batched transformation process. However, the intended exploitation of the dataset might permit to relax some conditions of the transformation process. In case of transforming independent datasets or in those cases where the identity is guaranteed by the construction of the URI, identity conditions could be skipped. In case of not requiring automated reasoning on the transformed dataset, the generation of other types

of axioms might be omitted, saving time and space. For example, it may not be needed to add the `owl:differentFrom` axiom to all instances with each other, and therefore the execution time in the algorithm is reduced, according to the study of complexity made on chapter 7 (section 7.2.2), in $O(i_t)^2$, being i_t the highest number of new instances created.

The instantiation of the transformation model into a output model defined by an ontological architecture, consisting on an OWL ontology and ontological design content patterns, allows the creation of an integration model guided by the transformation of heterogeneous resources into a common model, that is, guided by the output domain, and independent of the structure of the input resources.

The integration model is not oriented to the transformation and integration of complete repositories, but to the integration guided by an application domain, that selects only the input data of interest for the output model. However, the integration is done in a physical common repository, and the generated repository may be large and cause problems of efficiency, for example, in the inference processes. The limitation of size depends on the system that hosts the final repository.

Finally, the availability of OWL representations for EHR clinical models, together with management semantic methods and the transformation and integration models designed, allows the creation of an integrated platform where clinical models can be managed together with clinical data and their semantics can be exploited together with external resources. This platform is implemented in the tool ArchMS.

The use of a Semantic Web infrastructure is likely to be the major novelty of ArchMS over state of the art systems like the openEHR CKM, LinkEHR or CIMI which are based on ADL technology and are oriented to support the construction and publication of existing archetypes. The specifications of the archetype model have not been designed having in mind the Internet or the Semantic Web. An example is the fact that archetypes do not have URIs, which are the identifiers of resources in the Semantic Web. In ArchMS, a URI is generated for each archetype when represented and exploited in RDF/OWL. Also, ArchMS stores both archetypes and data extracts. However, the key advantage of ArchMS against other systems is the use of OWL technologies, which allow for the combination of information model, clinical models and terminologies.

ArchMS makes use of ontologies in different ways: controlled vocabulary, knowledge schema, consistent search, classifying instances, reuse and inferencing, all these uses being among the major applications of ontologies according to [51]. One major use of ontologies in biomedical domains is

annotation, the Gene Ontology [21] being the most important one. In this use, ontologies are exploited as controlled vocabularies, since the ontology classes are mainly the annotation entities. Archetype terminology bindings should not be confused with the annotations provided by ArchMS. These are usually added to the archetype data elements or terms during its building, and ArchMS does not intend to support the design and development of archetypes. The annotations provided by the system should be understood as archetype metadata since they are associated with the archetype as a whole and not with their individual terms. ArchMS is able to suggest archetype annotations in two different ways: (1) textual search; (2) archetype similarity.

In the one hand the textual description of the archetype is processed and issued against biomedical terminologies to get recommended annotations. Despite this approach has been helpful for different research projects, there are some specific archetype annotation automatic methods whose integration into ArchMS should be studied [251; 252].

In the other hand, annotations are based on the semantic similarity of archetypes calculated by applying state of the art semantic similarity functions. Such measurement is an example of the use of ontologies as domain schema since the classes and properties from the information and archetype model ontologies (the ontologies used by the PoseacleConverter) are used for the calculation. However, this measurement does not require using automated reasoning. The semantic similarity functions can be customised by the users by specifying the values to the threshold and the weights. There is no standard or automatic way to determine the best values for the weights, so an analysis has been carried out in order to suggest their potentially best range values. Generally speaking, a higher value of a weight means that we are providing more importance to that factor among the others. The weight for the linguistic similarity should be the smallest one because it does not really provide information about the particular structure or meaning of the knowledge entity. Provided that the similarity method compares classes in ontologies, the taxonomic distance should be considered the most important. However, the application of this method in use cases, such as the recommendation of learning resources, have shown the interest on defining different parameters to the annotations depending on their origin, so local decisions should be made due to the local nature of their meaning. This mechanism based on weights and thresholds permits each group of ArchMS users to obtain results adjusted to their notion of similarity. However, the selection of proper parameters requires a deep knowledge on Semantic Web technologies. Additional research should be made to learn optimal sets of parameters depending on the properties of the archetypes compared and the size of the

ontologies used in the annotations of the archetypes.

Reasoning with ontologies is also exploited in ArchMS with both archetypes and data. The archetype validation service included in ArchMS, namely Archeck, checks the correctness of archetypes including specialization by applying automatic reasoning over the OWL representation of archetypes. It should be reminded that Archeck represents archetypes using OWL classes and the PoseacleConverter represents them as OWL individuals given the different purpose of the tasks. The original purpose of the PoseacleConverter OWL archetypes representation was to support their transformation between standards, for which it demonstrated to be effective. It should be noted that none of the presented OWL archetypes representations are proposed as standard ones, but they constitute appropriate technological decisions for the different semantic activities performed in the system.

On the data side, automated reasoning is used for the classification of patients. Such activity is performed over the patient data imported into ArchMS as XML extracts and transformed into RDF/OWL using the SWIT methods. As it has been mentioned, such transformation is driven by domain ontologies, which play the role of knowledge schema in such transformation and enhanced with the use of semantic patterns. Once the EHR extracts are transformed into instances, inference is used for classifying such instances. For instance, in the colorectal cancer screening effort, the patient data were classified by level of risk according to the European and American protocols. This data transformation permits to move from the archetype technological space to the Semantic Web one. Currently, our transformed data do not keep information about the structure of the archetypes, since the transformation is purely driven by the domain ontology. In the future, could be interested to also transform the structure of the archetypes to investigate which transformation approach can be more appropriate for different tasks.

12.7 Hypothesis verification

The main hypothesis of this thesis is that by using Semantic Web technologies, two main advantages can be achieved: the creation of a generic method for integrating heterogeneous biomedical information, and the reduction of the effort on managing biomedical information, including clinical models and data. This hypothesis is divided in sub-hypothesis that have been proved by answering the following questions:

12.7.1 Sub-hypothesis 1

A transformation model for biomedical information guided by the domain of the output representation can be designed by the definition of transformation rules and design patterns.

What formats of representation are more common in biomedical information systems? In chapter 2, the most common representation formats of biomedical information are introduced. The chapter is divided into clinical and biological information.

In clinical information systems, data is stored in the health record of the patient. For the electronic version of the health record (Electronic Health Record (EHR)), there exits numerous standards and specifications defining its architecture. Among them, stand out those based on the dual model architecture, which distinguishes two modelling levels. On the one hand, the information model provides the generic building blocks to structure the EHR information. On the other hand, clinical models are used to specify clinical recording scenarios by constraining the information model structures. The chapter presents the standards and specifications based on dual model CEN/ISO 13606, openEHR, HL7 and CEM. In all the explained standards and specifications, clinical models are defined using languages with a syntactic orientation, such as ADL or CDL, while the clinical extracts are usually stored using XML files.

Biological data, including data about nucleotides, proteins, genomes, protein structures, specific organisms, etc., are commonly represented in plain text files; structured or semi-structured files, such as XML; relational databases and graph-based databases.

Both clinical and biological repositories use biomedical terminologies for the annotation of the entities, which gives standardization to the terms and makes easier the re-use of the data.

The variety of proposals for representing data and terminologies available makes difficult the access, management, interpretation and semantic interoperability of biomedical information. Languages such as ADL, used in archetype representation, or XML, used on biological resources representation, have a syntactic orientation and are not flexible enough for tasks that require exploiting the data and their associated semantics.

What methods of content transformation into a semantic representation are available and what are their associated problems? Different initiatives and studies propose the use of Semantic Web technologies for biomedical information representation, management and integration. On the one hand, ontologies allow the creation of complete knowledge mod-

els. On the other hand, the Web of Data, resulted from the application of the principles of Linked Data, allows publishing and sharing biomedical datasets.

Chapter 4 presents different tools used for getting a semantic representation of information resources based on OWL ontologies or RDF. The transformation methodologies follow by the tools have a common schema. Starting from a data repository following an input model, the definition of mappings between that model and the output model guides the process of data extraction and transformation. In some cases, the output model already exists, while in other cases, it is generated from the input model.

