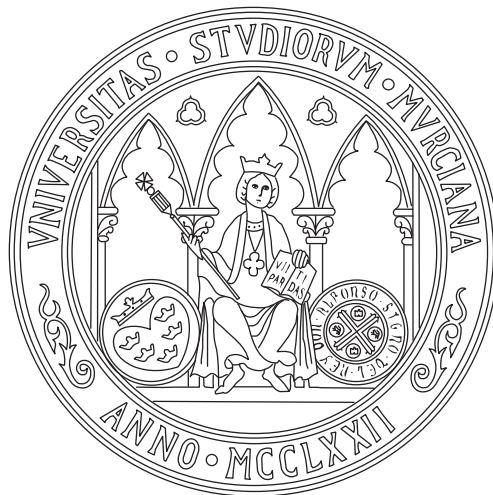


---

# Modelos de representación de arquetipos en sistemas de información sanitarios

---



**TESIS DOCTORAL**

**Marcos Menárguez Tortosa**

**Departamento de Informática y Sistemas**

**Facultad de Informática**

**Universidad de Murcia**

**Abril 2013**



# Modelos de representación de arquetipos en sistemas de información sanitarios

*Memoria que presenta para optar al título de Doctor en Informática*

**Marcos Menárguez Tortosa**

*Dirigida por el Doctor*

**Jesualdo Tomás Fernández Breis**

**Departamento de Informática y Sistemas**

**Facultad de Informática**

**Universidad de Murcia**

**Abril 2013**



# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Objetivos . . . . .	5
1.3. Estructura de la memoria . . . . .	6
<b>2. Contexto Tecnológico</b>	<b>9</b>
2.1. Historia Clínica Electrónica . . . . .	9
2.1.1. Interoperabilidad . . . . .	9
2.1.2. Estándares y especificaciones . . . . .	11
2.2. Terminologías clínicas . . . . .	16
2.3. Web Semántica . . . . .	19
2.4. Desarrollo de Software Dirigido por Modelos . . . . .	21
2.4.1. MDA . . . . .	21
2.4.2. Transformaciones de Modelos . . . . .	24
2.4.3. Relación con otros espacios tecnológicos . . . . .	24
<b>3. Arquitectura de Modelo Dual</b>	<b>27</b>
3.1. Introducción . . . . .	27
3.2. Lenguaje de Definición de Arquetipos . . . . .	29
3.3. Modelado de arquetipos . . . . .	33
3.4. Enlace terminológico . . . . .	35
3.5. Limitaciones del Modelo de Arquetipos . . . . .	38
3.6. Calidad en arquetipos . . . . .	39
3.7. Arquetipos en el desarrollo de aplicaciones sanitarias . . . . .	41
<b>4. Representación del conocimiento</b>	<b>45</b>
4.1. Introducción . . . . .	45
4.2. Ontologías . . . . .	46
4.2.1. Alineamiento de ontologías . . . . .	49
4.2.2. Ontologías en Ingeniería de Modelos . . . . .	50
4.3. Lógicas Descriptivas . . . . .	51

4.3.1.	Definiciones . . . . .	51
4.3.2.	Lógicas Descriptivas Básicas . . . . .	53
4.3.3.	Otras Lógicas Descriptivas . . . . .	54
4.3.4.	Razonamiento con lógicas descriptivas . . . . .	55
4.4.	OWL . . . . .	57
4.5.	Modelado ontológico en la arquitectura dual . . . . .	60
4.5.1.	Ontologías de los modelos de referencia . . . . .	60
4.5.2.	Ontologías aplicadas a modelos clínicos . . . . .	62
4.6.	Interoperabilidad de modelos clínicos . . . . .	65
<b>5.</b>	<b>Representación ontológica de arquetipos</b>	<b>69</b>
5.1.	Introducción . . . . .	69
5.2.	Marco de comparación . . . . .	70
5.3.	Poseacle Converter . . . . .	74
5.3.1.	Ontología <i>Poseacle</i> del Modelo de Arquetipos . . . . .	74
5.3.2.	Ontología del Modelo de Referencia . . . . .	74
5.3.3.	Arquetipos <i>Poseacle</i> . . . . .	77
5.3.4.	Restricciones sobre tipos primitivos . . . . .	81
5.4.	Archeck . . . . .	85
5.4.1.	Modelo de Referencia en <i>Archeck</i> . . . . .	85
5.4.2.	Modelo de arquetipos en <i>Archeck</i> . . . . .	87
5.5.	Encorsetable . . . . .	99
5.5.1.	Ontología de Restricciones . . . . .	99
5.5.2.	Representación del Modelo de Referencia . . . . .	100
5.5.3.	Representación de arquetipos con instancias . . . . .	103
5.5.4.	Representación de arquetipos como clases . . . . .	105
5.5.5.	Restricciones sobre tipos primitivos . . . . .	107
5.6.	Comparación de las representaciones . . . . .	109
<b>6.</b>	<b>Calidad de arquetipos</b>	<b>111</b>
6.1.	Introducción . . . . .	111
6.2.	Validación de las restricciones . . . . .	112
6.2.1.	Uso correcto de las entidades del modelo de referencia	113
6.2.2.	Redefinición de atributos . . . . .	116
6.2.3.	Referencias internas y slots . . . . .	124
6.3.	Detección de inconsistencias en especializaciones . . . . .	126
6.3.1.	Método de validación . . . . .	129
6.3.2.	Detección precisa de inconsistencias . . . . .	130
6.3.3.	Control de conceptos opcionales . . . . .	134
6.3.4.	Tipos de inconsistencias en especializaciones . . . . .	137
6.4.	Métricas de calidad . . . . .	138

---

6.4.1.	Consistencia de enlaces terminológicos . . . . .	138
6.4.2.	Calidad de enlaces terminológicos . . . . .	140
6.5.	Validación de los repositorios . . . . .	144
6.6.	Herramienta de validación . . . . .	146
<b>7.</b>	<b>Interoperabilidad de Modelos Clínicos</b>	<b>149</b>
7.1.	Introducción . . . . .	149
7.2.	Poseacle Converter . . . . .	150
7.2.1.	Ontología Común . . . . .	150
7.2.2.	Contexto tecnológico . . . . .	151
7.2.3.	Representación de arquetipos en OWL . . . . .	152
7.2.4.	Transformación de arquetipos . . . . .	154
7.2.5.	Limitaciones . . . . .	157
7.3.	Encorsetable . . . . .	158
7.3.1.	Introducción al proceso de transformación . . . . .	160
7.3.2.	Modelos de referencia con distinta expresividad . . .	164
7.3.3.	Arquetipos abstractos . . . . .	170
7.3.4.	Arquetipos abstractos como filtro de correspondencias	171
7.3.5.	Plantillas de transformación . . . . .	173
7.3.6.	Proceso de transformación . . . . .	176
7.3.7.	Framework <i>Encorsetable</i> . . . . .	178
<b>8.</b>	<b>Ingeniería de Modelos aplicada a la Arquitectura Dual</b>	<b>181</b>
8.1.	Introducción . . . . .	181
8.2.	Niveles de modelado en la arquitectura dual . . . . .	182
8.3.	Ontologías y modelos . . . . .	184
8.4.	Modelos de arquetipos semánticos . . . . .	186
8.5.	Semántica de visualización de arquetipos . . . . .	188
8.6.	ArchForms . . . . .	191
8.6.1.	Arquitectura . . . . .	192
8.6.2.	Evolución . . . . .	194
8.6.3.	Ejemplo de aplicación . . . . .	195
8.6.4.	Historia clínica normalizada . . . . .	199
8.6.5.	ArchForms en la arquitectura MDA . . . . .	200
<b>9.</b>	<b>Conclusiones y Trabajo Futuro</b>	<b>203</b>
9.1.	Resumen y conclusiones . . . . .	203
9.2.	Contribuciones . . . . .	213
9.3.	Trabajo futuro . . . . .	214
9.4.	Publicaciones . . . . .	216
	<b>Bibliografía</b>	<b>219</b>





# Índice de figuras

2.1. Fragmento del modelo de referencia de openEHR . . . . .	14
2.2. Arquitectura de la Web Semántica . . . . .	20
2.3. Arquitectura MDA . . . . .	22
2.4. Arquitectura de metamodelado del OMG . . . . .	23
3.1. Ejemplo de sección <i>cabecera</i> de un arquetipo . . . . .	29
3.2. Ejemplo de sección <i>definición</i> de un arquetipo . . . . .	30
3.3. Ejemplo de sección <i>ontología</i> de un arquetipo . . . . .	32
3.4. Modelo conceptual del concepto clínico <i>Examen</i> . . . . .	33
3.5. Esquema del arquetipo <i>Examen</i> en openEHR . . . . .	34
3.6. Mapa conceptual del concepto clínico <i>Examen del feto</i> . . . . .	35
3.7. Fragmento del modelo AOM del arquetipo <i>Examen</i> . . . . .	38
4.1. Relación entre los lenguajes y perfiles OWL . . . . .	58
5.1. Ejemplo de arquetipo . . . . .	71
5.2. Jerarquía <code>ITEM</code> del modelo de referencia de openEHR . . . . .	71
5.3. Extracto de la ontología del modelo de arquetipos en <i>Poseacle</i> . . . . .	75
5.4. Integración de ontologías en <i>Poseacle Converter</i> . . . . .	75
5.5. Jerarquía <code>ITEM</code> de openEHR en <i>Poseacle Converter</i> . . . . .	77
5.6. Representación en Archeck de la clase <code>CLUSTER</code> de openEHR . . . . .	87
5.7. Representación Archeck del concepto raíz del arquetipo . . . . .	89
5.8. Comparación axiomas de inclusión y equivalencia en OWL . . . . .	96
5.9. Axiomas OWL para validación cardinalidad en Archeck . . . . .	97
5.10. Definición completa del arquetipo en Archeck . . . . .	98
5.11. Ontología de Restricciones . . . . .	100
6.1. Fragmento del arquetipo <i>Examen</i> de openEHR . . . . .	113
6.2. Uso incorrecto del Modelo de Referencia . . . . .	114
6.3. Extracto del concepto raíz arquetipo <i>Examen</i> en OWL . . . . .	115
6.4. Propiedad de apoyo para detección de atributos no declarados . . . . .	116
6.5. Representación OWL del tipo de datos <code>CLUSTER</code> de openEHR . . . . .	117

6.6. Representación OWL del concepto raíz del arquetipo <i>Examen</i>	118
6.7. Restricción de rango en OWL	120
6.8. Validación conjunta de la restricción de ocurrencias	122
6.9. Representación extendida para la validación de ocurrencias	123
6.10. Representación OWL de una referencia interna	125
6.11. Representación OWL de un slot	125
6.12. Asignación de identificadores al arquetipo <i>Examen del Feto</i>	127
6.13. Representación de la relación de especialización	128
6.14. Ejemplo de propagación de inconsistencias	130
6.15. Restricción de orden en OWL	131
6.16. Detección precisa de la restricción de ocurrencias	132
6.17. Solución propagación de restricciones	133
6.18. Fases del proceso de validación de la relación de especialización	134
6.19. Ejemplo de inconsistencia provocada por conceptos opcionales	135
6.20. Clases de apoyo para el tratamiento de conceptos opcionales	136
6.21. Solución al problema de los conceptos opcionales	137
6.22. Ejemplo de enlace terminológico en SNOMED-CT	139
6.23. Representación de un enlace terminológico	140
6.24. Interfaz web de la herramienta de validación <i>Archeck</i>	147
7.1. Marco ontológico de la arquitectura <i>Poseacle</i>	151
7.2. Proceso de conversión de un arquetipo en OWL	154
7.3. Proceso de transformación de arquetipos	157
7.4. Arquetipo a transformar con la metodología <i>Encorsetable</i>	161
7.5. Correspondencias entre ISO EN 13606 y openEHR	162
7.6. Representación como instancias OWL del arquetipo	162
7.7. Inferencias obtenidas tras la clasificación del arquetipo	163
7.8. Resultado de la transformación del arquetipo	164
7.9. Estructura de datos <i>ITEM_TABLE</i> de openEHR	165
7.10. Arquetipo <i>tabla</i> en ISO EN 13606	165
7.11. Representación como clases del arquetipo <i>tabla</i>	166
7.12. Correspondencias entre <i>ITEM_TABLE</i> y el arquetipo <i>tabla</i>	167
7.13. Transformación de un arquetipo <i>ITEM_TABLE</i>	168
7.14. Fusión de arquetipos	169
7.15. Arquetipo abstracto <i>tabla</i> ISO EN 13606	170
7.16. Ejemplo de arquetipo <i>CLUSTER</i> ISO EN 13606	171
7.17. Transformación del arquetipo mediante un arquetipo abstracto	172
7.18. Arquetipo <i>plantilla</i> <i>OBSERVATION</i> de openEHR	174
7.19. Arquetipo <i>ENTRY</i> que caracteriza a una entidad observable	175
7.20. Correspondencias de una plantilla de transformación	176
7.21. Proceso de transformación <i>Encorsetable</i>	177

---

7.22. Framework <i>Encorsetable</i> . . . . .	180
8.1. Niveles de modelado en la arquitectura dual . . . . .	182
8.2. Arquitectura dual en la propuesta de modelado del OMG . .	183
8.3. Relación entre arquetipos y el modelo de referencia . . . . .	184
8.4. Arquitectura de modelado ontológico en ODM . . . . .	186
8.5. Arquitectura de modelado ontológico en <i>Poseacle Converter</i> .	187
8.6. Esquema de visualización del tipo de datos entero . . . . .	190
8.7. Proceso de generación de aplicaciones en ArchForms . . . . .	192
8.8. Módulos generadores de ArchForms . . . . .	193
8.9. Lógica de generación del transformador XForms2View . . . .	193
8.10. Extracto arquetipo <i>Revisión obstetricia</i> . . . . .	196
8.11. Controles XForms que capturan la frecuencia cardíaca del feto	197
8.12. Enlace que declara las restricciones de un control en XForms	198
8.13. Formulario de la aplicación Revisión obstetricia . . . . .	198
8.14. Aplicación Revisión obstetricia en iPhone . . . . .	199
8.15. Gestor de extractos de la aplicación Revisión Obstetricia . .	199
8.16. Extracto HCE de la aplicación Revisión Obstetricia . . . . .	200



# Índice de Tablas

4.1. Constructores de la lógica descriptiva $\mathcal{AL}$ . . . . .	53
4.2. Extensión $\mathcal{ALC}$ de la lógica descriptiva $\mathcal{AL}$ . . . . .	53
4.3. Constructores adicionales a lógica descriptiva $\mathcal{ALC}$ . . . . .	54
4.4. Axiomas adicionales a lógica descriptiva $\mathcal{ALC}$ . . . . .	55
4.5. Ejemplos de construcciones en notación OWL Manchester . . . . .	60
6.1. Errores de modelado en los repositorios CKM y NHS . . . . .	146



# Acrónimos

**ADL** Archetype Definition Language

**AML** Abstract Mapping Language

**ANSI** American National Standards Institute

**AOM** Archetype Object Model

**BFO** Basic Formal Ontology

**BNF** Backus Naur Form

**CDA** Clinical Document Architecture

**CEN** Comité Europeo de Normalización

**CEM** Clinical Element Model

**CIM** Computation Independent Model

**CIMI** Clinical Information Modeling Initiative

**CKM** Clinical Knowledge Manager

**D-MIM** Domain Message Information Model

**DL** Description Logics

**DSDM** Desarrollo de Software Dirigido por Modelos

**DSL** Domain Specific Language

**EDOAL** Expressive and Declarative Ontology Alignment Language

**EN** European Standard

**EMF** Eclipse Modeling Framework

**GWT** Google Web Toolkit

**HCE** Historia Clínica Electrónica

**HL7** Health Level Seven

**IHE** Integrating the Healthcare Enterprise

**IHSTDO** International Health Terminology Standards Development Organisation

**JSF** Java Server Faces

**IRI** Internationalized Resource Identifier

**ISO** Internacional Organization for Standarization

**LOINC** Logical Observation Identifiers Names and Codes

**MDA** Model Driven Architecture

**MOF** Meta-Object Facility

**NHS** National Health Service

**ODM** Ontology Definition Metamodel

**OMG** Object Management Group

**OWL** Web Ontology Language

**PIM** Platform Independent Model

**PSM** Platform Specific Model

**QVT** Query/View/Transformation

**R-MIM** Refined Message Information Model

**RDF** Resource Description Framework

**RDFS** Resource Description Framework Schema

**RIF** Rule Interchange Format

**RIM** Reference Information Model

**SNOMED-CT** Systematized Nomenclature of Medicine - Clinical Terms

**SPARQL** SPARQL Protocol and RDF Query Language

**UCUM** Unified Code for Units of Measure

**UMLS** Unified Medical Language System

**URI** Uniform Resource Identifier



**W3C** World Wide Consortium

**XMI** XML Metadata Interchange

**XML** Extensible Markup Language

**XUL** XML User Interface Language



# Capítulo 1

## Introducción

### 1.1. Motivación

La historia clínica electrónica (HCE) es el conjunto de documentos relativos a la salud de un paciente registrados en formato electrónico. Actualmente la HCE es considerada una pieza clave para la prestación eficiente de los servicios sanitarios y, en especial, para garantizar la calidad de la atención sanitaria y la seguridad del paciente. Asimismo, la información clínica almacenada en la HCE contribuye a la investigación médica, la formación de los profesionales sanitarios y la salud pública [183].