Tools such a as D2RQ, Triplify, Virtuoso Linked Data View or XS2-OWL generated and OWL or RDF representation of data stored in relational databases or XML files. The generation of the output repository is automatic or semi-automatic and it is guided by the logical schema of the source representation. Other tools, such a RDB2OWL or Karma make a domain-guided transformation, using pre-existing ontologies for the final representation of the data, but have the problem of being complex and oriented to a specific representation format.

What components define a generic transformation model? Chapter 7 presents the first proposal of this thesis, a generic transformation model of information resources. The transformation model consists on: an input model that defines the representation of instances in the input resources; an output model, that defines the representation of the instances resulted from the transformation; and transformation rules, that guide the transformation process and are divided into mapping and identity rules.

The input and output models are structured following a metamodel that defines the minimal components that a model should contain for being used in the transformation model. This model aims to accept input resources with structured information, such as relational database, XML files o clinical extracts based on archetypes.

The mapping rules define the association between the input model and the output model and this association defines the representation of instances of the input model as instances of the output model. For defining transformations of higher complexity, the transformation model incorporates design patterns that represent the definition of an entity in the output model, creating a view that prevents the user from the underlying complexity of the output model.

Identity rules check that the transformed instances are unique. They define the properties and relations that differentiates an instance in the output model for the others, so the transformation model uses these rules for identifying redundant instances.

Since the rules are defined using the components of the input and output model, any schema following the input and output metamodel defined by the transformation model can be used. If the model is applied to an output model defined by an OWL ontology, the transformation is guided by the semantic of the output model

The final result is a generic transformation model for any input and output schemata that follow the characteristics of the defined metamodel, that allows a flexible definition of mappings rules guiding the transformation and incorporates mechanisms for making complex transformation easier and controlling the problems of redundancy.

What advantages brings the use of transformation rules and design patterns in the transformation model? On the one hand, transformation rules provides flexibility to the transformation process, since different transformation can be obtained from the same source depending on the output domain. The patterns, implemented using OPPL 2 language, make easier the transformation of complex entities. A pattern contains all properties and associations that characterize an entity in the output model, while parametrizes those values that depends on the input model, so there is no need to have a deep knowledge about the output model, and the definition of the mapping rules can be guided by the pattern. On the other hand, identity rules, that define what makes an instance unique from one another, are used for identifying instances that are semantically identical and avoid redundancy on the final repository.

12.7.2 Sub-hypothesis 2

The application of the generic transformation model into an OWL representation allows to define a generic integration process for heterogeneous information sources.

What integration methods of heterogeneous resources are available and what are their associated problems? In chapter 5, existent existent integration architectures and systems are presented. The most common integration architectures are based on data warehouses, on mediators or on links. Regarding the use of Semantic Web technologies, ontologies are widely use for modelling global schemas in integration methods, while the Linked Open Data (LOD) principles promote the publication of information resources in semantic format.

The integration systems on biomedical domains presented in chapter 5 are solutions created specifically for the problem they have to resolve and oriented to the source resources. Besides, biomedical resource are very heterogeneous,

so an important aspect of integration systems is the resolution of conflict problems between schemata and data. The presented system relies on manual intervention for resolving those conflicts. Therefore, there exists a lack of generic solutions for resource integration and without manual intervention of users for the resolution of conflicts.

How can integration resources be generalized to be applied into any information resource? Chapter 8 presents the integrated model defined in this thesis, based on the transformation of heterogeneous resources into a global model defined by an ontological architecture. This architecture consists on an OWL ontology together with ontological content design patterns and define the application domain of the integrated data. Since the transformation model is guided by the mappings defined over the output ontology and patterns, the integration obtained is based on a transformation guided by the application domain. The resulted integrated repository is modeled by the ontological architecture independently of the structure of the source resources.

Hence, the integration through transformation results in a generic integrated model, suitable for any input model that follows the requirements of the transformation models, and flexible in the creation of the final integrated repository, that depends on the output model.

How the use of Semantic Web technologies improve the integration of heterogeneous resources? The global ontology used as output model offers a common vocabulary for representing the source content, so the mapping rules linking input schemata with the output ontology allows solving conflicts of nomenclature.

The joint use of transformation rules and design patterns make easier solving inconsistency problems between input and output schema. The use of an ontological architecture, provides the explicit semantic in the instances representation that makes easier the correct definition of identity rules. Therefore, redundant instances can be identified between the sources and the inconsistencies between them can be located.

The ontology provides a common formalisms for representing all the resources in the integration, and being guided by transformation allows getting the semantic representation of the source content, that facilitates their later exploitation. The integration model uses OWL ontologies that allow the use of inference for checking the consistency of the final integrated resource.

12.7.3 Sub-hypothesis 3

The application of the defined integrated process and methods based on semantic web technologies allows the integrated exploitation of the knowledge included on biomedical resources and the secondary use of information.

What tasks are key for managing, exploiting and the secondary use of biomedical information? Research on the different disciplines of biomedicine requires accessing information distributed across different systems and represented in different formats. Therefore, is important to have an integrated and homogeneous access to the information.

At EHR level, the existent clinical models management systems, some of them exposed in chapter 2, are oriented to the creation and publications of new clinical models in a specific standard or specification and make use of a syntactic representation of the clinical models, which makes difficult to perform semantic activities for exploiting them, such as validation of the consistency or semantic comparison, which are key for semantic interoperability and encouraging the sharing of clinical models between institutions.

The complete exploitation of clinical resources allows the integration with external resources for supporting different scientific studies and makes easier their access for activities like rapid cohort identification and quality of care assessment.

How the use of a semantic representation based on OWL ontologies makes easier the task of management and secondary use of biomedical information? Chapter 9 presents the last proposal of this thesis, a platform for managing EHR-related information together with external biomedical resources.

Using different OWL representations of ADL archetypes, the platform provides semantic activities including transformation between ISO 13606 and openEHR standards, the validation of archetype consistency regarding the reference model and the parent archetype, the semantic comparison between archetypes, the annotation of archetypes using external biomedical terminologies and the construction of semantic profiles. Together, these activities provide a management platform for archetypes that improves their exploitation and reuse. The platform accepts extracts of clinical data that can be exploited together with the archetypes used for recording the data.

With the inclusion of the transformation and integration model into the platform, clinical data can be transformed into an OWL representation guided by the domain, so they can be exploited using reasoning techniques provided by OWL and integrated with external resources.

12.8 Contributions

This thesis presents solutions for the transformation of biomedical information into a semantic representation with the final purpose of making easier its semantic interoperability, integration, management and exploitation and supporting research activities for achieving translational medicine. The solutions presented have been successfully used in the following validation scenarios:

- Study of data from clinical extracts based on archetypes from patients of a colorectal cancer screening program, defined by openEHR archetypes. The clinical extract are transformed into an OWL representation defined by a domain ontology that allows applying reasoning techniques and classifying patients according to their risk of developing colorectal cancer.
- Transformation between CEM clinical models and openEHR archetypes, with the purpose of encouraging the sharing of clinical models.
- Integration of heterogeneous resources about orthologous genes, genetic diseases and information about genomic sequences annotation, stored in relational databases and XML files, through their transformation into a representation based on an OWL ontology with the purpose of building an integrated RDF repository.
- Transformation of chemical components on XML files to an OWL representation that makes easier the search of new molecules.

The main contributions of these work are the following:

- Design of a generic data transformation model between structured representation schemata. The definition of mappings between the input model and the output model allows the transformation of input instances into a representation guided by the output model. Identity rules allows the identification of redundant instances. The accepted input and output models are defined by a metamodel and the use of design pattern allows making more complex transformations.

  Through the inclusion of the rules, the transformation model can be adapted to the transformation and creation of new entities in an output model, instead of transforming and creating new instances. The application of this adaptation into a validation scenario for obtaining openEHR archetypes from CEM resulted in a set of openEHR OWL templates and patterns for automatically perform this transformation.

- Design of a heterogeneous biomedical information integration model. Through the instantiation of the transformation model with an output model defining by an ontological architecture consisting on an OWL ontology and ontology design content patterns, different heterogeneous resources are integrated. The output model is defined independently of the structure of the source content. The result is the integration of resources guided by the application domain, since is based on the transformation of the content into a global output model, independent of the input models.

- Design of a platform for integrating, managing and exploiting biomedical information. The platform selects the most suitable OWL representations for clinical models and includes semantic methods for validating, annotating, comparing and searching, allowing a suitable management and share of the models. The platform includes the defined transformation and integration models, allowing the joint exploitation of clinical data, clinical models and external biomedical resources.