Los organismos de estandarización en el ámbito de la Informática Médica han propuesto normas que establecen los requisitos de información y organización de los documentos en la HCE. En 2008, ISO estandarizó parte de la norma europea EN 13606 [96] que ha sido fruto de más de dos décadas de proyectos de investigación y desarrollo [90, 67, 8]. El estándar ISO EN 13606 consolida la *arquitectura de modelo dual* que propone organizar la HCE en dos niveles conceptuales: modelo de referencia y modelo de arquetipos. El modelo de referencia define el conjunto de entidades que forman los bloques de construcción genéricos de la HCE, como documentos o carpetas. Por otro lado, el modelo de arquetipos especifica cómo representar los conceptos clínicos que han de registrarse en la HCE a partir de combinaciones válidas de las entidades del modelo de referencia dando lugar a los denominados *arquetipos*. Por lo tanto, el modelo de referencia define las características estables de la HCE y los arquetipos representan el nivel de conocimiento. En esta arquitectura de HCE, se definen mediante arquetipos conceptos sanitarios como la medida de la presión sanguínea o un informe de alta. La arquitectura de modelo dual ha sido promovida en la última década por la Fundación openEHR y ha influido en otros estándares como el modelo de plantillas de HL7 CDA [44].

El informe final del proyecto europeo SemanticHEALTH estableció 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 [183]. Por una parte, el uso de un modelo de referencia común permite intercambiar un extracto de HCE sin necesidad de acordar de antemano el contenido clínico. Por otra parte, el uso de arquetipos como especificaciones de modelos clínicos y la representación de la información clínica conforme a ellos permite interpretar el significado del contenido de la HCE. De este modo se alcanza la interoperabilidad semántica completa que permite que cualquier extracto de HCE pueda ser importado y combinado con los datos locales de un sistema de información sanitario.

Recientemente se ha constituido la iniciativa internacional CIMI (Clinical Information Modeling Initiative) con el objetivo de promover un formato común para el intercambio de modelos de contenido clínico que pueda ser implementable en cualquier sistema de información sanitario [31]. Inicialmente se ha acordado utilizar la especificación de arquetipos de ISO EN 13606 y openEHR como formato de representación de los modelos clínicos. Por lo tanto, el modelado con arquetipos está ganando aceptación como el formalismo mejor soportado para representar e intercambiar los modelos de contenido clínico.

Los arquetipos han sido diseñados para tener un papel clave en el funcionamiento de los sistemas de información sanitarios. Los arquetipos influyen en cómo interaccionan las aplicaciones sanitarias con los usuarios, más concretamente, guiando la introducción y presentación de la información clínica. Por consiguiente, los arquetipos deben ser considerados un activo de conocimiento que ha de ser incorporado en el diseño de las aplicaciones clínicas, al igual que otros recursos de conocimiento en el ámbito sanitario como los sistemas terminológicos. Sin embargo, para que los arquetipos sean aceptados y adoptados ampliamente en los sistemas de información sanitarios deben tener una calidad demostrable [103].

El instituto EuroRec ha liderado el proyecto Q-REC en el que se han estudiado criterios de calidad para arquetipos [104]. El estudio destaca la necesidad de métodos formales para la validación del diseño y contenido de los arquetipos. Por una parte, el lenguaje para especificar arquetipos (ADL) ha sido diseñado de forma genérica e independiente de cualquier modelo de referencia. La validación de la conformidad de las definiciones de los arquetipos respecto al modelo de referencia se deja a cargo de las herramientas de edición. Actualmente son muy pocas las herramientas que ofrecen algún soporte para el análisis semántico de los arquetipos. El caso más destacado es el editor LinkEHR-Ed [123] que define formalmente la semántica del modelo de arquetipos en base a un sistema de tipos propio. Por otro lado, varias recomendaciones internacionales están promoviendo el uso de terminologías en la especificación de arquetipos [183, 31], especialmente la terminología SNOMED-CT [89].

Recientemente se ha publicado un estudio sobre calidad de modelos clínicos en el que se ha propuesto un conjunto de métricas para evaluar la calidad de los modelos [2]. Una parte de las métricas definidas en este trabajo se evalúan positivamente en arquetipos que sean correctos técnicamente. Además, a través de varias métricas se destaca la importancia del uso consistente de las terminologías clínicas en la definición de arquetipos.

Dado que los arquetipos son considerados una técnica de modelado de conocimiento clínico en el dominio HCE, en los últimos años ha habido un creciente interés en explorar cómo las tecnologías semánticas en general, y las ontologías en particular, pueden facilitar la representación y gestión de arquetipos. La relevancia de los enlaces con terminologías en arquetipos y los esfuerzos en curso para proporcionar un mejor enlace entre las terminologías y las ontologías [173] refuerzan esta idea. OWL es una familia de lenguajes para la definición de ontologías promovida por el World Wide Web Consortium (W3C) [12], y que permite integrar arquetipos, modelos de información y terminologías en un marco de modelado único. Esto es especialmente relevante porque los modelos de información tratan con estructuras de datos HCE y las terminologías con los modelos de significado [162].

El sublenguaje OWL-DL se basa en el formalismo de lógica descriptiva [6]. Esto facilita la aplicación de técnicas de razonamiento basadas en lógica descriptiva, lo cual es de gran interés para la resolución de problemas relacionados con interoperabilidad semántica y el enlace entre arquetipos, terminologías y ontologías.

La especificación de arquetipos en OWL ha sido objeto de investigación en los últimos años. En el proyecto europeo Artemis se realizó la primera aproximación con objeto de establecer correspondencias entre modelos clínicos de distintos estándares [15]. Esta propuesta fue extendida en [118] para la representación semántica de reglas clínicas. Ambas representaciones adoptan un enfoque basado en clases OWL, que contrasta con la propuesta *Poseacle* que define un marco ontológico para las arquitecturas de HCE de modelo dual en el que los arquetipos son especificados como instancias en OWL [59].

Los arquetipos no han sido la única tecnología de HCE que ha aprovechado OWL y las ontologías. OWL ha sido utilizado con diferentes fines en los estándares HL7. La versión 3 de este estándar ha sido definida formalmente con ontologías en [191, 155, 26, 91] y se han empleado ontologías como soporte para la interoperabilidad de las versiones 2 y 3 del estándar HL7 en [15, 18, 167]. Además, la representación OWL de la terminología SNOMED-CT y del modelo de documentos clínicos CDA ha permitido validar la consistencia de los enlaces terminológicos en CDA [74]. Por último, las tecnologías semánticas han sido aplicadas con éxito al modelado de documentos clínicos fuera del ámbito de los estándares de HCE. Por ejemplo,

en [185] se ha presentado una propuesta para especificar documentos CEM con el lenguaje OWL.

El objetivo establecido por la iniciativa CIMI de disponer de modelos clínicos interoperables apenas ha sido estudiado hasta el momento. La mayor parte de los trabajos se han enfocado a la definición de correspondencias entre los modelos, obtenidas mediante técnicas semiautomáticas, como soporte para la transformación de datos clínicos [16, 167]. En [13] el alineamiento de los modelos clínicos se consigue a través del uso de terminologías clínicas y del lenguaje OWL, que se emplea también para establecer e inferir correspondencias. Finalmente, la herramienta Poseacle Converter define e implementa una metodología de transformación de arquetipos para los estándares ISO EN 13606 y openEHR [128].

Desde el punto de vista de los sistemas de información, la arquitectura de modelo dual pretende dar una respuesta a los problemas clásicos de evolución y mantenimiento de los sistemas, especialmente en dominios complejos como la medicina donde los cambios son frecuentes [9]. La arquitectura de modelo dual permite desarrollar los sistemas de información sin necesidad de que los conceptos del dominio estén definidos, ya que éstos son independientes del software y pueden ser introducidos cuando el sistema ya esté implantado. Así pues, la adopción de esta arquitectura en el ámbito sanitario ofrece a los expertos del dominio un modo de especificar los modelos de contenido de los conceptos clínicos sin necesidad de conocimientos técnicos. Por lo tanto, los sistemas de información pueden adaptarse a los cambios en las prácticas médicas y a la prestación de servicios sanitarios a lo largo del tiempo [63].

En los últimos años se ha investigado la aplicación de arquetipos en el desarrollo de aplicaciones sanitarias. En [172] se demostró que es factible obtener automáticamente interfaces gráficas de usuario a partir de arquetipos. Las ideas de este trabajo han sido aplicadas a varios generadores de aplicaciones sanitarias como EHRFlex [24], EHRGen [157] o ZK-ARCHE [112]. En el contexto del proyecto Opereffa [153] se ha realizado un estudio sobre los requisitos arquitectónicos que imponen los arquetipos al diseño de interfaces gráficas de usuario. El trabajo señala que actualmente hay disponibles tecnologías que ofrecen soluciones a los retos que plantea la arquitectura dual, aunque la conclusión principal del trabajo es que el desafío arquitectónico más importante en las aplicaciones sanitarias basadas en arquetipos es la evolución tecnológica. Por último, en [119] se aboga por la definición abstracta de interfaces de usuario independientes de la plataforma tecnológica y por la generación automática de interfaces para tecnologías específicas.

El Desarrollo de Software Dirigido por Modelos (DSDM) es una corriente de la Ingeniería del Software que promueve un papel más activo de los modelos en todas las etapas del ciclo de vida del software, permitiendo la generación automática de las aplicaciones y reduciendo el coste de mante-

nimiento de las aplicaciones [180]. MDA (Model Driven Architecture) es la principal iniciativa dentro de DSDM que destaca por estar liderada por el consorcio OMG, por el uso de estándares y por la definición de una arquitectura de modelado aceptada en la comunidad DSDM [146]. Además, la especificación ODM (Ontology Definition Metamodel) [149] del consorcio OMG establece las bases para la integración los lenguajes UML y OWL.

## 1.2. Objetivos

El trabajo de esta tesis doctoral se desarrolla en torno a los siguientes objetivos:

- Propuesta de una representación ontológica como soporte para evaluar la calidad de arquetipos. Creemos que una representación de la arquitectura dual de HCE basada en ontologías permitiría la validación de arquetipos y la definición de métricas que evalúen su calidad. En primer lugar, los modelos ontológicos podrían ser utilizados para una representación del conocimiento clínico, y esto facilitaría el desarrollo de métodos eficientes para la gestión del conocimiento. En segundo lugar, la combinación de modelos ontológicos avanzados, tales como OWL, con técnicas de razonamiento podría indudablemente reducir el esfuerzo necesario para implementar métodos de validación y de evaluación de la calidad.
- Definición de una metodología de interoperabilidad para modelos clínicos. El intercambio de modelos clínicos entre comunidades de usuarios es necesario para la reutilización del conocimiento de la representación de HCE. Los modelos clínicos, como los arquetipos, son activos de conocimiento que pueden ser empleados en el desarrollo de sistemas de información. Así pues, nuestra propuesta es analizar las posibilidades de OWL como lenguaje canónico para representar modelos clínicos y articular en torno a esta tecnología un marco de trabajo que formalice el proceso de transformación.
- Definición de un marco de desarrollo de aplicaciones sanitarias basadas en estándares. Los arquetipos pueden ser considerados modelos de dominio que pueden ser utilizados para guiar la generación de aplicaciones. La integración de la especificación de los estándares de HCE de modelo dual en una arquitectura de modelado como la propuesta por el OMG sería de utilidad para aprovechar las técnicas y herramientas disponibles en la comunidad DSDM. Además, creemos que las ontologías podrían tener un papel determinante en esta tarea. Por consiguiente, creemos que la obtención automática de aplicaciones basadas en estándares favorecerá la interoperabilidad de la HCE.

### **1.3. Estructura de la memoria**

#### **Capítulo 2. Contexto tecnológico**

En este capítulo presentamos el entorno tecnológico en el que se desarrolla el trabajo. En primer lugar, se introduce el concepto de Historia Clínica Electrónica y los estándares que promueven su interoperabilidad, especialmente aquellos que están basados en la arquitectura de modelo dual. Después se presenta el concepto de terminología clínica y su papel para lograr la interoperabilidad de la HCE. A continuación se motiva la importancia de las tecnologías semánticas, y en concreto, la iniciativa de la Web Semántica como soporte para la integración e interoperabilidad de la información. Por último, estudiamos el Desarrollo de Software dirigido por Modelos que promueve el uso de los modelos en todas las etapas del proceso software.

#### **Capítulo 3. Arquitectura de Modelo Dual**

En este capítulo desarrollamos la arquitectura de modelo dual propuesta por la Fundación openEHR y estandarizada en ISO EN 13606. Comenzamos con la motivación de la arquitectura dual y definiendo el concepto de arquetipo. Seguidamente se presenta el lenguaje de definición de arquetipos y se introduce informalmente el modelado de un concepto clínico mediante un arquetipo. Después se revisan varias propuestas que tratan de automatizar el proceso de enlace entre arquetipos y terminologías clínicas. A continuación se analizan las limitaciones del modelo de arquetipos y se motiva la necesidad de guías y procesos que aseguren la calidad de arquetipos. Finalmente, resumimos varios trabajos que han estudiado la aplicación de arquetipos al proceso de desarrollo de sistemas de información sanitarios.

#### **Capítulo 4. Representación del conocimiento**

En este capítulo presentamos las ontologías como modelos de representación del conocimiento y sus aplicaciones a la arquitectura dual. En primer lugar, introducimos el concepto de ontología, tipos y los procesos de construcción de ontologías. Seguidamente, nos centramos en el alineamiento de ontologías y en la relación con el Desarrollo de Software Dirigido por Modelos. Más tarde se presenta el lenguaje de modelado ontológico OWL y el formalismo de la Lógica Descriptiva en el que se apoya. Terminamos el capítulo revisando los principales trabajos que han estudiado la representación ontológica de la arquitectura dual y la interoperabilidad de modelos clínicos basada en ontologías.



## Capítulo 5. Representación ontológica de arquetipos

Este capítulo presenta varias representaciones de arquetipos en OWL que han sido utilizadas en distintas tareas de este trabajo. Comenzamos con el marco ontológico *Poseacle* que ha sido empleado como soporte para la transformación de arquetipos entre los estándares ISO EN 13606 y openEHR. A continuación describimos la representación de arquetipos propuesta por el método *Archeck* para validación de la corrección y consistencia de arquetipos. Por último, presentamos una representación en OWL de la arquitectura dual que ha sido aplicada a la interoperabilidad de modelos clínicos.

## Capítulo 6. Calidad de arquetipos

Este capítulo describe principalmente el método *Archeck* de validación de arquetipos. El capítulo comienza analizando la validación de las restricciones de un arquetipo respecto al modelo de referencia. Seguidamente estudiamos los requisitos de validación de la relación de especialización entre arquetipos. Después se propone un marco de definición de métricas de calidad de arquetipos, orientado especialmente a la evaluación de los enlaces terminológicos. Finalmente, presentamos los resultados del análisis de dos repositorios de arquetipos y la herramienta web que implementa el método *Archeck*.

## Capítulo 7. Interoperabilidad de Modelos Clínicos

En este capítulo estudiamos dos propuestas de transformación de modelos clínicos entre estándares de HCE. En primer lugar, presentamos la herramienta *Poseacle Converter* que define y implementa un método de transformación de arquetipos en arquitecturas de modelo dual. A continuación analizamos las principales limitaciones de *Poseacle Converter* y proponemos el marco de trabajo *Encorsetable* que da una respuesta a los problemas de *Poseacle Converter* y utiliza la tecnología OWL en todas las etapas de proceso de transformación de modelos clínicos.

## Capítulo 8. Ingeniería de Modelos aplicada a la Arquitectura Dual

Este capítulo analiza los beneficios de la aplicación de las técnicas y herramientas del Desarrollo de Software Dirigido por Modelos a las arquitecturas de HCE basadas en el modelo dual. En primer lugar, motivamos el uso de las ontologías, y en concreto, el concepto de modelo de arquetipo semántico como medio para representar la arquitectura dual en MDA. En segundo lugar, presentamos la herramienta ArchForms, que es un generador automático de aplicaciones sanitarias conforme al estándar ISO EN 13606.

**Capítulo 9. Conclusiones y Trabajo Futuro**

En el último capítulo resumimos el trabajo realizado, presentamos las conclusiones más destacadas, y proponemos posibles líneas de trabajo futuro.

## Capítulo 2

# Contexto Tecnológico

### 2.1. Historia Clínica Electrónica

La historia clínica electrónica (HCE) es el conjunto de documentos relativos a la salud de un paciente registrados en formato electrónico. Actualmente la HCE es considerada una pieza clave para la prestación eficiente de los servicios sanitarios y, en especial, para garantizar la calidad de la atención sanitaria y la seguridad del paciente. Asimismo, la información clínica almacenada en la HCE contribuye a la investigación médica, la formación de los profesionales sanitarios y a la salud pública [183].

La investigación llevada a cabo durante las últimas dos décadas ha sacado a la luz los requisitos clínicos, éticos y técnicos de los sistemas de HCE. Estos requisitos han sido consolidados por ISO en una especificación técnica que proporciona un punto de referencia para el desarrollo de la HCE [98]. En general, los retos más importantes a los que se enfrentan las tecnologías de la información y las comunicaciones para implantar con éxito la HCE son ofrecer confianza en la información compartida y garantizar que la información clínica es interpretada de forma correcta y segura. Con este fin, los sistemas sanitarios deben *interoperar a nivel semántico*, de manera que un sistema pueda entender el contexto y significado de la información proporcionada por otro.

#### 2.1.1. Interoperabilidad

En la década de los 90 arrancó la investigación en los aspectos técnicos de la interoperabilidad de la HCE a través de varios proyectos de investigación (GEHR [90], SYNAPSES [67]). Algunos de los resultados obtenidos de estos proyectos han sido trasladados a normas respaldadas por organismos de estandarización como el CEN (Comité Europeo de Estandarización), ISO y HL7. El resultado más destacado ha sido la norma ISO EN 13606 sobre la representación y comunicación de la HCE [96].

El estándar ISO EN 13606 propone la *arquitectura de modelo dual* que consiste en organizar la HCE en dos niveles conceptuales: modelo de referencia y modelo de arquetipos. El modelo de referencia define el conjunto de entidades que forman los bloques de construcción genéricos de la HCE, como documentos o carpetas. Por otro lado, el modelo de arquetipos especifica cómo representar los conceptos clínicos que han de registrarse en la HCE en base a combinaciones válidas de las entidades del modelo de referencia dando lugar a los denominados *arquetipos*. La semántica clínica de las estructuras de datos se consigue enlazando las estructuras de datos definidas por los arquetipos con sistemas terminológicos como SNOMED-CT [89]. La arquitectura de modelo dual ha sido promovida en la última década por la Fundación openEHR [152] y ha influido en otros estándares como el modelo de plantillas de CDA de HL7 [21].

La arquitectura de modelo dual aborda la interoperabilidad de la HCE desde dos perspectivas. En primer lugar, el uso de un modelo de información común y limitado solamente a los requisitos médico-legales y a las funciones de gestión de los historiales permite que un extracto de HCE sea interpretado fielmente en un sistema receptor incluso si el contenido clínico no ha sido acordado de antemano, es decir, se logra interoperabilidad a nivel funcional. Por otro lado, el uso de arquetipos y la construcción de extractos de HCE conforme a su definición permite que el sistema receptor pueda interpretar el significado de la información recibida.

En los últimos años varios estudios financiados por la Comisión Europea se han centrado en la interoperabilidad de la HCE en un sentido más amplio. El proyecto RIDE [43] definió una primera hoja de ruta, en EHR IMPACT [41] se han analizado los beneficios de los sistemas HCE interoperables y en SemanticHEALTH [183] se ha definido una hoja de ruta para la investigación y desarrollo de soluciones factibles en el corto y medio plazo para lograr la interoperabilidad de la HCE.

El informe final del proyecto SemanticHEALTH define varios niveles de interoperabilidad entre sistemas de HCE. El nivel más básico se consigue cuando la comunicación de la HCE se realiza a nivel de documentos, donde las consultas están limitadas poco más que a las fechas de registro, al proveedor de la información o al tipo de documento. Este nivel es denominado *interoperabilidad sintáctica* y es suficiente cuando el requisito de información es que los documentos deben ser legibles para los profesionales sanitarios. Si la HCE se comunica de forma estructurada, entonces podemos alcanzar la *interoperabilidad semántica*. Cuando los sistemas participantes deben hacer el esfuerzo de adaptación de la información a sus repositorios locales, entonces la interoperabilidad semántica es parcial. La *interoperabilidad semántica completa* se consigue cuando 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. Por

lo tanto, el objetivo de la interoperabilidad semántica es poder reconocer y procesar información semánticamente equivalente de manera homogénea, incluso si las instancias están representadas de forma heterogénea, es decir, si ellas están estructuradas de modo distinto, utilizan distintos sistemas terminológicos o están expresados en idiomas diferentes. Para alcanzar la interoperabilidad semántica completa el informe 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.

Sin embargo, la existencia de varios estándares para representar la HCE dificulta la comunicación y el intercambio de información entre sistemas sanitarios. Las principales organizaciones de estandarización en Informática Médica están trabajando en la armonización de sus normas. Un ejemplo de ello es la parte 3 del estándar ISO EN 13606 que recoge una guía sobre cómo representar mediante arquetipos información clínica de HL7 versión 3 o la propuesta de la Fundación openEHR para representar su modelo de referencia en ISO EN 13606 [188]. Recientemente se ha aprobado la norma ISO 21090 [99] que tiene como objetivo definir un conjunto común de tipos de datos en la HCE.

En este sentido, la iniciativa CIMI (Clinical Information Modeling Initiative) nace con el propósito de promover un formato común para la interoperabilidad de los modelos de información clínica [31]. Los integrantes de esta iniciativa son organismos de estandarización e instituciones en el ámbito de la Informática Médica motivados por la necesidad de proporcionar modelos clínicos que puedan ser implementados en cualquier sistema de información sanitario. Como punto de partida se ha propuesto la arquitectura de modelo dual, la especificación de arquetipos de la norma ISO EN 13606 parte 2, el lenguaje UML para definir los modelos de referencia y SNOMED-CT como terminología clínica. No obstante, el objetivo es que los modelos clínicos puedan ser representados en los distintos formatos que actualmente soportan los sistemas de HCE.

### 2.1.2. Estándares y especificaciones

El comité técnico TC 215 de ISO es responsable de la estandarización en el área de Informática Médica. El objetivo de las normas que publica es conseguir la compatibilidad e interoperabilidad entre sistemas de información sanitarios independientes. El trabajo de este comité tiene como base los resultados de otros organismos de normalización como HL7 o CEN. En el dominio de la HCE, ISO ha publicado el estándar de comunicación más relevante actualmente a nivel mundial, ISO EN 13606, así como otras normas que brevemente se describen a continuación. En el resto del apartado se describen otros estándares y especificaciones destacados en el ámbito de la HCE.

- ISO/TR 20514: Definición, alcance y contexto de la HCE [95]. El propósito de esta norma es definir de forma genérica la estructura de una HCE básica con el fin de que sirva de base para un amplio rango de sistemas de HCE presentes y futuros. Además, establece los requisitos médico-legales que son aplicables a cualquier HCE. Los modelos de referencia de openEHR y ISO EN 13606 son conformes con esta especificación.
- ISO/TR 18307: Interoperabilidad y compatibilidad en estándares de mensajería y comunicación [92]. Este estándar describe los principales requisitos para lograr la interoperabilidad y compatibilidad en el intercambio de información sanitaria que sirven como criterio para los desarrolladores e implementadores de estándares de mensajería y comunicación en el dominio sanitario.
- ISO 18308: Requisitos para una arquitectura de referencia para HCE [98]. Proporciona los requisitos arquitectónicos de los sistemas HCE. La especificación está destinada a organismos que trabajan en la estandarización de la HCE, como por ejemplo CEN.

#### 2.1.2.1. ISO EN 13606

El comité técnico TC 251 del CEN está encargado del desarrollo de estándares en Informática Médica en la Unión Europea. El resultado más notable ha sido la norma ISO EN 13606, que es el único estándar completo sobre interoperabilidad HCE a nivel internacional. El objetivo de esta norma es la definición de una arquitectura de información rigurosa y estable para la comunicación de la HCE, que destaca por la aplicación de la arquitectura de modelo dual. Asimismo, la especificación de la norma ha sido desarrollada para preservar el significado clínico establecido por el autor del registro y la confidencialidad de los datos tal como es establecida por el autor y el paciente.

El estándar ISO EN 13606 tiene cinco partes. Las dos primeras partes corresponden a los dos niveles de la arquitectura dual, el modelo de referencia (parte 1) y la especificación para el intercambio de arquetipos (parte 2). Esta última define los requisitos técnicos que han de cumplir los arquetipos, así como la sintaxis y semántica del lenguaje para definición de arquetipos (ADL). La parte 3 incluye guías sobre el uso del modelo de referencia y el diseño de arquetipos e incluye un vocabulario inicial de términos que pueden ser empleados al instanciar el modelo de referencia. Además, la parte 3 define un conjunto de arquetipos que ilustran cómo representar la información clínica definida en otros estándares como HL7. Las partes 4 y 5 tratan la seguridad en el acceso a la HCE y la especificación de interfaces de servicio y mensajes para la comunicación de HCE.

Hoy en día las herramientas disponibles para este estándar son escasas. Entre ellas, la plataforma que más ha contribuido a la difusión e implantación del estándar es *LinkEHR* [87], que está formado por un editor de arquetipos (*LinkEHR-Ed*), una herramienta de normalización de datos clínicos en sistemas heredados (*LinkEHR Studio*) y un visualizador web de HCE (*LinkEHR Viewer*), entre otras herramientas. Recientemente se ha constituido la asociación EN 13606 [51] con el propósito de promocionar y difundir el estándar.

El modelo de referencia de ISO EN 13606 se organiza en cuatro paquetes, siendo el paquete *Extracto* el que contiene las entidades de negocio que conforman la HCE:

- **Extracto (EHR\_EXTRACT):** representa una parte o toda la HCE de un paciente.
- **Carpeta (FOLDER):** entidad que organiza la HCE al primer nivel y que suele estar asociada a episodios clínicos o especialidades médicas.
- **Composición (COMPOSITION):** representa un documento de la HCE, como un informe o una prueba, que suele estar relacionados con una sesión clínica. Las composiciones se organizan en carpetas.
- **Sección (SECTION):** las entradas clínicas que forman un documento clínico (COMPOSITION) se organizan en secciones o encabezamientos que facilitan el registro y presentación de la información.
- **Entrada (ENTRY):** una entrada o declaración clínica es la unidad de registro de la información en una HCE. Puede corresponder, por ejemplo, con una observación, una petición de procedimiento o un diagnóstico.
- **Clúster (CLUSTER):** es una estructura de datos que permite organizar la información en series, tablas o árboles.
- **Elemento (ELEMENT):** en la organización de la HCE representa a los contenedores de valores y se encuentra al nivel más profundo de la jerarquía.

El modelo de referencia cumple con la norma ISO/TS 18308 que establece los requisitos de información de la HCE, de modo que es posible representar el contexto de la información (autor, fecha, etc.) y que esta información puede ser interpretada correctamente al ser intercambiada con otros sistemas de información sanitarios. Las entidades que representan esta información son, entre otras, ATTESTATION\_INFO, FUNCTIONAL\_ROLE O RELATED\_PARTY.

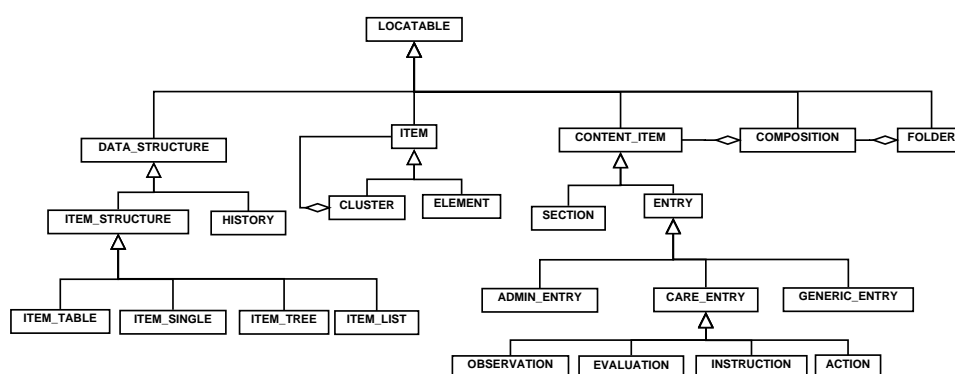


Figura 2.1: Fragmento del modelo de referencia de openEHR

### 2.1.2.2. openEHR

La Fundación openEHR es una organización sin ánimo de lucro cuyo fin es el desarrollo de especificaciones abiertas, software y recursos de conocimiento para sistemas de información sanitarios, y en concreto, sistemas de HCE [102]. La arquitectura de información propuesta en sus especificaciones es fiel al modelo dual y comparte con la norma ISO EN 13606 el modelo de arquetipos.

Las especificaciones openEHR no han sido creadas a través de un proceso formal de estandarización, sino que han sido el fruto de varios proyectos de investigación. Así pues, los trabajos están respaldados por los resultados de varios proyectos de investigación, principalmente, GEHR [90], y los proyectos que sucedieron a éste, Synapses [67] en Europa, y los proyectos australianos GEHR [8]. A pesar de no ser una organización de estandarización, los trabajos de openEHR han influido en normas internacionales, especialmente en la norma ISO EN 13606.

El modelo de referencia de openEHR es más amplio y detallado que el propuesto por ISO EN 13606 (véase figura 2.1). Así, por ejemplo, las clases del paquete *Extracto* de ISO EN 13606 también están incluidas y se añaden más tipos de entradas clínicas (observaciones, evaluaciones, etc.). Sin embargo, los tipos de datos fundamentales de openEHR son diferentes a otros estándares, lo que dificulta notablemente la interoperabilidad, aunque cabe esperar que este problema se resuelva con la introducción del conjunto de tipos de armonizado propuestos por ISO.