- Implementation of the transformation model, the integration models and the integrated platform in two web applications. SWIT implements the transformation model, assisting during the process of transformation. ArchMS implements the integrated platform, manages the clinical models taking into account their semantics, given by their terminological bindings and external annotations, and manages clinical data, allowing its secondary use. With the transformation and integration model, ArchMS allows the creation of semantic repositories of clinical data, that can be exploited and integrated with other resources.

12.9 General conclusions

Translational research requires the integrated access to heterogeneous resources. At the clinical level, different standards and specifications proposed intend the achievement of semantic interoperability, while initiatives like Linked Open Data pursues the publication and sharing of biomedical datasets. However, the syntactic nature of languages used for clinical models representation is not enough for their management, while methods for datasets publication in the Web of Data make a syntactic transformation, guided by the logical schema of the source representation and there exists problems in the generalization of the methods.

The use of global models based on OWL ontologies for representing the source content allows the definition of a transformation guided by the se-

mantic of the domain and uses this semantic for exploiting the final resulted repository. An OWL representation allows the validation and comparison of the content attending to its semantic, making easier the integration of different resources.

The developed tools have demonstrated their effectiveness in different validation scenarios, creating semantic open datasets that will contribute to the development of the Web of Data and allowing the exploitation of them in the Semantic Web technological space.

Bibliografía

[1] Shortliffe EH (2014) Biomedical Informatics: The Science and the Pragmatics. In: Shortliffe EH, Cimino JJ, editors, Biomedical Informatics, Springer London. pp. 3–37.

[2] Kuhn KA, Knoll A, Mewes HW, Schwaiger M, Bode A, et al. (2008) Informatics and medicine–from molecules to populations. Methods of Information in Medicine 47: 283–295.

[3] Woolf SH (2008) The meaning of translational research and why it matters. JAMA 299: 211–213.

[4] Sarkar IN (2010) Biomedical informatics and translational medicine. Journal of Translational Medicine 8: 22.

[5] Tenenbaum JD, Shah NH, Altman RB (2014) Translational Bioinformatics. In: Shortliffe EH, Cimino JJ, editors, Biomedical Informatics, Springer London. pp. 721–754.

[6] Richesson RL, Krischer J (2007) Data Standards in Clinical Research: Gaps, Overlaps, Challenges and Future Directions. Journal of the American Medical Informatics Association : JAMIA 14: 687–696.

[7] Galperin MY, Rigden DJ, Fernández-Suárez XM (2015) The 2015 Nucleic Acids Research Database Issue and Molecular Biology Database Collection. Nucleic Acids Research 43: D1–D5.

[8] Danciu I, Cowan JD, Basford M, Wang X, Saip A, et al. (2014) Secondary use of clinical data: the Vanderbilt approach. Journal of Biomedical Informatics 52: 28–35.

[9] Tapuria A, Kalra D, Kobayashi S (2013) Contribution of Clinical Archetypes, and the Challenges, towards Achieving Semantic Interoperability for EHRs. Healthcare Informatics Research 19: 286–292.

[10] Kalra D, Lewalle P, Rector A, Rodrigues J, Stroetman K, et al. (2009) Semantic interoperability for better health and safer healthcare. Semantic HEALTH Report, European Commission, Luxembourg.

[11] Saleem JJ, Flanagan ME, Wilck NR, Demetriades J, Doebbeling BN (2013) The next-generation electronic health record: perspectives of key leaders from the US Department of Veterans Affairs. Journal of the American Medical Informatics Association: JAMIA 20: e175–177.

[12] EN 13606 Association. The CEN/ISO EN13606 standard. `http://www.en13606.org/`. Último acceso: Mayo 2015.

[13] The openEHR Foundation. Especificación openEHR. `http://www.openehr.org/`. Último acceso: Mayo 2015.

[14] Health Level Seven International (HL7). Health Level Seven Standards. `http://www.hl7.org/`. Último acceso: Mayo 2015.

[15] Clinical Element Model (CEM). CEM Browser. `http://www.clinicalelement.com/#/`. Último acceso: Mayo 2015.

[16] Berners-Lee T, Hendler J, Lassila O (2001) The Semantic Web. Scientific American 284: 34–43.

[17] Goble C, Stevens R (2008) State of the nation in data integration for bioinformatics. Journal of Biomedical Informatics 41: 687–693.

[18] Gruber TR (1993) A translation approach to portable ontology specifications. Knowledge Acquisition 5: 199–220.

[19] Bizer C, Heath T, Berners-Lee T (2009) Linked Data - The Story So Far. International Journal on Semantic Web and Information Systems 5: 1–22.

[20] Linked Data - Connect Distributed Data across the Web. `http://linkeddata.org/`. Último acceso: Mayo 2015.

[21] Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, et al. (2000) Gene Ontology: tool for the unification of biology. Nature genetics 25: 25–29.

[22] Smith B, Ashburner M, Rosse C, Bard J, Bug W, et al. (2007) The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. Nature Biotechnology 25: 1251–1255.

[23] Noy NF, Shah NH, Whetzel PL, Dai B, Dorf M, et al. (2009) BioPortal: ontologies and integrated data resources at the click of a mouse. Nucleic Acids Research 37: W170–W173.

[24] Jupp S, Malone J, Bolleman J, Brandizi M, Davies M, et al. (2014) The EBI RDF Platform: Linked Open Data for the Life Sciences. Bioinformatics : btt765.

[25] Belleau F, Nolin MA, Tourigny N, Rigault P, Morissette J (2008) Bio2rdf: Towards a mashup to build bioinformatics knowledge systems. Journal of Biomedical Informatics 41: 706–716.

[26] Martínez-Costa C, Menárguez-Tortosa M, Fernández-Breis JT (2010) An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes. Journal of Biomedical Informatics 43: 736–746.

[27] Martínez-Costa C, Menárguez-Tortosa M, Fernández-Breis JT (2011) Clinical data interoperability based on archetype transformation. Journal of Biomedical Informatics 44: 869–880.

[28] Menárguez-Tortosa M, Fernández-Breis JT (2013) OWL-based reasoning methods for validating archetypes. Journal of Biomedical Informatics 46: 304–317.

[29] Heiler S (1995) Semantic Interoperability. ACM Computer Surveys 27: 271–273.

[30] Shortliffe EH (1999) The evolution of electronic medical records. Academic Medicine: Journal of the Association of American Medical Colleges 74: 414–419.

[31] Häyrinen K, Saranto K, Nykänen P (2008) Definition, structure, content, use and impacts of electronic health records: a review of the research literature. International Journal of Medical Informatics 77: 291–304.

[32] Griffith SM, Kalra D, Lloyd DS, Ingram D (1995) A portable communicative architecture for electronic healthcare records: the Good European Healthcare Record project (Aim project A2014). Medinfo 8 Pt 1: 223–226.

[33] Beale T (2002) Archetypes: Constraint-based Domain Models for Future-proof Information Systems. In: Workshop on Behavioural Semantics (OOPSLA'02). Seattle.

[34] The openEHR Foundation. Meta-Arquitectura OpenEHR. http://www.openehr.org/programs/specification/releases/1.0.2. Último acceso: Mayo 2015.

[35] EN 13606 Association. CEN/ISO 13606 in use. http://www.en13606.org/ceniso-13606-in-use. Último acceso: Mayo 2015.

[36] ISO. Health informatics - Requirements for an electronic health record architecture - ISO 18308:2011. http://www.iso.org/iso/catalogue_detail?csnumber=52823. Último acceso: Mayo 2015.

[37] Health Level Seven International (HL7). The HL7 Version 3 Clinical Document Architecture (CDA). http://www.hl7.org/implement/standards/product_brief.cfm?product_id=7. Último acceso: Mayo 2015.

[38] Dolin RH, Alschuler L, Boyer S, Beebe C, Behlen FM, et al. (2006) HL7 Clinical Document Architecture, Release 2. Journal of the American Medical Informatics Association : JAMIA 13: 30–39.

[39] Health Level Seven International (HL7). Fast Healthcare Interoperability Resources (FHIR). http://www.hl7.org/FHIR/. Último acceso: Mayo 2015.

[40] HealthITgov. Strategic Health IT Advanced Research Project (SHARP). http://www.healthit.gov/policy-researchers-implementers/strategic-health-it-advanced-research-projects-sharp. Último acceso: Mayo 2015.

[41] The openEHR Foundation. Archetype Definition Language. http://www.openehr.org/downloads/ADLworkbench/learning_about. Último acceso: Mayo 2015.

[42] The openEHR Foundation. Clinical Knowledge Manager. http://www.openehr.org/ckm. Último acceso: Mayo 2015.

[43] IBIME ITACA Universitat Politècnica de València. LinkEHR Platform. http://www.linkehr.com/. Último acceso: Mayo 2015.

[44] EN 13606 Association. Clinical Information Model Manager. http://cimm.en13606.org/. Último acceso: Mayo 2015.

[45] Mooney SD, Tenenbaum JD, Altman RB (2014) Bioinformatics. In: Shortliffe EH, Cimino JJ, editors, Biomedical Informatics, Springer London. pp. 695–719.

[46] Hagen JB (2000) The origins of bioinformatics. Nature Reviews Genetics 1: 231–236.

[47] Sanger F (1959) Chemistry of Insulin Determination of the structure of insulin opens the way to greater understanding of life processes. Science 129: 1340–1344.

[48] Sawicki MP, Samara G, Hurwitz M, Passaro Jr E (1993) Human Genome Project. The American Journal of Surgery 165: 258–264.

[49] Guyer M (1998) Statement on the Rapid Release of Genomic DNA Sequence: Notes from the meeting. Genome Research 8: 413–413.

[50] Benson DA, Cavanaugh M, Clark K, Karsch-Mizrachi I, Lipman DJ, et al. (2013) GenBank. Nucleic Acids Research 41: D36–42.

[51] Stevens R, Lord P (2009) Application of Ontologies in Bioinformatics. In: Staab S, Studer R, editors, Handbook on Ontologies, Springer Berlin Heidelberg, International Handbooks on Information Systems. pp. 735–756.

[52] The UniProt Consortium (2010) The Universal Protein Resource (UniProt) in 2010. Nucleic Acids Research 38: D142–D148.

[53] Rose PW, Bi C, Bluhm WF, Christie CH, Dimitropoulos D, et al. (2013) The RCSB Protein Data Bank: new resources for research and education. Nucleic Acids Research 41: D475–D482.

[54] National Center for Biotechnology Information (NCBI). Genome. `http://www.ncbi.nlm.nih.gov/genome`. Último acceso: Mayo 2015.

[55] Eppig JT, Blake JA, Bult CJ, Kadin JA, Richardson JE, et al. (2015) The Mouse Genome Database (MGD): facilitating mouse as a model for human biology and disease. Nucleic Acids Research 43: D726–736.

[56] National Institute of Health (NIH). U. S. National Library of Medicine(NLM). `http://www.nlm.nih.gov/`. Último acceso: Mayo 2015.

[57] National Center for Biotechnology Information (NCBI). PubMed. `http://www.ncbi.nlm.nih.gov/pubmed`. Último acceso: Mayo 2015.

[58] BioPerl. FASTA Format. http://www.bioperl.org/wiki/FASTA_sequence_format. Último acceso: Mayo 2015.

[59] BioPerl. FASTQ sequence format. http://www.bioperl.org/wiki/FASTQ_sequence_format. Último acceso: Mayo 2015.

[60] Wellcome Trust Sanger Institute. General Feature Format (GFF) Specifications Document. https://www.sanger.ac.uk/resources/software/gff/spec.html#t_1. Último acceso: Mayo 2015.

[61] Schmitt T, Messina DN, Schreiber F, Sonnhammer ELL (2011) SeqXML and OrthoXML: standards for sequence and orthology information. Briefings in Bioinformatics : bbr025.

[62] Stockholm University. Stockholm Bioinformatics Center (SBC). http://www.sbc.su.se/. Último acceso: Mayo 2015.

[63] Östlund G, Schmitt T, Forslund K, Köstler T, Messina DN, et al. (2010) InParanoid 7: new algorithms and tools for eukaryotic orthology analysis. Nucleic Acids Research 38: D196–D203.

[64] Mi H, Dong Q, Muruganujan A, Gaudet P, Lewis S, et al. (2009) PANTHER version 7: improved phylogenetic trees, orthologs and collaboration with the Gene Ontology Consortium. Nucleic Acids Research : gkp1019.

[65] Huerta-Cepas J, Capella-Gutierrez S, Pryszcz LP, Denisov I, Kormes D, et al. (2011) PhylomeDB v3.0: an expanding repository of genome-wide collections of trees, alignments and phylogeny-based orthology and paralogy predictions. Nucleic Acids Research 39: D556–D560.

[66] Hanisch D, Zimmer R, Lengauer T (2002) ProML–the protein markup language for specification of protein sequences, structures and families. In Silico Biology 2: 313–324.

[67] Fenyö D (1999) The Biopolymer Markup Language. Bioinformatics (Oxford, England) 15: 339–340.

[68] Generic Model Organism Database project (GMOD). Chado XML. http://gmod.org/wiki/Chado_XML. Último acceso: Mayo 2015.

[69] Mungall CJ, Emmert DB (2007) A Chado case study: an ontology-based modular schema for representing genome-associated biological information. Bioinformatics 23: i337–i346.

[70] Codd EF (1970) A Relational Model of Data for Large Shared Data Banks. Communications of the ACM 13: 377–387.

[71] dos Santos G, Schroeder AJ, Goodman JL, Strelets VB, Crosby MA, et al. (2014) FlyBase: introduction of the Drosophila melanogaster Release 6 reference genome assembly and large-scale migration of genome annotations. Nucleic Acids Research : gku1099.

[72] Gene Ontology Consortium. GO Database. `http://geneontology.org/page/go-database`. Último acceso: Mayo 2015.

[73] Robinson I, Webber J, Eifrem E (2013) Graph Databases. O'Reilly Media, Inc.

[74] Vishveshwara S, Brinda KV, Kannan N (2002) Protein structure: insights from graph theory. Journal of Theoretical and Computational Chemistry 01: 187–211.

[75] Pareja-Tobes P, Pareja-Tobes E, Manrique M, Pareja E, Tobes R (2013) Bio4j: An Open source biological data integration platform. In: Proceedings of IWBBIO 2013. Granada, Spain, p. 281.

[76] Magrane M, The UniProt Consortium (2011) UniProt Knowledgebase: a hub of integrated protein data. Database 2011: bar009.

[77] Suzek BE, Huang H, McGarvey P, Mazumder R, Wu CH (2007) UniRef: comprehensive and non-redundant UniProt reference clusters. Bioinformatics 23: 1282–1288.

[78] Pruitt KD, Tatusova T, Maglott DR (2007) NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Research 35: D61–D65.

[79] Federhen S (2012) The NCBI Taxonomy database. Nucleic Acids Research 40: D136–D143.

[80] Gasteiger E, Gattiker A, Hoogland C, Ivanyi I, Appel RD, et al. (2003) ExPASy: the proteomics server for in-depth protein knowledge and analysis. Nucleic Acids Research 31: 3784–3788.

[81] Rector A (2008) Deliverable 6.1: Barriers, approaches and research priorities for integrating biomedical ontologies. Technical Report 6.1, European Commission.

[82] Bodenreider O (2004) The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Research 32: D267–D270.

[83] Chute CG (2000) Clinical Classification and Terminology. Journal of the American Medical Informatics Association : JAMIA 7: 298–303.

[84] International Classification of Diseases (ICD). `http://www.who.int/classifications/icd/en/`. Último acceso: Mayo 2015.

[85] Logical Observation Identifiers Names and Codes (LOINC). `http://loinc.org/`. Último acceso: Mayo 2015.

[86] International Health Terminology Standards Development Organisation. Systematized Nomenclature of Medicine Clinical Terms (SNOMED CT). `http://www.ihtsdo.org/snomed-ct`. Último acceso: Mayo 2015.

[87] Martínez-Costa C, Menárguez-Tortosa M, Fernández-Breis JT, Maldonado JA (2009) A model-driven approach for representing clinical archetypes for Semantic Web environments. Journal of Biomedical Informatics 42: 150–164.

[88] Wang X, Gorlitsky R, Almeida JS (2005) From XML to RDF: how semantic web technologies will change the design of ómic'standards. Nature Biotechnology 23: 1099–1103.