Las herramientas que respaldan las especificaciones de openEHR son numerosas y gran parte de ellas son de código abierto. La empresa Ocean Informatics también ofrece software para esta plataforma, como por ejemplo el editor de plantillas *Ocean Template Designer* [143]. Finalmente, la Fundación openEHR ha desarrollado un entorno colaborativo para la gestión y publicación de arquetipos denominado *Clinical Knowledge Manager (CKM)* [189].



### 2.1.2.3. HL7

HL7 (Health Level Seven) es una organización de estandarización perteneciente a ANSI que desarrolla estándares para el intercambio de datos clínicos y administrativos entre sistemas de información sanitarios. El término HL7 es utilizado tanto para referirse a la organización como al conjunto de estándares de mensajería. HL7 versión 2 es el estándar en Informática Médica más implementado en el mundo. Sin embargo, a pesar de su éxito, la versión 2 del estándar HL7 presenta ciertos problemas de interoperabilidad debidos fundamentalmente a la flexibilidad para la definición de mensajes que obliga a que las entidades participantes en la comunicación deban llegar a acuerdos sobre la interpretación de los mensajes [50].

HL7 versión 3 supera las limitaciones de la versión anterior proponiendo una metodología formal para la definición de los mensajes y la transición hacia la sintaxis XML. Todas las especificaciones de mensajes son definidas a partir de un modelo de información genérico conocido como RIM (Reference Information Model) [73]. Esta versión del estándar propone también un proceso para derivar los modelos de mensajes a partir de RIM. Asimismo, los modelos de información están diseñados para ser asociados con sistemas terminológicos y se han definido vocabularios controlados para algunos dominios.

Los modelos de mensajes en HL7 versión 3 son derivados a partir de RIM a través de un proceso de refinamiento hasta obtener una representación de mensajes implementable [75]. Para un dominio concreto, se obtiene un modelo D-MIM (Domain Message Information Model) definido a partir de las entidades del RIM. Seguidamente, se construye un modelo R-MIM (Refined Message Information Model) que contiene los elementos de información necesarios para definir uno o varios HMD (Hierarchical Message Description). Los modelos HDM son una representación tabular de una secuencia de elementos de un R-MIM independientes de cualquier implementación tecnológica. Por último, el estándar también define la especificación *XML Implementable Technology* para expresar los modelos HMD en XML.

HL7 CDA (Clinical Document Architecture) describe la estructura y semántica de los documentos clínicos intercambiados entre las aplicaciones sanitarias [21]. HL7 CDA tiene dos versiones, la *release 1* propuesta en 2000 y la *release 2* en 2005. Un documento CDA se especifica en XML y consta de dos partes, una cabecera y un cuerpo. La diferencia entre las dos *releases* de CDA es que en la 1 sólo la cabecera está estructurada como un modelo derivado de RIM, mientras que la *release 2* permite introducir entradas clínicas estructuradas en el cuerpo, aunque no es obligatorio. Los únicos campos obligatorios en el cuerpo de un documento CDA son los bloques de texto en lenguaje natural.

La especificación *templates* de HL7 ha sido propuesta para definir modelos de documentos HL7 CDA conforme a la arquitectura de modelo dual [22]. Así pues, los *templates* tienen un rol equivalente a los arquetipos ISO EN 13606 y openEHR, es decir, permiten describir la estructura del contenido para documentar conceptos del dominio clínico.

#### 2.1.2.4. IHE XDS

IHE (Integrating the Healthcare Enterprise) es una organización cuyo objetivo es promover la interoperabilidad entre empresas dedicadas al ámbito sanitario [181]. La especificación XDS (Cross Enterprise Document Sharing) define servicios de registro y repositorio que pueden funcionar como almacén centralizado o distribuido de documentos clínicos.

La especificación IHE XDS permite a varias instituciones sanitarias el intercambio de documentos clínicos mediante el acuerdo sobre un conjunto de políticas en común como la identificación de los pacientes, el formato de los documentos y los vocabularios clínicos para representar los metadatos de los documentos. Sin embargo, no establece ninguna restricción sobre el tipo de documento, de modo que pueden utilizarse desde documentos en formato PDF hasta documentos clínicos HL7 CDA o extractos ISO EN 13606, aunque no se comparte una HCE completa. En general, esta propuesta es adecuada para la integración de sistemas de información sanitarios a pequeña escala.

## 2.2. Terminologías clínicas

Las terminologías clínicas tienen un papel fundamental en la consecución de la interoperabilidad semántica de la HCE [183]. Una terminología puede definirse como un conjunto de términos que son utilizados en un contexto concreto con el fin de favorecer la comunicación entre las entidades. Así pues, una terminología clínica podría definirse de forma sencilla como un conjunto de términos empleados en el dominio sanitario.

Sin embargo, el vocabulario utilizado para describir terminologías, ontologías y sistemas de clasificación ha sido una fuente de confusión, dado que distintos autores han usado las mismas palabras de forma diferente [161]. A continuación se aportan varias definiciones que precisan el significado de una terminología:

- Vocabulario controlado: lista de elementos que son usados con algún propósito específico y que se emplea habitualmente en los sistemas de información para reducir la ambigüedad, evitar errores ortográficos, etc.

- Sistema de identificadores o códigos: se emplean para hacer referencia a las entidades de forma no ambigua. Muchos vocabularios controlados, léxicos, ontologías y tesauros están habitualmente acompañados de un sistema de identificadores.
- Léxico: es una lista de unidades lingüísticas en uno o varios idiomas que pueden ser adjuntadas a un vocabulario controlado o a una ontología, como por ejemplo, para establecer términos preferentes o sinónimos.
- Ontología: descripción formal y explícita de los conceptos de un dominio, las propiedades que describen las características y atributos de esos conceptos, y las restricciones sobre esas propiedades. Un ejemplo sería el *Core Model* de GALEN [163].
- Clasificación: una organización de entidades en clases para propósitos concretos, como por ejemplo la elaboración de informes.
- Tesoro: un sistema de términos organizado en base a las relaciones “término más general” y “término más restringido”. Un ejemplo de tesoro es WordNet [136].
- Terminología: cualquiera o todo lo anterior en distintas combinaciones. La mayoría de terminologías en salud consisten, como mínimo, en un vocabulario controlado y en un sistema de identificadores.
- Sistema de codificación: una terminología con identificadores.

En el ámbito sanitario las terminologías tienen un papel variado. Así, por ejemplo, CIE (Clasificación Internacional de Enfermedades) [154] es una terminología promovida por la Organización Mundial de la Salud para la clasificación de diagnósticos clínicos que es utilizada principalmente en estudios de epidemiología y en general con fines administrativos y de gestión sanitaria, mientras que la terminología LOINC [132] contiene un conjunto de códigos e identificadores para documentar pruebas de laboratorio. Por otro lado, el grado de formalización de las terminologías clínicas también es muy dispar. Encontramos terminologías como GALEN [163], que está definida en una lógica descriptiva de bastante expresividad, y otros vocabularios como UCUM [170] que no tiene ninguna formalización. Además, son muchos los vocabularios controlados empleados en biomedicina, lo que ha motivado que sean integrados en el sistema terminológico UMLS que actualmente agrega y organiza semánticamente más de cien terminologías y que representa más de un millón de conceptos biomédicos [19].

En el dominio de la HCE la terminología de referencia a nivel mundial es SNOMED-CT [89]. Esta terminología es el fruto de la fusión de las terminologías SNOMED RT, del Colegio Americano de Patólogos, y la terminología

Clinical Terms versión 3 desarrollada por el servicio nacional de salud del Reino Unido (NHS). Actualmente, la propiedad intelectual SNOMED-CT pertenece a IHTSDO. La terminología cuenta con más de 300.000 conceptos y es distribuida en varios idiomas. La relevancia de esta terminología a nivel internacional queda de manifiesto al ser utilizada en varios organismos de estandarización en el ámbito sanitario como ISO o HL7, y por estar alineada con otras terminologías importantes en sanidad como CIE o LOINC.

La terminología SNOMED-CT se organiza en conceptos, relaciones y descripciones. Cada concepto representa un significado clínico único y es identificado por un código. Los conceptos se organizan en jerarquías y se asocian con otros conceptos a través de relaciones. Una característica destacada de la estructura de SNOMED-CT es que introduce la jerarquía *Linkage concept* que representa los tipos de relaciones que se pueden establecer entre los conceptos. Las descripciones proporcionan el término preferente para los conceptos así como varios sinónimos. SNOMED-CT está definida formalmente con una lógica descriptiva de expresividad limitada ( $\mathcal{EL}$ , véase apartado 4.3.3), que está justificada por los requisitos de escalabilidad. En cualquier caso, es suficiente para las tareas de validación necesarias durante el proceso de desarrollo. Además, la distribución de la terminología incluye un programa que permite obtener su representación en OWL.

SNOMED-CT es una terminología composicional, es decir, podemos componer nuevos conceptos a partir de otros ya existentes a través de un mecanismo denominado *postcoordinación*. La postcoordinación ofrece la posibilidad de definir conceptos complejos a partir de otros más simples aprovechando las relaciones que se establecen entre ellos. Por ejemplo, el concepto que define la *medida de la presión arterial* (163020007) está relacionado a través del atributo *sitio del hallazgo* (363698007) con el concepto *estructura del sistema cardiovascular* (113257007). El concepto postcoordinado *medida de la presión sanguínea en el brazo* se obtendría combinando el concepto *medida de la presión arterial* y *estructura de la arterial braquial* (17137000) a través de la relación *sitio del hallazgo*. Este nuevo concepto ha podido ser definido gracias a que *estructura de la arterial braquial* es descendiente de *estructura del sistema cardiovascular*.

Las terminologías pueden clasificarse en dos grupos: terminologías de interfaz y terminologías de referencia [30]. Las primeras son concebidas para ser usadas por los usuarios finales de los sistemas de información [165]. Los términos de estas terminologías se enlazan con una terminología de referencia o bien son un subconjunto de ella. En cambio, las terminologías de referencia son las que definen la semántica de las entidades que se emplean en los sistemas de información, es decir, representan la ontología del dominio. En este sentido, hoy en día SNOMED-CT se está imponiendo como terminología de referencia para abordar la interoperabilidad semántica de la HCE.

Finalmente, los modelos de información de la HCE suelen presentar ciertos solapamientos semánticos con las terminologías clínicas, debido en especial a que han sido desarrollados independientemente [113]. Por otro lado, los modelos de dominio (arquetipos) deben ser alineados con las terminologías para aportar el significado a los datos de la HCE. El uso de las terminologías clínicas y su alineamiento correcto con los modelos clínicos es un factor determinante para lograr la interoperabilidad semántica de la HCE.

## 2.3. Web Semántica

La Web Semántica es una extensión de la web que tiene como objetivo la automatización del procesamiento de los documentos y la recuperación de la información, proporcionando a la información un significado bien definido que permita a los ordenadores y a las personas trabajar en cooperación [14]. El significado de los recursos web es establecido mediante metadatos formales, definidos gracias a nuevos estándares y tecnologías, que serán gestionados por agentes software para descubrir e integrar la información. La iniciativa de la Web Semántica está coordinada por el consorcio W3C <sup>1</sup>, que es una organización internacional dedicada a la creación y difusión de estándares y guías para la web.

La propuesta de la Web Semántica está ganando aceptación conforme se están publicando las especificaciones y se dispone de aplicaciones que demuestran sus beneficios. En la actualidad el mayor interés en las tecnologías de la Web Semántica está en el área de la integración de la información, y en especial, en el dominio de la salud y biología para el que se ha definido un grupo de interés en W3C [56].

La arquitectura de la Web Semántica se organiza como una pila de lenguajes y tecnologías, donde cada capa aprovecha las capacidades de las capas inferiores (véase figura 2.2, [174]). El nivel inferior de la arquitectura está representado por la capa URI/IRI que tiene como propósito la identificación de los recursos en la Web Semántica, mientras que la capa XML proporciona un metalenguaje para definir la sintaxis de los lenguajes de la Web Semántica.

RDF (Resource Description Framework) es el lenguaje para la descripción de los recursos en la Web Semántica [124]. RDF permite establecer metadatos en forma de tripletas, es decir, sentencias que relacionan un sujeto, un predicado y un objeto. Un conjunto de tripletas forman un grafo RDF. Las tres partes de una tripleta pueden ser identificadas con una URI, aunque el sujeto y el objeto pueden ser nodos en blanco, es decir, nodos con identificadores locales dentro del grafo. El objeto de una tripleta también

---

<sup>1</sup><http://www.w3c.org>

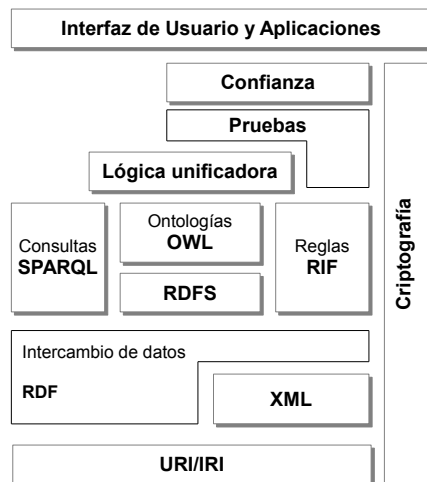


Figura 2.2: Arquitectura de la Web Semántica

puede ser un valor literal como una cadena o un número. Las tripletas en RDF pueden ser *reificadas*, es decir, es posible asignar una URI a una triplete y que ésta participe como sujeto u objeto en otra triplete. Igualmente, el mecanismo de reificación es aplicable a un grafo completo. Finalmente, los datos RDF se almacenan en repositorios y se utiliza el lenguaje SPARQL [160] para realizar consultas.

La sencillez de las primitivas de representación RDF hace que sean insuficientes para determinados escenarios. RDF Schema (RDFS) es una extensión de RDF para representar ontologías básicas mediante constructores adicionales como la definición de dominios y rangos de propiedades o subclases [38]. Sin embargo, la expresividad de RDFS es aún limitada, ya que por ejemplo no es posible expresar que dos conceptos son disjuntos o definir restricciones de la cardinalidad sobre propiedades. Por este motivo, hoy en día el lenguaje recomendado para representar conocimiento ontológico en la Web Semántica es OWL (Web Ontology Language) [12]. El lenguaje OWL es la tecnología semántica utilizada en esta tesis para la representación de modelos clínicos. Por este motivo, se desarrolla la sección 4.4 en el contexto de los lenguajes de representación de conocimiento.

RIF (Rule Interchange Format) es la especificación de un lenguaje de intercambio de reglas que tiene como objetivo proporcionar un marco común para la gestión de reglas en la Web Semántica [20]. Finalmente, las capas superiores de la arquitectura apenas han sido desarrolladas hasta el momento.

## 2.4. Desarrollo de Software Dirigido por Modelos

En Ingeniería del Software los modelos han sido utilizados tradicionalmente como herramienta de documentación y comunicación para representar los aspectos de interés del dominio de un problema. En este sentido, han resultado una herramienta útil en las primeras fases de los proceso de desarrollo de software hasta alcanzar la etapa de implementación. Una vez iniciado el mantenimiento y evolución del software, éstos quedaban obsoletos y ni siquiera servían como documentación del diseño o la arquitectura del sistema.

El Desarrollo de Software Dirigido por Modelos (DSDM), también denominado Ingeniería de Modelos, promueve un papel más activo de los modelos en todas las etapas del ciclo de vida del software. Los modelos pasan a ser artefactos procesables que guían el proceso de desarrollo permitiendo describir tanto el dominio del problema como el dominio de la solución [180].

Para que los modelos puedan ser procesados deben cumplir con una serie de requisitos tanto a nivel técnico como semántico. Un modelo bien definido debe ser conforme a las reglas de un lenguaje de modelado. Éste proporciona las definiciones que permiten la construcción de modelos. Por un lado, define la sintaxis en la que deben ser expresados los modelos y, por otro lado, la semántica de los conceptos y relaciones. A este último aspecto de un lenguaje de modelado se denomina metamodelo. A su vez, los metamodelos son definidos mediante lenguajes de metamodelado.