[89] Clinical Infomation Modeling Initiative (CIMI). `http://opencimi.org/`. Último acceso: Mayo 2015.

[90] SemanticHealthNet. `http://www.semantichealthnet.eu/`. Último acceso: Mayo 2015.

[91] Schulz S, Martínez-Costa C (2013) How Ontologies Can Improve Semantic Interoperability in Health Care. In: Riaño D, Lenz R, Miksch S, Peleg M, Reichert M, et al., editors, Process Support and Knowledge Representation in Health Care, Springer International Publishing, number 8268 in Lecture Notes in Computer Science. pp. 1–10.

[92] Gangemi A, Presutti V (2009) Ontology Design Patterns. In: Staab S, Studer R, editors, Handbook on Ontologies, Springer Berlin Heidelberg, International Handbooks on Information Systems. pp. 221–243.

[93] Berners-Lee T. Uniform Resource Identifiers (URI): Generic Syntax. `http://tools.ietf.org/html/rfc2396`. Último acceso: Mayo 2015.

[94] W3C. Extensible Markup Language (XML). http://www.w3.org/XML/. Último acceso: Mayo 2015.

[95] W3C. XML Schema. http://www.w3.org/XML/Schema. Último acceso: Mayo 2015.

[96] RDF Working Group. Resource Description Framework (RDF). http://www.w3.org/RDF/. Último acceso: Mayo 2015.

[97] W3C. Resource Description Framework Schema (RDFS) 1.1. http://www.w3.org/TR/rdf-schema/. Último acceso: Mayo 2015.

[98] Brewster C, O'Hara K (2004) Knowledge representation with ontologies: the present and future. IEEE Intelligent Systems 19: 72–81.

[99] Guarino N (1998) Formal Ontology and Information Systems. IOS Press, pp. 3–15.

[100] W3C. OWL Web Ontology Language. http://www.w3.org/TR/owl-features/. Último acceso: Mayo 2015.

[101] Baader F, Nutt W (2003) Basic description logics. In: Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF, editors, The Description Logic Handbook, New York, NY, USA: Cambridge University Press. pp. 43–95.

[102] W3C. OWL 2 Web Ontology Language. http://www.w3.org/TR/owl2-overview/. Último acceso: Mayo 2015.

[103] Baader F, Brand S, Lutz C (2005) Pushing the EL envelope. In: Proceedings of International Joint Conferences on Artificial Intelligence (IJCAI 2005). Morgan-Kaufmann Publishers, pp. 364–369.

[104] W3C. Inference. http://www.w3.org/standards/semanticweb/inference. Último acceso: Mayo 2015.

[105] Horrocks I, Patel-schneider PF, Mcguinness DL, Welty CA (2007) OWL: a Description Logic Based Ontology Language for the Semantic Web. In: Baader F, Calvanese D, McGuinness DL, Nardi D, Patel-Schneider PF, editors, The Description Logic Handbook: Theory, Implementation, and Applications, Cambridge University Press. 2nd edition edition.

[106] Sirin E, Parsia B, Grau BC, Kalyanpur A, Katz Y (2007) Pellet: A practical OWL-DL reasoner. Web Semantics: Science, Services and Agents on the World Wide Web 5: 51–53.

[107] Horrocks I (1998) The FaCT system. In: Proceedings of the International Conference on Automated Reasoning with Analytic Tableaux and Related Methods (TABLEAUX '98), volume 1397 in Lecture Notes in Artificial Intelligence. Springer, pp. 307–312.

[108] Shearer R, Motik B, Horrocks I (2008) HermiT: A Highly-Efficient OWL Reasoner. In: OWLED. volume 432.

[109] W3C. SPARQL Query Language for RDF. `http://www.w3.org/TR/rdf-sparql-query/`. Último acceso: Mayo 2015.

[110] Berners-Lee T. Linked Data - Design Issues. `http://www.w3.org/DesignIssues/LinkedData.html`.

[111] Heath T, Bizer C (2011) Linked Data: Evolving the Web into a Global Data Space. Synthesis Lectures on the Semantic Web: Theory and Technology 1: 1–136.

[112] W3C. The Linking Open Data cloud diagram . `http://lod-cloud.net/`. Último acceso: Mayo 2015.

[113] Lenat DB, Guha RV (1989) Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st edition.

[114] Uschold M, King M (1995) Towards a Methodology for Building Ontologies. In: Workshop on Basic Ontological Issues in Knowledge Sharing, held in conjunction with IJCAI-95.

[115] Sure Y, Staab S, Studer R (2009) Ontology Engineering Methodology. In: Handbook on Ontologies, Berlin Heidelberg: Springer Berlin Heidelberg.

[116] Gruninger M, Fox MS (1996) The Logic of Enterprise Modelling. In: Bernus P, Nemes L, editors, Modelling and Methodologies for Enterprise Integration, Springer US, IFIP - The International Federation for Information Processing. pp. 140–157.

[117] Bernaras A, Laresgoiti I, Correa J (1996) Building and reusing ontologies for electrical network applications. In: Proceedings of the European

Conference on Artificial Intelligence (ECAI96). Hungary, pp. 298–302. Undefined European Conference for Artificial Intelligence Building and reusing ontologies for electrical network applications.

[118] Lopez M, Gomez-Perez A, Sierra J, Sierra A (1999) Building a chemical ontology using Methontology and the Ontology Design Environment. IEEE Intelligent Systems and their Applications 14: 37–46.

[119] Gómez-Pérez A (1996) Towards a framework to verify knowledge sharing technology. Expert Systems with Applications 11: 519–529.

[120] Swartout B, Patil R, Knight K, Russ T (1997) Toward Distributed Use of Large-Scale Ontologies. In: Ontological Engineering, AAAI-97 Spring Symposium Series. pp. 138–148.

[121] Pinto HS, Tempich C, Staab S (2009) Ontology Engineering and Evolution in a Distributed World Using DILIGENT. In: Staab S, Studer R, editors, Handbook on Ontologies, Springer Berlin Heidelberg, International Handbooks on Information Systems. pp. 153–176.

[122] Fernández-Breis JT (2003) Un entorno de integración de ontologías para el desarrollo de sistemas de gestión del conocimiento. Tesis Doctoral, Universidad de Murcia. `https://digitum.um.es/xmlui/handle/10201/185`.

[123] Neches R, Fikes RE, Finin T, Gruber T, Patil R, et al. (1991) Enabling Technology for Knowledge Sharing. AI Magazine 12: 36.

[124] Simperl E (2009) Reusing ontologies on the Semantic Web: A feasibility study. Data & Knowledge Engineering 68: 905–925.

[125] Gómez-Pérez A, Rojas-Amaya MD (1999) Ontological Reengineering for Reuse. In: Fensel D, Studer R, editors, Knowledge Acquisition, Modeling and Management, Springer Berlin Heidelberg, number 1621 in Lecture Notes in Computer Science. pp. 139–156.

[126] Uschold M, Healy M, Williamson K, Clark P, Woods S (1998) Ontology Reuse and Application. In: Proceedings of the 1st International Conference on Formal Ontology in Information Systems (FOIS). IOS Press, pp. 179–192.

[127] Russ T, Valente A, MacGregor R, Swartout W (1999) Practical Experiences in Trading Off Ontology Usability and Reusability. In: Proceedings of the 12th Workshop on Knowledge Acquisition, Modeling and Management (KAW'99). pp. 16–21.

[128] Rector A, Brandt S, Drummond N, Horridge M, Pulestin C, et al. (2012) Engineering use cases for modular development of ontologies in OWL. Applied Ontology 7: 113–132.

[129] Peralta DN, Pinto HS, Mamede NJ (2004) Reusing a Time Ontology. In: Camp O, Filipe JBL, Hammoudi S, Piattini M, editors, Enterprise Information Systems V, Springer Netherlands. pp. 241–248.

[130] Coulet A, Smaïl-Tabbone M, Napoli A, Devignes MD (2006) Suggested Ontology for Pharmacogenomics (SO-Pharm): Modular Construction and Preliminary Testing. In: Meersman R, Tari Z, Herrero P, editors, On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, Springer Berlin Heidelberg, number 4277 in Lecture Notes in Computer Science. pp. 648–657.

[131] Arpírez JC, Gómez-Pérez A, Lozano-Tello A, Pinto HSANP (2000) Reference Ontology and (ONTO)2 Agent: The Ontology Yellow Pages. Knowledge and Information Systems 2: 387–412.