Las relaciones entre los modelos son definidas utilizando lenguajes de transformación. Estas relaciones se expresan mediante reglas que relacionan los conceptos de dos metamodelos. Un motor de transformación se encarga de obtener un modelo de salida a partir de un modelo de entrada de manera que se mantengan las relaciones expresadas en las reglas de transformación. Finalmente, los modelos son representados como artefactos software utilizando transformación de modelo a texto.

### 2.4.1. MDA

La propuesta más destacada en DSDM es la iniciativa MDA (Model Driven Architecture) liderada por el consorcio OMG [146]. MDA propone el uso de modelos de dominio donde se recoja la lógica de negocio de una organización, al margen de las plataformas tecnológicas concretas donde eventualmente se implementarán las funcionalidades requeridas [111]. Este enfoque fomenta la autonomía de un modelo de negocio con respecto a las tecnologías empleadas para su implementación, y así, ambos pueden evolucionar de forma más independiente. Asimismo, se favorece la interoperabilidad dentro y a través de distintas plataformas tecnológicas.

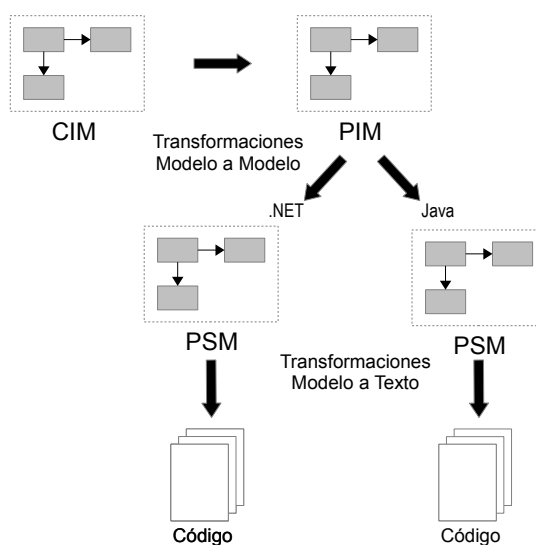


Figura 2.3: Arquitectura MDA

La realización con éxito del enfoque propuesto por la filosofía MDA pasa por establecer adecuadamente las correspondencias entre los modelos de sistema especificados en un nivel de abstracción y los modelos en un nivel de abstracción menor, de manera que se preserven los requisitos iniciales al transformar unos modelos en otros (véase figura 2.3). En el primer nivel se define el modelo independiente de la computación (Computation Independent Model, CIM) que consiste en una representación del problema independiente del espacio de la solución. El segundo nivel contiene los modelos independientes de la plataforma (Platform Independent Model, PIM) que utilizan conceptos de computación tales como clases y objetos. El tercer nivel define los modelos específicos de la plataforma (Platform Specific Model, PSM) que son refinamientos de los modelos PIM que añaden elementos propios de una plataforma de desarrollo, como por ejemplo, transformar clases en componentes de la plataforma Java. Los modelos de estos tres niveles se relacionan a través de transformaciones verticales modelo a modelo, mientras que las transformaciones modelo a texto se utilizan para refinar los modelos PSM y obtener los artefactos software concretos de la plataforma destino: ficheros de configuración, clases, etc.

El OMG ha definido la denominada arquitectura de cuatro capas de metamodelado que establece la base de las relaciones entre modelos y lenguajes de modelado en el desarrollo de software. En cada nivel de modelado se utiliza un lenguaje de metamodelado con el que se definen los modelos de la capa inferior (véase figura 2.4).



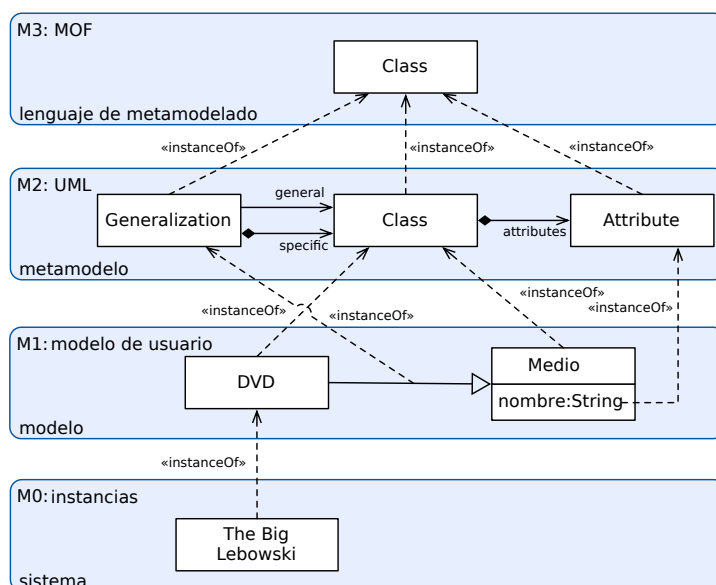


Figura 2.4: Arquitectura de metamodelado del OMG

La capa superior se denomina M3 y contiene los lenguajes de meta-modelado. En esta capa el OMG propone el estándar MOF (Meta Object Facility). Este lenguaje tiene la particularidad de definirse a sí mismo, es decir, es un lenguaje de metamodelado que es conforme a sí mismo. De este modo se consigue limitar las capas de metamodelado de la arquitectura. En la capa inferior se encuentra el nivel M2 en el que se definen los lenguajes de modelado como UML. Estos lenguajes son conformes al estándar MOF. La capa M1 contiene los modelos de un sistema expresados mediante los lenguajes de modelado de la capa M2. Por ejemplo, en esta capa podemos encontrar diagramas de clases UML. En la capa inferior de la arquitectura, denominada M0, se sitúan las instancias del sistema. Aquí encontraríamos los objetos conformes a los diagramas de clases definidos en M1. Por último, la definición de modelos según la arquitectura de cuatro capas del OMG no garantiza que puedan ser procesados e intercambiados por distintas herramientas. El OMG ha propuesto el estándar XMI (XML Metadata Interchange) como formato común para el intercambio de modelos.

La iniciativa MDA se basa fundamentalmente en el uso de estándares: MOF como lenguaje de metamodelado [150], UML como lenguaje de modelado software [166], XMI como formato de intercambio de modelos y metamodelos [151], QVT lenguaje de transformación modelo a modelo [147] y OMG Model to Text como lenguaje de transformación modelo a texto [148]. No obstante, algunos de estos estándares no están bien soportados por las herramientas. Un ejemplo es el lenguaje QVT para que el actualmente no hay disponible ninguna implementación completa.

En la comunidad DSDM han surgido alternativas a los distintos estándares propuestos por el OMG. Muchos de ellos se desarrollaron como respuesta a la petición de propuestas realizada por el OMG. Cabe destacar el proyecto Eclipse Modelling Framework (EMF) que se ha convertido en un estándar de facto en la gestión de modelos [182]. Este framework facilita la definición de modelos mediante diagramas UML, interfaces Java y esquemas XML, y utiliza una simplificación de MOF como lenguaje de metamodelado (Ecore).

### 2.4.2. Transformaciones de Modelos

Las transformaciones de modelos pueden clasificarse atendiendo a los niveles de abstracción de los metamodelos origen y destino en transformaciones verticales u horizontales. Se dice que una transformación es vertical cuando los niveles de abstracción son diferentes, como por ejemplo, cuando se refina un modelo para añadir nuevos detalles. En cambio, una transformación horizontal se expresa en el mismo nivel de abstracción. Así, por ejemplo una refactorización es una transformación que mejora la calidad de un modelo aunque manteniendo el significado. Las transformaciones también pueden ser clasificadas atendiendo al resultado de la transformación en modelo a modelo y modelo a texto.

Las transformaciones modelo a modelo tienen como origen y destino de la transformación modelos. En general, este tipo de transformaciones se emplea en cualquier tarea que quiera automatizarse en Ingeniería de Modelos. El enfoque más común es el denominado *híbrido* que consiste en la combinación de reglas declarativas (establecen las relaciones entre los metamodelos) y construcciones imperativas. Un ejemplo destacado de este enfoque es RubyTL [168].

Las transformaciones modelo a texto tienen como origen modelos y como salida artefactos textuales, como documentos XML o clases Java. La técnica más implementada se basa en *plantillas*. Una plantilla representa la estructura del artefacto textual que va a generarse y se introducen sentencias de control, como bloques condicionales o repeticiones, que tienen acceso a la información del modelo. Un ejemplo de lenguaje que implementa la técnica basada en plantillas es MOFScript [144].

### 2.4.3. Relación con otros espacios tecnológicos

Una de las características más destacadas de la Ingeniería de Modelos ha sido el papel de puente que establece entre distintos espacios tecnológicos. Un espacio tecnológico puede definirse como un contexto de trabajo en el que una comunidad de usuarios comparte conceptos y herramientas [114]. Por ejemplo, la Web Semántica es un espacio tecnológico en el que se manejan conceptos como ontología, se utilizan lenguajes de modelado como

OWL y herramientas como los razonadores. Otros ejemplos de espacios tecnológicos con el propio DSDM o XML.

En la comunidad de Ingeniería de Modelos ha surgido un gran interés en el desarrollo de lenguajes específicos de dominio (Domain Specific Language, DSL). Estos lenguajes pretenden elevar la abstracción del programador permitiendo representar la solución utilizando conceptos del dominio del problema. Para ello se ha integrado el espacio tecnológico de las gramáticas (notación BNF, analizadores léxicos y sintácticos, etc.) con los modelos. Se dice que una gramática expresa la sintaxis concreta de un DSL, mientras que la sintaxis abstracta o semántica se especifica a través de un metamodelo. La relación entre ambas sintaxis define el puente entre los espacios tecnológicos de las gramáticas y los modelos.

El consorcio OMG ha propuesto la especificación ODM (Ontology Definition Metamodel) [149] que recoge la definición de lenguajes de representación de conocimiento, como por ejemplo OWL, utilizando un metamodelo en el nivel M2 de la arquitectura de cuatro capas de metamodelado. Además, incluye una recomendación de correspondencias entre lenguajes de modelado en Ingeniería del Software como UML con los lenguajes de representación de conocimiento. Estas correspondencias son expresadas a través de reglas de transformación QVT.



## Capítulo 3

# Arquitectura de Modelo Dual

### 3.1. Introducción

La arquitectura de modelo dual nació como una solución a los problemas clásicos de evolución y mantenimiento de los sistemas de información. Thomas Beale argumentó que muchos de los sistemas de información son desarrollados de un modo en el que los conceptos del dominio que deben ser procesados por el sistema son codificados directamente en el software y en los modelos de bases de datos dando lugar a sistemas con un mantenimiento costoso y que tienen una vida limitada, especialmente en dominios complejos como la medicina donde los cambios son frecuentes [9].

La aproximación basada en la arquitectura de modelo dual, o simplemente arquitectura dual, define un *modelo de referencia*, utilizado para representar las propiedades genéricas de la información del historial de salud de los pacientes, y un modelo de arquetipos, que recoge las necesidades de información de una determinada especialidad o servicio sanitario. Por lo tanto, esta arquitectura requiere que sean identificados los requisitos de información básicos del dominio que han de ser establecidos en el modelo de referencia. Los conceptos que pertenecen al modelo de referencia representan las características estables del dominio y establecen entre sí las relaciones válidas que permiten configurar las entidades del dominio (informes de alta, medida de la presión sanguínea, etc.). Los conceptos del dominio en la arquitectura dual se denominan *arquetipos* y su formalización es diferente a los modelos de información.

Desde el punto de vista del desarrollo de un sistema de información, la característica más destacada de la arquitectura de modelo dual es la separación entre el modelo de información, que es implementado en el software y en la base de datos (modelo de referencia), y los conceptos del dominio (arquetipos) que son responsabilidad de los expertos clínicos. Esta característica permite que los sistemas pueden ser desarrollados sin la necesidad de que los conceptos clínicos estén definidos. Éstos son independientes

del proceso software y pueden ser introducidos cuando el sistema ya esté implantado. Así pues, la adopción de la arquitectura dual permite a los expertos en el dominio sanitario especificar los modelos de contenido para los conceptos clínicos en uso sin necesidad de conocimientos técnicos. De este modo se logra separar el conocimiento del dominio y la información que maneja un sistema de HCE. Por lo tanto, los sistemas HCE pueden adaptarse a los cambios en la práctica médica y en la prestación de servicios sanitarios a lo largo del tiempo [63].

Un arquetipo es la definición *formal* de un modelo de conocimiento que establece la estructura de la información utilizada en un contexto particular, como por ejemplo un formulario de admisión en urgencias, por medio de combinaciones legales de los conceptos de negocio definidos en el modelo de referencia. El *modelo de arquetipos* (AOM) es el formalismo para representar los arquetipos [10]. Los conceptos que incluye son independientes de cualquier modelo de referencia y sólo se relaciona formalmente con los conceptos del lenguaje de modelado tales como clases, atributos y asociaciones, si usamos un modelo orientado a objetos expresado en UML [166]. El modelo de arquetipos ofrece las construcciones necesarias para establecer el conjunto de restricciones que definen las estructuras de datos válidas para representar un concepto clínico de acuerdo a un modelo de información: nombres de los componentes, opcionalidad, cardinalidad, tipos de datos, rangos de valores que pueden tomar los elementos, etc.

La arquitectura dual ofrece una solución al desafío de la interoperabilidad en los sistemas de información sanitarios. El uso de un modelo de información común y limitado solamente a los requisitos médico-legales y a las funciones de gestión de los historiales permite que un extracto de HCE sea interpretado fielmente en un sistema receptor incluso si el contenido clínico no ha sido acordado de antemano (interoperabilidad funcional). Además, si las estructuras de datos clínicas son compartidas utilizando arquetipos y los extractos de HCE son construidos conforme a la definición de los arquetipos, el sistema receptor podrá interpretar el significado de la información (interoperabilidad semántica). Sin embargo, para que los arquetipos resulten una herramienta eficaz para lograr la interoperabilidad semántica, es necesario que sean compartidos y utilizados de forma consistente entre sistemas HCE. Por tanto, deben ser considerados como activos de conocimiento que han de incorporarse en el diseño de aplicaciones clínicas de forma análoga a los sistemas terminológicos [104].

La arquitectura dual y los arquetipos clínicos son fruto de más de una década de investigación y demostración clínica en Europa y Australia. La arquitectura dual ha influido en el diseño de la arquitectura de información de openEHR, aunque en los últimos años ha sido empleada como una solución para la estandarización de la comunicación de la HCE en el estándar ISO EN 13606 y para el desarrollo del modelo de *templates* HL7 CDA.

## 3.2. Lenguaje de Definición de Arquetipos

El lenguaje de definición de arquetipos (ADL) [11] ha sido desarrollado por la Fundación openEHR y recientemente se ha estandarizado como parte de la norma ISO EN 13606 parte 2 [97]. El lenguaje ADL contiene dos sintaxis: dADL (definición de datos) y cADL (definición de restricciones). La sintaxis dADL se utiliza para representar instancias del modelo de referencia, la metainformación incluida en la cabecera del arquetipo y para la descripción de los términos empleados en la definición. En cambio, la sintaxis cADL se utiliza en la definición del contenido clínico que representa el arquetipo.

```
archetype (adl_version=1.4)
  openEHR-EHR-CLUSTER.exam.v1
  concept
    [at0000] -- Examen
  language
    original_language = <[ISO_639-1::en]>
  description
    original_author = <
      ["name"] = <"Heather Leslie">
      ["email"] = <"heather.leslie@oceaninformatics.com">
      ["date"] = <"10/07/2010">
    >
    lifecycle_state = <"AuthorDraft">
    ...
```

Figura 3.1: Ejemplo de sección *cabecera* de un arquetipo

Un arquetipo consta de tres secciones. En primer lugar, la sección *cabecera* describe el arquetipo (véase figura 3.1): el identificador, el identificador del arquetipo padre en caso de que establezca una relación de especialización, el concepto del dominio que está definiendo y metainformación como el autor, el propósito, versión, etc. En segundo lugar, la sección *definición* contiene la descripción formal del arquetipo organizada en *términos* que establecen restricciones sobre las entidades del modelo de referencia (véase figura 3.2). Por último, la sección *ontología* contiene fundamentalmente la descripción textual en uno o más idiomas del vocabulario empleado en la definición del arquetipo y el enlace de los códigos asociados al vocabulario del arquetipo con sistemas terminológicos como SNOMED-CT o LOINC. En la figura 3.3 se muestra la sección *ontología* de un arquetipo que define textualmente la raíz del arquetipo (término *at0000*) y establece un enlace terminológico con SNOMED-CT.