[132] Egaña M, Rector A, Stevens R, Antezana E (2008) Applying ontology design patterns in bio-ontologies. In: Knowledge Engineering: Practice and Patterns, Springer. pp. 7–16.

[133] Blomqvist E (2009) Semi-automatic Ontology Construction based on Patterns. Tesis Doctoral. `http://liu.diva-portal.org/smash/record.jsf?pid=diva2%3A207543&dswid=-914`.

[134] Presutti V, Blomqvist E, Daga E, Gangemi A (2012) Pattern-Based Ontology Design. In: Suárez-Figueroa MC, Gómez-Pérez A, Motta E, Gangemi A, editors, Ontology Engineering in a Networked World, Springer Berlin Heidelberg. pp. 35–64.

[135] Baker CJO, Shaban-Nejad A, Su X, Haarslev V, Butler G (2011) Semantic Web Infrastructure for Fungal Enzyme Biotechnologists. Web Semantics: Science, Services and Agents on the World Wide Web 4.

[136] Egaña-Aranguren M, Antezana E, Kuiper M, Stevens R (2008) Ontology Design Patterns for bio-ontologies: a case study on the Cell Cycle Ontology. BMC Bioinformatics 9: S1.

[137] NeOn Project. Ontology Design Patterns .org (ODP). `http://ontologydesignpatterns.org/wiki/Main_Page`. Último acceso: Mayo 2015.

[138] Iannone L, Rector A, Stevens R (2009) Embedding Knowledge Patterns into OWL. In: Aroyo L, Traverso P, Ciravegna F, Cimiano P, Heath T, et al., editors, The Semantic Web: Research and Applications, Springer Berlin Heidelberg, number 5554 in Lecture Notes in Computer Science. pp. 218–232.

[139] Iannone L, Egaña M, Rector A, Stevens R (2008) Augmenting the expressivity of the Ontology Pre-Processor Language. In: Proceedings of the 5th International workshop on OWL Experiences and Directions 2008. Karlsruhe, Germany.

[140] Martínez-Costa C, Schulz S (2014) Ontology content patterns as bridge for the semantic representation of clinical information. Applied Clinical Informatics 5: 660–669.

[141] Kiryakov A, Popov B, Terziev I, Manov D, Ognyanoff D (2004) Semantic annotation, indexing, and retrieval. Web Semantics: Science, Services and Agents on the World Wide Web 2: 49–79.

[142] Oren E, Möller K, Scerri S, Handschuh S, Sintek M (2006) What are semantic annotations? Technical Report, Digitial Enterprise Research Institute (DERI), Galway.

[143] Pesquita C, Faria D, Falcao AO, Lord P, Couto FM (2009) Semantic similarity in biomedical ontologies. PLoS computational biology 5: e1000443.

[144] Hliaoutakis A, Varelas G, Voutsakis E, Petrakis EG, Milios E (2006) Information Retrieval by Semantic Similarity:. International Journal on Semantic Web and Information Systems 2: 55–73.

[145] Rada R, Mili H, Bicknell E, Blettner M (1989) Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man and Cybernetics 19: 17–30.

[146] Lord PW, Stevens RD, Brass A, Goble CA (2003) Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. Bioinformatics 19: 1275–1283.

[147] Resnik P (1999) Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. Journal of Artificial Intelligence Research 11: 95–130.

[148] Rodriguez M, Egenhofer M (2003) Determining semantic similarity among entity classes from different ontologies. IEEE Transactions on Knowledge and Data Engineering 15: 442–456.

[149] Bodenreider O (2008) Biomedical Ontologies in Action: Role in Knowledge Management, Data Integration and Decision Support. Yearbook of medical informatics : 67–79.

[150] Stanford University. Saccharomyces Genome Database. `http://www.yeastgenome.org/`. Último acceso: Mayo 2015.

[151] Musen MA, Noy NF, Shah NH, Whetzel PL, Chute CG, et al. (2012) The National Center for Biomedical Ontology. Journal of the American Medical Informatics Association: JAMIA 19: 190–195.

[152] Whetzel PL, Noy NF, Shah NH, Alexander PR, Nyulas C, et al. (2011) BioPortal: enhanced functionality via new Web services from the National Center for Biomedical Ontology to access and use ontologies in software applications. Nucleic Acids Research 39: W541–W545.

[153] European Molecular Biology Laboratory. The European Bioinformatics Institute (EMBL-EBI). `http://www.ebi.ac.uk/`. Último acceso: Mayo 2015.

[154] EMBL-EBI. RDF Platform. `https://www.ebi.ac.uk/rdf/platform`. Último acceso: Mayo 2015.

[155] EMBL-EBI. BioModels Database - RDF Platform. `https://www.ebi.ac.uk/rdf/services/biomodels/`. Último acceso: Mayo 2015.

[156] EMBL-EBI. BioSamples Database - RDF Platform. `https://www.ebi.ac.uk/rdf/services/biosamples/`. Último acceso: Mayo 2015.

[157] EMBL-EBI. ChEMBL - RDF Platform. `https://www.ebi.ac.uk/rdf/services/chembl/`. Último acceso: Mayo 2015.

[158] EMBL-EBI. Gene Expression Atlas - RDF Platform. `https://www.ebi.ac.uk/rdf/services/atlas/`. Último acceso: Mayo 2015.

[159] EMBL-EBI. Reactome - RDF Platform. `https://www.ebi.ac.uk/rdf/services/reactome/`. Último acceso: Mayo 2015.

[160] EMBL-EBI. UniProt Linked Data - RDF Platform. `https://www.ebi.ac.uk/rdf/services/uniprot/`. Último acceso: Mayo 2015.

[161] Tudorache T, Nyulas CI, Noy NF, Musen MA (2013) Using Semantic Web in ICD-11: Three Years Down the Road. In: Alani H, Kagal L, Fokoue A, Groth P, Biemann C, et al., editors, The Semantic Web ISWC 2013, Springer Berlin Heidelberg, number 8219 in Lecture Notes in Computer Science. pp. 195–211.

[162] Rector AL, Brandt S (2008) Why Do It the Hard Way? The Case for an Expressive Description Logic for SNOMED. Journal of the American Medical Informatics Association : JAMIA 15: 744–751.

[163] Callahan A, Cruz-Toledo J, Ansell P, Dumontier M (2013) Bio2rdf Release 2: Improved Coverage, Interoperability and Provenance of Life Science Linked Data. In: Cimiano P, Corcho O, Presutti V, Hollink L, Rudolph S, editors, The Semantic Web: Semantics and Big Data, Springer Berlin Heidelberg, number 7882 in Lecture Notes in Computer Science. pp. 200–212.

[164] Bio2RDF Release 3, Datasets. `http://download.bio2rdf.org/release/3/release.html`. Último acceso: Mayo 2015.

[165] Miñarro-Giménez JA, Egaña Aranguren M, Villazón-Terrazas B, Fernández Breis JT (2014) Translational research combining orthologous genes and human diseases with the OGOLOD dataset. Semantic Web 5: 145–149.

[166] Tatusov RL, Fedorova ND, Jackson JD, Jacobs AR, Kiryutin B, et al. (2003) The COG database: an updated version includes eukaryotes. BMC Bioinformatics 4: 41.

[167] National Center for Biotechnology Information (NCBI). HomoloGene. `http://www.ncbi.nlm.nih.gov/homologene`. Último acceso: Mayo 2015.

[168] Li L, Stoeckert CJ, Roos DS (2003) OrthoMCL: Identification of Ortholog Groups for Eukaryotic Genomes. Genome Research 13: 2178–2189.

[169] Hamosh A, Scott AF, Amberger JS, Bocchini CA, McKusick VA (2005) Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic Acids Research 33: D514–D517.

[170] Chibucos MC, Mungall CJ, Balakrishnan R, Christie KR, Huntley RP, et al. (2014) Standardized description of scientific evidence using the Evidence Ontology (ECO). Database 2014: bau075–bau075.

[171] Smith B, Ceusters W, Klagges B, Köhler J, Kumar A, et al. (2005) Relations in biomedical ontologies. Genome Biology 6: R46.

[172] Köhler S, Doelken SC, Mungall CJ, Bauer S, Firth HV, et al. (2014) The Human Phenotype Ontology project: linking molecular biology and disease through phenotype data. Nucleic Acids Research 42: D966–D974.