La sintaxis cADL ofrece las construcciones necesarias para establecer restricciones sobre estructuras de datos conformes a un modelo de referencia orientado a objetos. Una definición válida en cADL debe cumplir dos requisitos: (1) los identificadores de las clases, atributos y asociaciones utilizados en las expresiones cADL han de corresponder con algún identificador del modelo de referencia y (2) no puede relajar una definición del modelo de referencia, por ejemplo, no puede declarar un atributo opcional si en el modelo de referencia es obligatorio. Las restricciones en cADL se organizan de forma jerárquica, de modo que se van alternando restricciones de clases (términos) y restricciones de atributos. La sección *definición* de un arquetipo comienza con la restricción sobre la clase de la entidad de negocio del modelo de referencia que está especializando el arquetipo (véase figura 3.2).

```
definition
  CLUSTER[at0000] matches { -- Examen
    items cardinality matches {1..*; unordered} matches {
      -- Hechos normales
      CLUSTER[at0001] occurrences matches {0..1} matches {
        items cardinality matches {1..*; unordered} matches {
          -- Hecho normal
          ELEMENT[at0005] occurrences matches {0..*} matches {
            value matches {
              DV_TEXT matches {*}
            }
          }
        }
      }
    }
  ...
```

Figura 3.2: Ejemplo de sección *definición* de un arquetipo

En la sección 3.3 se desarrolla la construcción un arquetipo para un concepto clínico donde se motivan algunas de las restricciones del lenguaje ADL. A continuación se incluyen las restricciones más relevantes que pueden definirse en cADL:

- **Existencia.** Es una restricción aplicable sólo a los atributos y significa que el atributo debe tener un valor.
- **Cardinalidad.** Esta restricción puede ser aplicada solamente en atributos multivaluados. Permite indicar las características de la colección que almacena las instancias (orden y unicidad) y el número de elementos que puede contener. Por ejemplo, restringir la cardinalidad



con una colección no ordenada y sin repetidos correspondería con a establecer la semántica de un conjunto.

- **Ocurrencia.** Es aplicable sobre clases y especifica el número de instancias de la clase que pueden estar asociadas al contenedor en el que está declarada. Los arquetipos definen estructuras jerárquicas en las que, a excepción de la raíz, cada componente está contenido por otro (contenedor) con el que se relaciona a través de un atributo.
  
- **Restricciones sobre tipos primitivos.** Las posibilidades de restricción de los valores de los tipos primitivos son amplias. Por ejemplo, las cadenas pueden ser acotadas a través de una expresión regular o los tipos numéricos pueden ser restringidos con un intervalo. En el capítulo 5 que trata la semántica del modelo de arquetipos en OWL se analizan en más detalle las principales restricciones sobre tipos primitivos (véase sección 5.2).
  
- **Alternativas.** Como rango de un atributo sencillo (cardinalidad máxima 1) puede aparecer la declaración de dos o más términos. Esta restricción debe interpretarse como que los términos son alternativas y una instancia concreta sólo podrá contener una de ellas.
  
- **Restricción *any*.** Es un tipo de restricción especial que indica que no se establece restricción. Resulta de utilidad para especificar que el rango de un atributo es de un cierto tipo, pero sin importar su contenido.

El lenguaje cADL ofrece dos mecanismos de reutilización. Por un lado, podemos declarar que las restricciones asociadas a un término corresponden con otro previamente declarado. Esta relación se conoce como *referencia interna* y se establece con la palabra clave `use_node` y la ruta del término cuya definición se reutiliza. Por otro lado, ADL ofrece un mecanismo de composición que permite definir los términos en la definición de un arquetipo que deben ser completados con otros arquetipos (*slot*). La definición de un *slot* consiste normalmente en una expresión regular que restringe el identificador de los arquetipos que están permitidos.

```

ontology
  terminologies_available = <"SNOMED-CT", ...>
  term_definitions = <
    ["en"] = <
      items = <
        ["at0000"] = <
          text = <"Examen">
          description = <"Describe hallazgos de exámenes físicos">
        >
        ...
      >
    >
  >
  term_bindings = <
    ["SNOMED-CT"] = <
      items = <
        ["at0000"] = <[SNOMED-CT(2010)::425044008]>
        ...
      >
    >
  >

```

Figura 3.3: Ejemplo de sección *ontología* de un arquetipo

El modelado con arquetipos ofrece mecanismos de versionado, especialización y composición que favorecen la reutilización y mantenimiento de los modelos de contenido clínico. En primer lugar, la modificación de un arquetipo debe dar lugar a una nueva versión que ha cumplir que los datos conformes a la versión revisada sean conformes también con las versiones previas. Si no fuera así, debería crearse un nuevo arquetipo o marcar como obsoleta la versión anterior. En segundo lugar, la especialización permite relacionar dos arquetipos de modo que los datos creados de acuerdo al arquetipo especializado deben ser conformes con el arquetipo padre (o arquetipo base). En otras palabras, las restricciones que establece el arquetipo especializado (o arquetipo hijo) deben ser más fuertes que las del arquetipo base. Además, es importante destacar que la relación de especialización no es una relación de herencia. En la sección 3.3 se desarrolla un ejemplo que introduce una relación de especialización entre arquetipos. Por último, la composición de arquetipos es ofrecida a través de los *slots*.

La Fundación openEHR añade a la arquitectura dual el concepto de *plantilla*. Una plantilla openEHR resulta de la composición de varios arquetipos en estructuras que suelen corresponder con informes o formularios en pantalla. Las plantillas restringen normalmente las definiciones de los arquetipos para ser utilizados en escenarios concretos. En general, los ar-

quetipos definen todo lo que puede ser documentado sobre una entidad clínica concreta, mientras que las plantillas definen la información que debe ser almacenada, recuperada, presentada o intercambiada en un contexto específico. Las relaciones de *slot* definidas en los arquetipos controlan las posibilidades de composición de las plantillas. Las plantillas permiten elegir de entre todos los arquetipos candidatos aquellos que son adecuados para una tarea particular. Finalmente, la noción de plantilla no ha sido incorporada al estándar ISO EN 13606 debido a que su utilidad no está orientada a la construcción de sistemas de HCE, sino a la comunicación de la HCE.

### 3.3. Modelado de arquetipos

El modelado de arquetipos consiste principalmente en traducir un concepto clínico a entidades de un modelo de referencia y restringir sus estructuras para definir sus requisitos de información. En [33] se ha propuesto un proceso de desarrollo de arquetipos basado en un proceso iterativo de refinamiento. En esta sección se desarrolla un proceso informal que es ilustrado con un ejemplo de concepto clínico. La figura 3.4 presenta el modelo conceptual que representa el concepto clínico *Examen*. La interpretación intuitiva de este concepto es que un examen tiene una serie de componentes que son: hechos normales, una descripción clínica, una serie de hallazgos e imágenes. Además, cada hallazgo tiene asociado detalles.

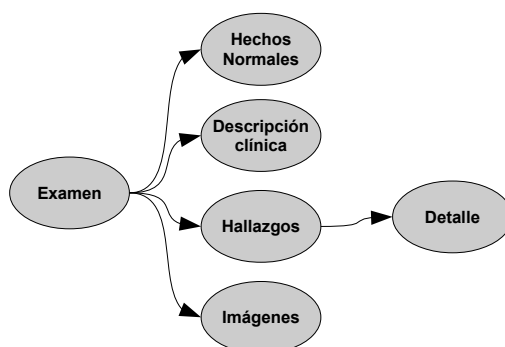


Figura 3.4: Modelo conceptual del concepto clínico *Examen*

Esta idea conceptual puede ser representada como un arquetipo. Una vista abstracta del arquetipo openEHR *Examen* es mostrada en la figura 3.5. El concepto raíz (*Examen*) se representa por medio de un `CLUSTER`, que permite definir conceptos compuestos a través del atributo `items`. Los conceptos simples son expresados como `ELEMENT`. Cada concepto tiene un identificador (por ejemplo, `at0000` corresponde con *Examen*) cuyo significado es definido en la sección *ontología* del arquetipo.

Los conceptos en los arquetipos se caracterizan por el número de instancias que pueden tomar parte de la asociación a la que pertenecen. Esta propiedad se denomina ocurrencias y se define con un rango de enteros que puede no estar acotado superiormente. Por ejemplo, la *Descripción Clínica* es opcional y como mucho puede haber una sola instancia (0..1).

Los atributos multivaluados (por ejemplo, *items*) pueden ser restringidos de distintos modos. En primer lugar, el intervalo de cardinalidad de un atributo puede ser restringido con un rango de enteros. Por ejemplo, una instancia de *Examen* está asociada con al menos una instancia (1..\*) de otros conceptos a través del atributo *items*. La segunda restricción permite indicar si las instancias deben estar ordenadas. Por último, se puede especificar si está permitido que haya instancias repetidas en la asociación.

Los datos son almacenados en estructuras de datos primitivas como textos, fechas o cantidades. Estas estructuras son contenidas en entidades ELEMENT. La figura 3.5 muestra que la *Descripción Clínica* está representada como texto y que *Imágenes* es un dato multimedia que sólo permite imágenes en formatos png, gif y jpeg.

En el arquetipo *Examen*, el concepto *Detalle* se especifica como un *slot* que permite incluir otros arquetipos como “Auscultación”, “Inspección” o “Palpación”.

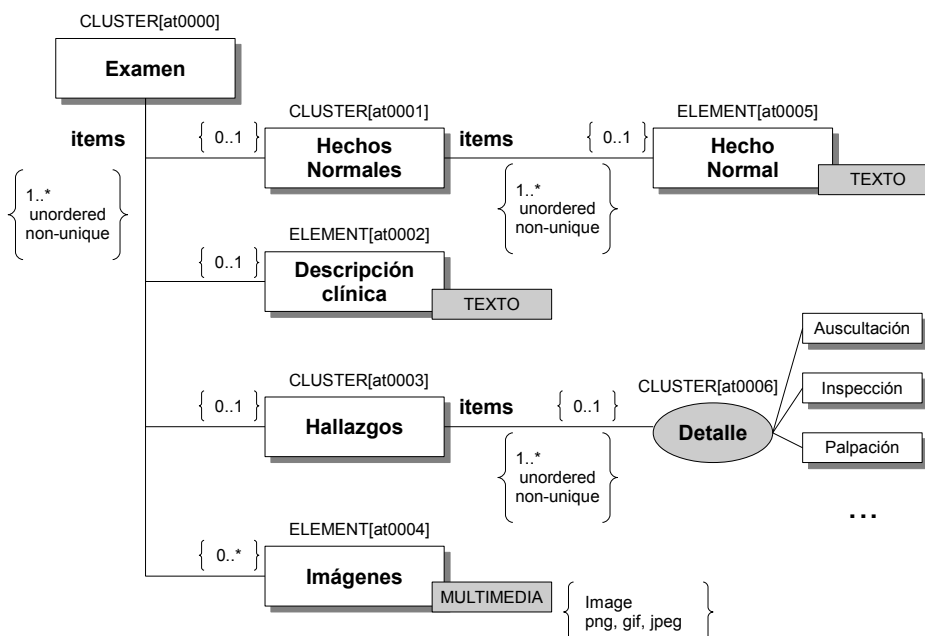


Figura 3.5: Esquema del arquetipo *Examen* en openEHR

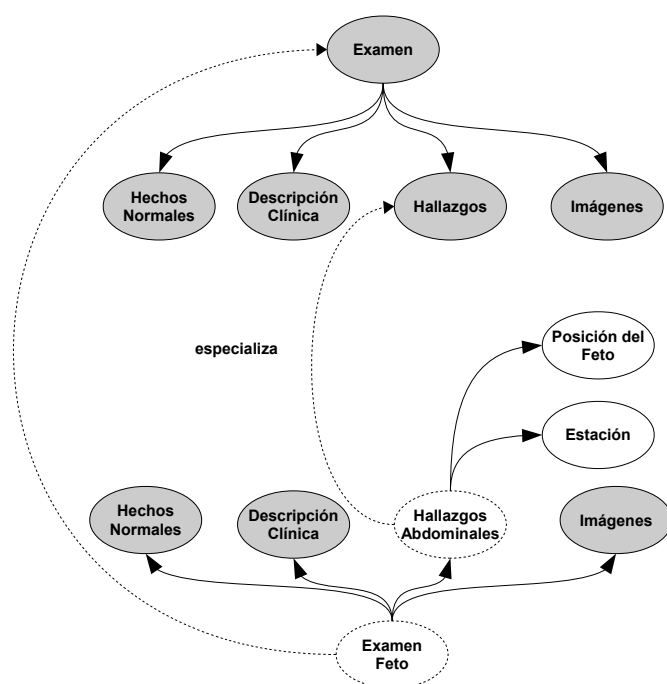


Figura 3.6: Mapa conceptual del concepto clínico *Examen del feto* como una especialización de *Examen*

Un arquetipo puede ser definido como la especialización de otro. Esta especialización no implica la reutilización de definiciones como sucede en el modelado orientado a objetos, sino que es una relación de conformidad. En otras palabras, si un arquetipo B especializa un arquetipo A, entonces todos los extractos HCE que son compatibles con el arquetipo B también debe ser compatibles con el arquetipo A. La figura 3.6 muestra parte del modelo conceptual del concepto clínico *Examen del feto*, que se define como una especialización de *Examen*. El arquetipo especializa el concepto raíz, *Examen*, y también el concepto *Hallazgos* para introducir información concreta sobre la posición del feto, la presentación, etc.

### 3.4. Enlace terminológico

En el modelado de arquetipos el proceso de enlace terminológico consiste en asignar conceptos de terminologías clínicas al vocabulario empleado en la definición del arquetipo. Por una parte, es habitual que un arquetipo restrinja los valores aceptables en una entrada clínica a un conjunto de datos que aporten significado a la información. Así, por ejemplo, un arquetipo que modela la medida de la presión sanguínea puede definir el campo

*posición del paciente* limitando los valores a *sentado, acostado* o *de pie*. Por otro lado, los términos en los que está organizada la definición del arquetipo representan campos de información cuya semántica debe ser especificada, como el caso de la *presión sistólica* en la medida de la presión sanguínea.

Actualmente la mayoría de los arquetipos públicos apenas están enlazados con terminologías. Esta situación ha motivado la investigación en técnicas que faciliten esta tarea. Es más, el desarrollo de buenas prácticas para el enlace terminológico de los arquetipos es reconocido como una necesidad para alcanzar la interoperabilidad semántica de la HCE [183]. En este apartado se presentan los trabajos más significativos que han estudiado el problema del enlace terminológico, poniendo el foco en las características que determinan que un enlace sea consistente o de buena calidad. La mayoría de estos trabajos utilizan SNOMED-CT como terminología de referencia.

El sistema MoST es una metodología para abordar la interoperabilidad de modelos de información y terminologías clínicas propuesta en la Universidad de Manchester [34]. Esta aproximación ha sido implementada y validada con arquetipos openEHR y la terminología SNOMED-CT. El proceso propuesto consiste en utilizar recursos lingüísticos y en la aplicación de técnicas de procesamiento del lenguaje natural a las definiciones textuales de los códigos de un arquetipo para obtener un conjunto de conceptos candidatos en SNOMED-CT. La elección del código correcto no se realiza de forma automática, sino que se deja en manos del diseñador del arquetipo. La limitación principal de la metodología MoST es la ausencia de técnicas semánticas que aprovechen las relaciones de SNOMED-CT para reducir el número de conceptos candidatos.

En [134] se propone un método automático de enlace terminológico que combina técnicas lingüísticas y semánticas. El método ha sido analizado con 25 arquetipos anotados manualmente por los autores del estudio como referencia para la validación (477 términos anotados). Los resultados muestran que la explotación de las relaciones semánticas de SNOMED-CT mejora significativamente el rendimiento frente a otras técnicas basadas sólo en técnicas y recursos lingüísticos. Asimismo, el estudio revela que más de una tercera parte de los términos que definen un campo del arquetipo (clase `ELEMENT` de openEHR) están relacionados semánticamente con otros términos del contexto de su declaración y una cuarta parte de los códigos asociados a los conjuntos de valores (nodos hoja en la estructura jerárquica del arquetipo) están relacionados en SNOMED-CT. Los autores argumentan que estos resultados son fruto de la aplicación del *principio de proximidad* que establece que si dos términos están agrupados lógicamente en un arquetipo, entonces los conceptos de la terminología también lo estarán.

Algunas propuestas han tratado el proceso de enlace terminológico como soporte para otros objetivos de investigación. En [193] se realiza un estudio de la cobertura de la terminología SNOMED-CT por parte de los arquetipos del repositorio NHS. La técnica de enlace aplicada es de naturaleza algorítmica, es decir, no emplea recursos lingüísticos ni aprovecha la semántica de la terminología. El resultado del proceso se representa en un documento denominado *sombra terminológica*, que asocia los términos del arquetipo con los conceptos SNOMED-CT. Los autores justifican la utilidad de las *sombras terminológicas* como mecanismo para comparar arquetipos y evaluar sus diseños, aunque la representación propuesta no es ontológica, lo que impediría aplicar técnicas semánticas de comparación.

En [117] se ha analizado el enlace de arquetipos openEHR con la terminología UMLS. El objetivo del trabajo es obtener relaciones semánticas entre arquetipos que faciliten la organización y navegación de los repositorios. El método propuesto aprovecha las herramientas de análisis lingüístico ofrecidas por UMLS para obtener un conjunto de códigos que definen semánticamente el arquetipo. Las relaciones entre arquetipos se construyen a partir de las relaciones semánticas que establecen los códigos de los arquetipos en la terminología UMLS.

El proceso de enlace terminológico se ha empleado como mecanismo para obtener la ontología de un arquetipo. En [169] se propone un método de extracción de una subontología de SNOMED-CT a partir de los enlaces terminológicos de un arquetipo. El objetivo del trabajo es asistir el proceso de edición de plantillas openEHR seleccionando arquetipos relevantes semánticamente en las relaciones de *slot*. La técnica aplicada destaca por construir la ontología del arquetipo aprovechando la semántica de las relaciones de SNOMED-CT. Sin embargo, la aplicación de las ontologías a la selección de arquetipos candidatos en las relaciones de *slot* se basa en buscar relaciones entre los conceptos asociados al *slot* y los conceptos de la ontología en la terminología SNOMED-CT sin considerar el papel tienen los conceptos en la definición del arquetipo (término raíz, nodo hoja).

Por último, la relación semántica entre los modelos de información y las terminologías ha sido estudiada desde otra perspectiva. En [162] se ha definido una *interfaz* para el enlace de códigos de una terminología basada en OWL. Esta interfaz especifica cómo se pueden utilizar los códigos de los sistemas terminológicos en las estructuras de datos de la HCE. La propuesta consiste en el uso de OWL como mecanismo de restricciones para establecer el conjunto de posibles códigos que pueden ser asociados con una estructura de datos.

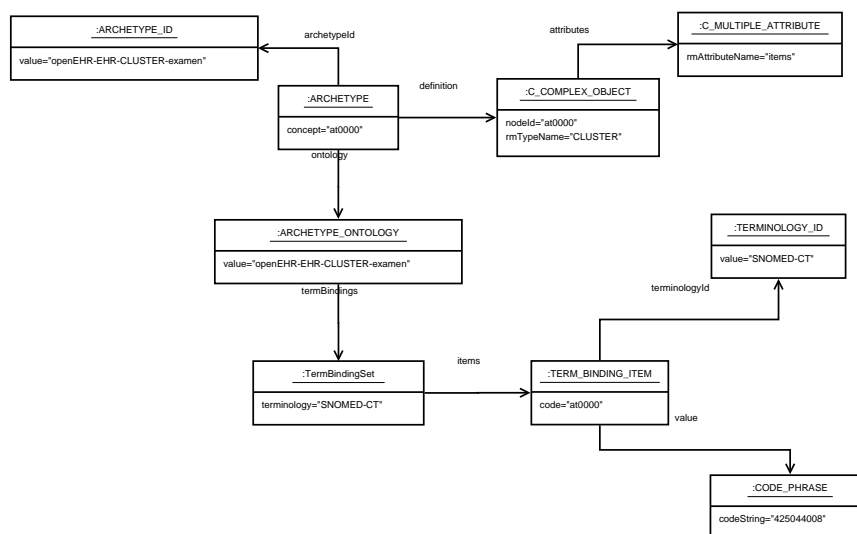


Figura 3.7: Fragmento del modelo AOM del arquetipo *Examen*

### 3.5. Limitaciones del Modelo de Arquetipos

El paradigma de modelado con arquetipos tiene la ventaja de ser una metodología neutral para definir los conceptos de un dominio con independencia de un modelo de referencia concreto. No obstante, esta flexibilidad plantea algunos inconvenientes, debido principalmente a que la definición de un arquetipo no puede ser analizada de forma independiente. La figura 3.7 muestra un fragmento del modelo AOM del arquetipo *Examen* presentado en la sección 3.3. Los términos del arquetipo son objetos de tipo `C_COMPLEX_OBJECT` y las restricciones de atributos se expresan en objetos `C_MULTIPLE_ATTRIBUTE`, en el caso de que el atributo sea multivaluado como en el ejemplo. Nótese que las referencias al modelo de información de openEHR que se expresan en los atributos `rmTypeName` y `rmAttributeName` son definidas simplemente como cadenas.

La semántica del modelo de arquetipos establece que un arquetipo especializa una entidad de un modelo de referencia. Sin embargo, las herramientas de análisis del lenguaje ADL se limitan a comprobar la corrección sintáctica del documento y, por tanto, no validan las declaraciones cADL. Es responsabilidad de los entornos de edición de arquetipos analizar que se respeta la semántica de las definiciones respecto al modelo de referencia. Este análisis resulta costoso debido a que el formalismo en el que se expresan los arquetipos es distinto a los lenguajes de modelado utilizados para construir los modelos de información. Por ejemplo, el entorno de edición LinkEHR-Ed implementa un sistema de tipos específico para la validación semántica de los arquetipos [123].



Los arquetipos pueden ser representados con una sintaxis basada en XML. Esta opción tiene como ventaja que tanto arquetipos como modelos de referencia pueden estar expresados en el mismo lenguaje de modelado (XML Schema). Sin embargo, el esquema asociado al modelo de arquetipos no permite referenciar correctamente las entidades del modelo de referencia. Así pues, no podemos aprovechar las herramientas de validación XML para chequear al menos que los identificadores del modelo de referencia que se emplean en la definición del arquetipo son correctos.

Por último, a pesar de que el modelo de arquetipos ofrece soporte para el enlace del vocabulario utilizado en el arquetipo con sistemas terminológicos, la relación semántica que se establece es *débil*. En la figura 3.7 se observa que el enlace del concepto raíz del arquetipo *Examen* se expresa con cadenas, tanto el código como la terminología. Por tanto, para poder validar la consistencia del significado clínico asociado a un arquetipo es necesario que tanto arquetipos como terminologías estén representados en el mismo formalismo.

### 3.6. Calidad en arquetipos

Los arquetipos clínicos están ganando aceptación como el formalismo mejor soportado para representar e intercambiar los modelos de contenido clínico, gracias especialmente a la contribución de la Fundación openEHR al desarrollo y promoción del modelo de arquetipos. El respaldo oficial a la arquitectura dual se ha logrado gracias a la estandarización en Europa de la norma EN 13606, que ha sido adoptada a nivel internacional como norma ISO. Recientemente el proyecto SemanticHEALTH [183] y la iniciativa CIMI han propuesto el paradigma de arquetipos como representación canónica de los modelos clínicos [31].

Los arquetipos han sido diseñados para tener un papel clave en el funcionamiento de los sistemas de información sanitarios. En concreto, los arquetipos influyen en el modo en el que las aplicaciones sanitarias interactúan con los usuarios, en particular, guiando la introducción y presentación de la información clínica. Por consiguiente, los arquetipos deben ser considerados un activo de conocimiento que ha de ser incorporado en el diseño de las aplicaciones clínicas, al igual que otros recursos de conocimiento en el ámbito sanitario como los sistemas terminológicos. Sin embargo, para que los arquetipos sean aceptados y adoptados ampliamente en los sistemas de información sanitarios deben tener una calidad demostrable. En la actualidad no se dispone de experiencia suficiente en el diseño de arquetipos para poder aplicar un proceso riguroso de certificación de calidad. La mayoría de arquetipos públicos desarrollados hasta la fecha han sido publicados por la Fundación openEHR y por NHS, donde no se han aplicado procesos que garanticen la calidad de los arquetipos [103].

El instituto EuroRec ha elaborado un estudio sobre los criterios de calidad de arquetipos a través de los requisitos administrativos, clínicos, tecnológicos, de gestión de la información y de funcionamiento de los repositorios [104]. El estudio destaca la importancia del modelado formal de arquetipos y la necesidad de establecer el significado clínico de los modelos de forma consistente. En este sentido, cabe destacar la recomendación de enlazar las definiciones de los arquetipos con terminologías clínicas y que los conceptos elegidos sean precisos con el fin de minimizar el solapamientos y lagunas entre arquetipos.

Entre las acciones recomendadas en el informe final del proyecto SemanticHEALTH para promover la calidad de los arquetipos destacan la elaboración de buenas prácticas de modelado de arquetipos, metodologías para enlazar arquetipos con terminologías clínicas y el desarrollo de herramientas de validación de arquetipos. Los requisitos técnicos que debe cumplir la definición de cualquier arquetipo están recogidos en la norma ISO EN 13606 Parte 2.

Sin embargo, para poder analizar la calidad técnica y la consistencia clínica de un arquetipo es necesario formalizar el modelo de arquetipos y la relación que establece con el modelo de referencia y las terminologías clínicas. En este sentido, en [122] se realizó el primer análisis exhaustivo de la semántica de la arquitectura dual en el marco de la norma ISO EN 13606. El trabajo formaliza un sistema de tipos que unifica el modelo de referencia y el modelo de arquetipos. Los resultados del estudio han sido implementados en el editor de arquetipos LinkEHR-Ed [123].

La característica más destacada del modelo formal de LinkEHR-Ed es la definición de la relación de subsunción entre tipos y un algoritmo que encuentra las relaciones de subsunción entre tipos de datos. La relación de subsunción es aplicada para formalizar la relación de especialización entre arquetipos. Además, dado que el modelo de referencia también puede ser abstraído con el sistema de tipos propuesto, la relación de subsunción también es utilizada para formalizar la relación entre los conceptos de negocio del modelo de referencia y los arquetipos. Por último, el sistema de tipos propuesto en LinkEHR-Ed ofrece la posibilidad de construir un editor de arquetipos flexible capaz de trabajar con varios modelos de referencia como ISO EN 13606, openEHR o HL7 CDA.

La calidad de los modelos clínicos detallados [66] ha sido objeto de investigación en los últimos años ([36], [1], [86]). En general, estos trabajos consolidan la arquitectura de modelo dual y establecen una serie de criterios que son satisfechos por los arquetipos: uso de un lenguaje formal y estándar de modelado (ADL), soporte para enlazar conceptos de terminologías internacionales, mecanismos de evolución (versiones) y poder expresar semánticas entre los modelos (composición y especialización). También se establecen otros requisitos relativos al proceso de desarrollo y difusión de

los modelos clínicos, como por ejemplo que el modelado esté basado en la evidencia clínica, un mecanismo de aprobación por expertos o un repositorio para la compartición de modelos.

Recientemente se ha publicado un estudio que analiza los trabajos más relevantes sobre calidad en modelos clínicos detallados que propone un conjunto de métricas que permiten evaluar la calidad de arquetipos [2]. Para la definición de las métricas se ha aplicado el modelo de evaluación de calidad de software ISO/IEC 25010 [100]. Las métricas obtenidas inicialmente fueron validadas por 9 expertos internacionales, de las que finalmente fueron seleccionadas 29 métricas que están organizadas en 8 dominios de evaluación.

Los dominios de evaluación que son de interés para el análisis de la calidad del contenido clínico de los arquetipos son: *claridad y presentación y cumplimiento de los estándares*. El primer dominio evalúa el uso de un lenguaje formal de modelado, la codificación de los elementos de datos textuales, el uso adecuado de la cardinalidad y la definición precisa del alcance del modelo. Por otro lado, el dominio *cumplimiento de los estándares* está enfocado a métricas que favorezcan el intercambio de los modelos clínicos. En ellas se propone el uso de estándares para terminologías clínicas, tipos de datos y unidades de medida, y la evaluación de la consistencia de los enlaces con las terminologías clínicas.

### **3.7. Arquetipos en el desarrollo de aplicaciones sanitarias**

Los arquetipos están llamados a ser elementos esenciales en el diseño y desarrollo de los sistemas de información sanitarios [104]. La arquitectura de modelo dual promueve la separación entre el modelo de información, que es implementado en el software y en la base de datos, y los conceptos del dominio que son responsabilidad de los expertos que pueden especificarlos sin necesidad de conocimientos técnicos [9]. En otras palabras, los arquetipos quedan al margen del proceso de desarrollo software y pueden ser introducidos cuando el sistema ya esté implantado. Por lo tanto, los sistemas de información sanitarios pueden adaptarse a los cambios en las prácticas médicas y a la prestación de servicios sanitarios [63].

En este apartado se presentan estudios y herramientas que promueven un papel más activo de los arquetipos en los procesos de desarrollo de software. La mayor parte de los trabajos están orientados a la generación de interfaces gráficas en aplicaciones sanitarias y sólo unos pocos han analizado las posibilidades y limitaciones del modelado de arquetipos en relación con este objetivo.

El proyecto Opereffa es una iniciativa de la Fundación openEHR y el University College of London (CHIME) para el análisis de las relaciones de la arquitectura de modelo dual y las tecnologías para la programación de interfaces gráficas de usuario [153]. El software obtenido en el proyecto consiste en un generador automático de aplicaciones sanitarias y un entorno de desarrollo que asiste el proceso de construcción de las aplicaciones. El estudio realizado señala que los profesionales sanitarios requieren interfaces más ricas que las ofrecidas en las aplicaciones convencionales y como prueba de este argumento es la biblioteca de componentes gráficos desarrollada por Microsoft para el ámbito sanitario (CUI, [135]). Por otro lado, en las aplicaciones basadas en la arquitectura dual es necesario que los arquetipos estén conectados con la interfaz gráfica. Por estos motivos Opereffa propone la tecnología JSF de Java para el desarrollo de interfaces web [64]. Sin embargo, la conclusión más importante de este trabajo es que el principal reto al que se enfrenta el desarrollo de aplicaciones sanitarias es la evolución tecnológica.

El primer trabajo que motivó el uso de arquetipos en el desarrollo de sistemas de información sanitarios fue el sistema PropeR [120]. En 2003 arrancó este proyecto que adopta una solución basada en la arquitectura dual, aunque formalmente no utiliza el paradigma de arquetipos definido por entonces en openEHR. La arquitectura propuesta estaba basada en la definición de modelos de clínicos (arquetipos) sobre el modelo de referencia COAS (Clinical Observation Access Specification) del OMG [145]. Las interfaces de usuario se especificaban en XML a partir de las definiciones de los arquetipos y la implementación como formularios web se obtenía aplicando transformaciones XSLT. A pesar de que en el sistema PropeR las interfaces y las transformaciones se generan manualmente, el proyecto sienta las bases de la definición declarativa de interfaces de usuario utilizando arquetipos.