[173] Martínez-Costa C, Menárguez-Tortosa M, Fernández-Breis JT. Poseacle converter. `http://miuras.inf.um.es/PoseacleConverter/`. Último acceso: Mayo 2015.

[174] Object Management Group. Ontology Definition Metamodel 1.0. `http://www.omg.org/spec/ODM/1.0/`. Último acceso: Mayo 2015.

[175] Menárguez-Tortosa M, Fernández-Breis JT. Archeck. `http://miuras.inf.um.es/archeck`. Último acceso: Mayo 2015.

[176] Lezcano L, Sicilia MA, Serrano-Balazote P (2008) Combining OpenEHR Archetype Definitions with SWRL Rules A Translation Approach. In: Lytras MD, Carroll JM, Damiani E, Tennyson RD, editors, Emerging Technologies and Information Systems for the Knowledge Society, Springer Berlin Heidelberg, number 5288 in Lecture Notes in Computer Science. pp. 79–87.

[177] Tao C, Jiang G, Oniki TA, Freimuth RR, Zhu Q, et al. (2013) A semantic-web oriented representation of the clinical element model for secondary use of electronic health records data. Journal of the American Medical Informatics Association: JAMIA 20: 554–562.

[178] Iqbal AM (2011) An OWL-DL ontology for the HL7 reference information model. In: Toward Useful Services for Elderly and People with Disabilities, Springer. pp. 168–175.

[179] Hernandez T, Kambhampati S (2004) Integration of Biological Sources: Current Systems and Challenges Ahead. Sigmod Record 33: 51–60.

[180] Uschold M, Gruninger M (2004) Ontologies and Semantics for Seamless Connectivity. ACM SIGMOD Record 33: 58–64.

[181] Spanos DE, Stavrou P, Mitrou N (2012) Bringing Relational Databases into the Semantic Web: A Survey. Semantic Web 3: 169–209.

[182] Cimiano P, Mädche A, Staab S, Völker J (2009) Ontology Learning. In: Staab S, Studer R, editors, Handbook on Ontologies, Berlin, Heidelberg: Springer Berlin Heidelberg, International Handbooks on Information Systems. pp. 245–267.

[183] Maedche A (2002) Ontology Learning for the Semantic Web, volume 665 of *The Kluwer International Series in Engineering and Computer Science*. Boston, MA: Springer US.

[184] Berners-Lee T. Relational Databases on the Semantic Web. `http://www.w3.org/DesignIssues/RDB-RDF.html`. Último acceso: Mayo 2015.

[185] W3C. RDB2RDF Working Group. `http://www.w3.org/2001/sw/rdb2rdf/`. Último acceso: Mayo 2015.

[186] W3C. A Direct Mapping of Relational Data to RDF. `http://www.w3.org/TR/2012/REC-rdb-direct-mapping-20120927/`. Último acceso: Mayo 2015.

[187] Rahm E (2011) Towards Large-Scale Schema and Ontology Matching. In: Bellahsene Z, Bonifati A, Rahm E, editors, Schema Matching and Mapping, Springer Berlin Heidelberg, Data-Centric Systems and Applications. pp. 3–27.

[188] W3C. R2RML: RDB to RDF Mapping Language. `http://www.w3.org/TR/2012/REC-r2rml-20120927/`. Último acceso: Mayo 2015.

[189] W3C. Turtle. Terse RDF Triple Language. `http://www.w3.org/TR/2012/WD-turtle-20120710/`. Último acceso: Mayo 2015.

[190] Freie Universität. The D2RQ Mapping Language - The D2RQ Platform. `http://d2rq.org/d2rq-language`. Último acceso: Mayo 2015.

[191] Bizer C, Seaborne A (2004) D2RQ - treating non-RDF databases as virtual RDF graphs. In: Proceedings of the 3rd International Semantic Web Conference (ISWC2004).

[192] Freie Universität. D2R Server - The D2RQ Platform. `http://d2rq.org/d2r-server`. Último acceso: Mayo 2015.

[193] Auer S, Dietzold S, Lehmann J, Hellmann S, Aumueller D (2009) Triplify: Light-weight Linked Data Publication from Relational Databases. In: Proceedings of the 18th International Conference on World

Wide Web. New York, NY, USA: ACM, WWW '09, pp. 621–630. doi: 10.1145/1526709.1526793.

[194] Erling O, Mikhailov I (2009) RDF Support in the Virtuoso DBMS. In: Pellegrini T, Auer S, Tochtermann K, Schaffert S, editors, Networked Knowledge - Networked Media, Springer Berlin Heidelberg, number 221 in Studies in Computational Intelligence. pp. 7–24.

[195] OpenLink. Virtuoso Open-Source: Mapping Relational Data to RDF in Virtuoso. `http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VOSSQLRDF`. Último acceso: Mayo 2015.

[196] Tsinaraki C, Christodoulakis S (2007) XS2owl: A Formal Model and a System for Enabling XML Schema Applications to Interoperate with OWL-DL Domain Knowledge and Semantic Web Tools. In: Thanos C, Borri F, Candela L, editors, Digital Libraries: Research and Development, Springer Berlin Heidelberg, number 4877 in Lecture Notes in Computer Science. pp. 124–136.

[197] Stavrakantonakis I, Tsinaraki C, Bikakis N, Gioldasis N, Christodoulakis S (2010) SPARQL2xquery 2.0: Supporting Semantic-based queries over XML data. In: 2010 5th International Workshop on Semantic Media Adaptation and Personalization (SMAP). pp. 76–84. doi: 10.1109/SMAP.2010.5706860.

[198] Bumans G, Cerans K (2010) RDB2owl: A Practical Approach for Transforming RDB Data into RDF/OWL. In: Proceedings of the 6th International Conference on Semantic Systems. New York, NY, USA: ACM, I-SEMANTICS '10, pp. 25:1–25:3. doi:10.1145/1839707.1839739.

[199] Knoblock CA, Szekely P, Ambite JL, Goel A, Gupta S, et al. (2012) Semi-automatically Mapping Structured Sources into the Semantic Web. In: Simperl E, Cimiano P, Polleres A, Corcho O, Presutti V, editors, The Semantic Web: Research and Applications, Springer Berlin Heidelberg, number 7295 in Lecture Notes in Computer Science. pp. 375–390.

[200] Jupp S, Horridge M, Iannone L, Klein J, Owen S, et al. (2012) Populous: a tool for building OWL ontologies from templates. BMC Bioinformatics 13: S5.

[201] Miñarro Giménez JA (2012) Entorno para la gestión semántica de información biomédica en investigación traslacional. Tesis Doctoral, Universidad de Murcia. http://digitum.um.es/xmlui/handle/10201/27691.

[202] Sujansky W (2001) Heterogeneous Database Integration in Biomedicine. Journal of Biomedical Informatics 34: 285–298.

[203] Louie B, Mork P, Martin-Sanchez F, Halevy A, Tarczy-Hornoch P (2007) Data integration and genomic medicine. Journal of Biomedical Informatics 40: 5–16.

[204] Calvanese D, Giacomo GD (2005) Data Integration: A Logic-Based Perspective. AI Magazine 26: 59–70.

[205] Pasquier C (2008) Biological data integration using Semantic Web technologies. Biochimie 90: 584–594.

[206] Stein LD (2003) Integrating biological databases. Nature Reviews Genetics 4: 337–345.

[207] Davidson SB, Overton C, Buneman P (1995) Challenges in Integrating Biological Data Sources. Journal of Computational Biology 2: 557–572.

[208] Hammer J, Schneider M (2002) Genomics Algebra: A New, Integrating Data Model, Language, and Tool for Processing and Querying Genomic Information. In: Proceedings of the 2002 CIDR Conference. pp. 176–187.

[209] Critchlow T, Fidelis K, Ganesh M, Musick R, Slezak T (2000) DataFoundry: information management for scientific data. IEEE transactions on information technology in biomedicine: a publication of the IEEE Engineering in Medicine and Biology Society 4: 52–57.

[210] Chute CG, Beck SA, Fisk TB, Mohr DN (2010) The Enterprise Data Trust at Mayo Clinic: a semantically integrated warehouse of biomedical data. Journal of the American Medical Informatics Association : JAMIA 17: 131–135.

[211] Lenzerini M (2002) Data Integration: A Theoretical Perspective. In: Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems. New York, NY, USA: ACM, PODS '02, pp. 233–246. doi:10.1145/543613.543644.

[212] Sheth A, Larson J (1990) Federated Database Systems for managing Distributed Heterogeneous, and Autonomous Databases. ACM Computing Surveys 22: 183–236.

[213] Astakhov V, Gupta A, Santini S, Grethe JS (2005) Data Integration in the Biomedical Informatics Research Network (BIRN). In: Ludäscher B, Raschid L, editors, Data Integration in the Life Sciences, Springer Berlin Heidelberg, number 3615 in Lecture Notes in Computer Science. pp. 317–320.

[214] Zdobnov EM, Lopez R, Apweiler R, Etzold T (2002) The EBI SRS server–recent developments. Bioinformatics (Oxford, England) 18: 368–373.

[215] Feigenbaum L, Herman I, Hongsermeier T, Neumann E, Stephens S (2007) The Semantic Web in Action. Scientific American 297: 64–71.

[216] Wache H, Vögele T, Visser U, Stuckenschmidt H, Schuster G, et al. (2001) Ontology-Based Integration of Information - A Survey of Existing Approaches. In: Proceedings of the IJCAI-01 Workshop on Ontologies and Information Sharing. pp. 108–117.

[217] Kalfoglou Y, Schorlemmer M (2003) Ontology Mapping: The State of the Art. The Knowledge Engineering Review 18: 1–31.

[218] Euzenat J, Shvaiko P (2013) Ontology Matching. Berlin, Heidelberg: Springer Berlin Heidelberg.

[219] W3C. SWRL: A Semantic Web Rule Language Combining OWL and RuleML. `http://www.w3.org/Submission/SWRL/`. Último acceso: Mayo 2015.

[220] Euzenat J, Scharffe F, Zimmermann A (2007) Expressive alignment language and implementation. Knowledge Web Consortium D2.2.10.

[221] David J, Euzenat J, Scharffe F, Trojahn dos Santos C (2011) The Alignment API 4.0. Semantic Web 2: 3–10.

[222] Chen H, Yu T, Chen JY (2013) Semantic Web meets Integrative Biology: a survey. Briefings in Bioinformatics 14: 109–125.

[223] Cheung KH, Yip KY, Smith A, deKnikker R, Masiar A, et al. (2005) YeastHub: a semantic web use case for integrating data in the life sciences domain. Bioinformatics 21: i85–i96.

[224] Miñarro-Giménez JA, Egaña Aranguren M, Martínez Béjar R, Fernández-Breis JT, Madrid M (2011) Semantic integration of information about orthologs and diseases: The OGO system. Journal of Biomedical Informatics 44: 1020–1031.

[225] Rebholz-Schuhmann D, Grabmüller C, Kavaliauskas S, Croset S, Woollard P, et al. (2014) A case study: semantic integration of gene-disease associations for type 2 diabetes mellitus from literature and biomedical data resources. Drug Discovery Today 19: 882–889.

[226] Kapushesky M, Emam I, Holloway E, Kurnosov P, Zorin A, et al. (2010) Gene Expression Atlas at the European Bioinformatics Institute. Nucleic Acids Research 38: D690–D698.

[227] EMBL-EBI. Lexical Entities of Biological Interest (LexEBI). `http://www.ebi.ac.uk/Rebholz-srv/LexEBI/index.html`. Último acceso: Mayo 2015.

[228] Malone J, Holloway E, Adamusiak T, Kapushesky M, Zheng J, et al. (2010) Modeling sample variables with an Experimental Factor Ontology. Bioinformatics 26: 1112–1118.

[229] Schriml LM, Arze C, Nadendla S, Chang YWW, Mazaitis M, et al. (2012) Disease Ontology: a backbone for disease semantic integration. Nucleic Acids Research 40: D940–D946.

[230] Noy NF, Klein M (2004) Ontology Evolution: Not the Same as Schema Evolution. Knowledge and Information Systems 6: 428–440.

[231] Legaz-García MdC, Miñarro-Giménez JA, Fernández-Breis JT. Semantic Web Integration Tool (SWIT). `http://sele.inf.um.es/swit`. Último acceso: Mayo 2015.

[232] Bhatt M, Rahayu W, Soni SP, Wouters C (2009) Ontology driven semantic profiling and retrieval in medical information systems. Web Semantics: Science, Services and Agents on the World Wide Web 7: 317–331.

[233] Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions and reversals. In: Soviet physics doklady. volume 10, p. 707.

[234] Protégé. DL-query tab. `http://protegewiki.stanford.edu/wiki/DLQueryTab`. Último acceso: Mayo 2015.

[235] Horridge M, Bechhofer S (2011) The OWL API: A Java API for OWL ontologies. Semantic Web 2: 11–21.

[236] Dentler K, Numans ME, Teije At, Cornet R, Keizer NFd (2014) Formalization and computation of quality measures based on electronic medical records. Journal of the American Medical Informatics Association : JAMIA 21: 285–291.

[237] Legaz-García MdC, Martínez-Costa C, Menárguez-Tortosa M, Fernández-Breis JT. Archetype Management Tool (ArchMS). `http://sele.inf.um.es/archms`. Último acceso: Mayo 2015.

[238] The Apache Foundation. Apache Jena. `https://jena.apache.org/`. Último acceso: Mayo 2015.

[239] The Apache Foundation. Lucene. `https://lucene.apache.org/core/`. Último acceso: Mayo 2015.

[240] Menárguez-Tortosa M, Martínez-Costa C, Fernández-Breis JT (2012) A generative tool for building health applications driven by ISO 13606 archetypes. Journal of Medical Systems 36: 3063–75.

[241] Martínez-Costa C, Miñarro-Giménez JA, Menárguez-Tortosa M, Valencia-García R, Fernández-Breis JT (2010) Flexible semantic querying of clinical archetypes. In: Knowledge-Based and Intelligent Information and Engineering Systems, Springer. pp. 597–606.

[242] Bohl O, Scheuhase J, Sengler R, Winand U (2002) The sharable content object reference model (SCORM) - a critical review. In: International Conference on Computers in Education, 2002. Proceedings. pp. 950–951 vol.2. doi:10.1109/CIE.2002.1186122.

[243] Fernández-Breis JT, Maldonado JA, Marcos M, Legaz-García MdC, Moner D, et al. (2013) Leveraging electronic healthcare record standards and semantic web technologies for the identification of patient cohorts. Journal of the American Medical Informatics Association: JAMIA 20: e288–296.

[244] Atkin W, Valori R, Kuipers E, Hoff G, Senore C, et al. (2012) European guidelines for quality assurance in colorectal cancer screening and diagnosis. Endoscopy 10: 0032–1309821.

[245] National Institute for Health and Care Excellence. Colorectal cancer: The diagnosis and management of colorectal cancer. `https://www.nice.org.uk/guidance/cg131`. Último acceso: Mayo 2015.

[246] Legaz-García MdC, Menárguez-Tortosa M, Fernández-Breis JT, Chute CG, Tao C. CEM to openEHR archetypes transformation (CEM2Archetypes). http://sele.inf.um.es/CEM2Archetypes. Último acceso: Mayo 2015.

[247] The Quest for Orthologs consortium. Quest for Orthologs. http://questfororthologs.org/. Último acceso: Mayo 2015.

[248] Irwin JJ, Sterling T, Mysinger MM, Bolstad ES, Coleman RG (2012) ZINC: A Free Tool to Discover Chemistry for Biology. Journal of Chemical Information and Modeling 52: 1757–1768.

[249] Guarino N (1999) The Role of Identity Conditions in Ontology Design. In: Freksa C, Mark DM, editors, Spatial Information Theory. Cognitive and Computational Foundations of Geographic Information Science, Springer Berlin Heidelberg, number 1661 in Lecture Notes in Computer Science. pp. 221–234.

[250] W3C. OWL 2 Web Ontology Language Structural Specification and Functional-Style Syntax. Keys. http://www.w3.org/TR/owl2-syntax/#Keys. Último acceso: Mayo 2015.

[251] Yu S, Berry D, Bisbal J (2012) Clinical coverage of an archetype repository over SNOMED-CT. Journal of Biomedical Informatics 45: 408–418.

[252] Qamar R, Rector A (2006) MoST: A system to semantically map clinical model data to SNOMED-CT. In: Proceedings of Semantic Mining Conference on SNOMED-CT. pp. 38–43.