En 2006, Schuler *et al.* proponen la generación automática de interfaces de usuario a partir de arquetipos openEHR [172]. La aproximación está basada en la especificación declarativa de interfaces utilizando el lenguaje XUL (XML User Interface Language) [187]. Las características técnicas de XUL permiten que las interfaces incluyan validación de datos y sean conectadas con un sistema de información a través de servicios web. El trabajo destaca especialmente por demostrar que los arquetipos contienen la información necesaria para generar aplicaciones totalmente operativas. No obstante, la tecnología XUL tiene el inconveniente de estar vinculada al software de la Fundación Mozilla <sup>1</sup>.

En los últimos años se han desarrollado varias herramientas de software libre o prototipos en proyectos de investigación que automatizan la generación de aplicaciones sanitarias a partir de arquetipos (EHRFlex, EHR-Gen, GastrOS, ZK-ARCHE). Las diversas propuestas se diferencian por ser

---

<sup>1</sup><http://www.mozilla.org/>

neutrales respecto al modelo de referencia o específicas de un estándar, y también por otras características como las tecnologías de desarrollo o la relación con otras tecnologías en Informática Médica.

EHRFlex [24] es un proyecto liderado por el grupo de investigación IBIME, integrado en el instituto ITACA de la Universidad Politécnica de Valencia, también responsable del entorno LinkEHR. La propuesta de EHRFlex consiste en la definición de un modelo de información genérico que es un compendio de varios modelos de referencia. En concreto, está inspirado principalmente por la idea de plantillas operacionales de openEHR. Por tanto, el generador de aplicaciones puede funcionar con cualquier estándar basado en la arquitectura dual, aunque actualmente está implementado sólo para ISO EN 13606. EHRFlex utiliza el toolkit GWT de Google [40] para la implementación de la interfaz de usuario y almacena los extractos de HCE en XML.

El proyecto EHRGen [157] es un generador de aplicaciones conformes a las especificaciones de openEHR. En relación al tratamiento de los arquetipos en el proceso de generación, EHRGen ofrece un mayor control a los ingenieros de software sobre el proceso. Así, por ejemplo, ofrece la posibilidad de especificar el componente gráfico que representa visualmente a un término o ajustar la visualización por medio de directivas de interfaz de usuario. Para ello implementa un mecanismo propio de plantillas que también puede ser utilizado para agregar arquetipos o para seleccionar los términos de un arquetipo que formarán parte de la visualización.

El control de la visualización por medio de directivas de presentación visual también ha sido propuesto en el proyecto GastrOS [3]. La herramienta desarrollada en este proyecto genera dinámicamente formularios a partir de arquetipos openEHR para la elaboración de informes de endoscopia digestiva.

El sistema ZK-ARCHE es otro generador automático de formularios web a partir de arquetipos ISO EN 13606 [112]. El aspecto más destacado del sistema es su integración con IHE XDS para el intercambio de extractos de HCE. Desde el punto de vista del tratamiento de arquetipos, ZK-ARCHE aplica la idea de *arquetipo exhaustivo* inspirada en la herramienta LinkEHR-Ed [123]. Un arquetipo exhaustivo resulta de la combinación de las definiciones de un arquetipo y la entidad del modelo de referencia que restringe. Según los autores del trabajo, los arquetipos *exhaustivos* permiten el procesamiento de los arquetipos sin necesidad de tratar con las definiciones del modelo de referencia.

Los sistemas y herramientas presentadas hasta este punto están orientadas a la generación de interfaces de usuario o aplicaciones completas orientadas a la captura de información clínica. Sin embargo, los arquetipos también pueden tener un papel relevante en la presentación de la información. Recientemente, los investigadores responsables del proyecto

PropeR han realizado un estudio sobre la aplicación de los principios del modelo dual a la visualización de datos clínicos [119]. Los autores señalan que la representación de los modelos de contenido (arquetipos y modelos de información) en interfaces gráficas debería especificarse sobre modelos abstractos. Esta propuesta sostiene la idea de que el conocimiento sobre la visualización de una estructura de datos debe ser responsabilidad de los expertos en el diseño de interfaces gráficas, con independencia de las tecnologías de implementación. La definición abstracta de las vistas de la aplicación es transformada automáticamente en otra representación dependiente de la plataforma. El estudio ha sido validado utilizando tres frameworks de presentación, Cocoon [194], XUL y un framework propio. A pesar de que el trabajo está enfocado a la visualización de información clínica, el principio de separación entre modelos de interfaz de usuario abstractos y específicos de la plataforma, y la automatización de la generación de las segundas a partir de las primeras, también puede ser aplicado a la generación de aplicaciones basadas en la arquitectura dual.

Por último, la generación de aplicaciones sanitarias a partir de arquetipos ha sido abordada desde la perspectiva de los sistemas heredados. La propuesta parte del estudio realizado sobre la viabilidad de la transformación de arquetipos openEHR en el modelo de información propietario del sistema Cambio COSMIC [28]. Los arquetipos transformados al formato propietario pueden ser utilizados para obtener las interfaces de usuario en el sistema de información. La idea ha sido desarrollada y generalizada en [47] donde se analizan las características que ha de soportar un sistema HCE para poder integrar la definición de arquetipos, que son establecidas en una serie de requisitos de acuerdo con el estándar ISO/TS 18308. En este trabajo también se adopta la noción de arquetipo *exhaustivo* que es transformado al modelo de datos del sistema heredado. La propuesta ha sido validada con la especificación openEHR en el sistema ArchiMed [62]. Según los autores, la extensión de la propuesta al estándar ISO EN 13606 es inmediata, especialmente porque la solución está basada en arquetipos y no en plantillas openEHR que no son estándares.

## Capítulo 4

# Representación del conocimiento

### 4.1. Introducción

La Ingeniería del Conocimiento es un área de la Inteligencia Artificial dedicada a la representación del conocimiento que manejan los agentes en los sistemas inteligentes. La representación del conocimiento consiste básicamente en describir las entidades de un dominio utilizando un lenguaje. Para que el conocimiento pueda ser procesado automáticamente y obtener conclusiones implícitas a partir del conocimiento representado explícitamente es necesario que los lenguajes de representación del conocimiento sean formales.

El primer lenguaje formal de representación del conocimiento fue la Lógica de Primer Orden. A pesar de la sólida base matemática de este lenguaje, su uso ha presentado inconvenientes tales como la sintaxis tediosa y poco intuitiva. Sin embargo, el principal inconveniente que presenta la Lógica de Primer Orden es que el problema general de razonamiento es *semidecidible*, es decir, que no se puede saber si un algoritmo encargado de determinar si un predicado es cierto finalizará para uno que no lo sea.

Para facilitar la tarea adquisición y gestión del conocimiento se han propuesto otros lenguajes que han sacrificado la expresividad de la Lógica de Primer Orden para ganar en facilidad de uso, como por ejemplo las redes semánticas [116]. Por otro lado, el problema de razonamiento se ha resuelto utilizando subconjuntos decidibles y completos de la Lógica de Primer Orden. Es el caso de las Lógicas Descriptivas [6] que han sido diseñadas para garantizar que los problemas de razonamiento finalizarán en un tiempo finito y que se alcanzarán todas las conclusiones.

Los sistemas inteligentes afrontan actualmente dos dificultades adicionales. Por una parte, la necesidad de incorporar una gran cantidad de fuentes de datos distribuidas y heterogéneas al modelo de conocimiento

de un sistema y, por otra, la necesidad de reutilizar bases de conocimiento existentes para minimizar el esfuerzo de desarrollo de una nueva aplicación.

Las ontologías se han propuesto como un formalismo de representación de conocimiento cuyo propósito es tratar todos estos problemas: son tan intuitivas como las redes semánticas, pueden tener una semántica formal sólida basada en Lógicas Descriptivas y promueven la integración de la información y la reutilización del conocimiento [69]. En este capítulo se introduce el concepto de ontología, el soporte formal proporcionado por las Lógicas Descriptivas y el lenguaje ontológico OWL. A continuación se relacionan los trabajos más relevantes que han aplicado el modelado con ontologías en las arquitecturas de HCE de modelo dual. Finalmente, se analiza el papel de las ontologías en las propuestas de interoperabilidad de modelos clínicos.

## 4.2. Ontologías

La comunidad de Inteligencia Artificial utiliza el término ontología para referirse a un modelo explícito de un dominio de conocimiento. Desde su introducción, el concepto de ontología ha sido ampliamente utilizado en otras disciplinas de la Informática donde las ontologías se están convirtiendo en un componente esencial de algunos tipos de aplicaciones. En la literatura podemos encontrar numerosas definiciones de ontología, pero sin duda la más citada es la de Gruber: una ontología es una especificación explícita de una conceptualización [68]. Esta definición ha sido revisada por otros autores ofreciendo una definición más precisa y actualizada: una ontología es una descripción formal y explícita de los conceptos de un dominio del discurso, las propiedades que describen las características y atributos de esos conceptos, y las restricciones sobre esas propiedades [140]. En la actualidad existe un amplio consenso en calificar de ontología a una conceptualización (modelo de conocimiento) si es aceptada y compartida por una comunidad, permitiendo así la reutilización del conocimiento [140].

Las ontologías son uno de los formalismos de modelado más utilizados actualmente y se consideran el formalismo de representación del conocimiento más adecuado en los sistemas inteligentes actuales [27]. Prueba de ello es el hecho de que las ontologías han sido propuestas para el soporte de la gestión de los metadatos en la Web Semántica [14, 7]. El principal beneficio del uso de ontologías es que clarifican la estructura del conocimiento del dominio que representan, ya que proporcionan un vocabulario preciso para representar el conocimiento de ese dominio. Las características que distinguen a las ontologías frente a otros formalismos son:

- Reutilización del conocimiento. Las ontologías están especialmente diseñadas para ahorrar tiempo y esfuerzo en el proceso de adquisición



de conocimiento promoviendo la combinación de modelos previos que describen partes concretas del dominio, y refinando otros modelos más generales para representar los detalles particulares del dominio.

- Facilitar la adquisición y análisis del conocimiento. El desarrollo de una ontología equivale a tener disponible una especificación formal de un dominio. Esta característica hace posible validar y verificar automáticamente el conocimiento representado, y así poder incorporarlo a un repositorio de forma fiable.

Los elementos que constituyen una ontología varían según las aplicaciones donde vayan a ser utilizadas, las necesidades del problema y el lenguaje que se utilice para definir las. En general, los componentes comunes de una ontología son:

- *Conceptos*. Los conceptos o clases representan las ideas básicas del dominio que debe ser interpretado, y determinan las categorías que clasifican los objetos del dominio. Los conceptos habitualmente son organizados en jerarquías en las que un concepto de un nivel superior se considera más general que un concepto de nivel inferior.
- *Instancias*. Las instancias o individuos son las ocurrencias concretas de un concepto.
- *Relaciones*. Las relaciones o roles representan conexiones binarias entre individuos o entre individuos y datos (cadenas, enteros, etc.)
- *Axiomas*. Los axiomas establecen restricciones sobre conceptos, instancias y relaciones, que delimitan su interpretación.

Las ontologías pueden clasificarse atendiendo al tipo de realidad que representan. A partir de la clasificación de [65] destacamos los siguientes tipos:

- Ontología de alto nivel. Describe conceptos y propiedades generales, independientes de un dominio concreto, a partir de los cuales se definen otras ontologías. Un ejemplo sería la ontología BFO (Basic Formal Ontology) [88].
- Ontología de dominio. Representa el vocabulario y las relaciones en un dominio concreto, como por ejemplo, el de la Medicina. Los conceptos y las relaciones de estas ontologías suelen ser definidos como especializaciones de otras ontologías de alto nivel. Por ejemplo, la ontología OGMS (Ontology for General Medical Science) <sup>1</sup> es una ontología para el dominio médico que ha sido definida a partir de BFO.

---

<sup>1</sup><https://code.google.com/p/ogms/>

- **Ontología de tarea.** Define el vocabulario especializado de los conceptos que se relacionan con tareas genéricas, como por ejemplo, el diagnóstico clínico. Suelen construirse utilizando el conocimiento proporcionado por las ontologías de alto nivel o por las ontologías de dominio.
- **Ontología de aplicación.** Está formada por las definiciones necesarias para representar el conocimiento requerido por una aplicación. Las definiciones suelen reutilizar los conceptos de ontologías de dominio o de tareas. A diferencia de las categorías anteriores, estas ontologías son dependientes de la aplicación a la que están destinadas, por lo que apenas son reutilizables.

Bajo el principio de reutilización del conocimiento consensuado se definen dos operaciones básicas sobre ontologías que facilitan la agregación de conocimiento: alineamiento y unión de ontologías. Estas operaciones son de utilidad en tareas de mediación de datos entre fuentes heterogéneas facilitando la resolución de conflictos:

- **Alineamiento de ontologías:** es el proceso de búsqueda de correspondencias entre entidades semánticamente relacionadas en dos ontologías.
- **Unión de ontologías:** es la operación que consiste en la creación de una ontología tomando como fuente dos o más ontologías. La nueva ontología puede considerarse un puente entre las ontologías originales y contiene las correspondencias expresadas mediante axiomas.

La construcción de ontologías es un conjunto de actividades que incluyen las etapas de desarrollo de una ontología, la gestión del ciclo de vida, herramientas de soporte y los lenguajes ontológicos aplicados [65]. En las dos últimas décadas se han propuesto varias metodologías para la creación de ontologías. Hasta mediados de los 90, el proceso de desarrollo de ontologías fue más un arte que una actividad ingenieril donde habitualmente cada equipo de desarrollo seguía sus propios criterios y fases para la construcción manual de ontologías [61]. A partir de entonces se publicaron propuestas metodológicas como METHONTOLOGY [60] y On-To-Knowledge [179]. Más tarde surgieron metodologías para la construcción cooperativa de ontologías [57] y en 2004 se propuso la metodología DILIGENT [158].

En los últimos años los procesos de modelado de ontologías han comenzado a reutilizar otras ontologías y otros recursos de conocimiento no ontológicos que han sido construidos con alto grado de consenso expresados como tesauros, esquemas XML, modelos UML o sistemas terminológicos [101]. Asimismo, recientemente ha surgido interés por la aplicación

de patrones de diseño en ontologías [159]. Sin embargo, el desarrollo de ontologías está aún en su infancia si es comparado con los métodos aplicados en Ingeniería del software.

#### 4.2.1. Alineamiento de ontologías

Dado el interés que para esta tesis tienen los proceso de integración ontológica, en este apartado se presenta en mayor detalle la tarea de alineamiento o *matching* ontológico [54]. El problema que trata el proceso de alineamiento ontológico es el siguiente: dadas dos ontologías que describen un conjunto de entidades, encontrar las relaciones que se mantienen entre ellas, como por ejemplo, relaciones de equivalencia o subsunción. El resultado de este proceso es un conjunto de correspondencias que pueden utilizarse en tareas como la unión de ontologías, consultas o la transformación de datos. Por lo tanto, el alineamiento ontológico permite la interoperabilidad del conocimiento y de los datos expresados en distintas ontologías.

En los últimos años se han propuesto varios algoritmos de alineamiento ontológico que han sido presentados a “Ontology Alignment Evaluation Initiative” [53]. Entre ellos, los que han mostrado mejores resultados han sido: H-Match [25], Falcon-AO [85] y o AgreementMaker [37] En [54] se presenta una clasificación de técnicas de alineamiento ontológico que puede resumirse en:

- Técnicas terminológicas. El emparejamiento está basado en el texto que identifica a las entidades (etiquetas) así como definiciones textuales como los comentarios. Se aplican técnicas de procesamiento del lenguaje natural y recuperación de la información.
- Técnicas estructurales. Se utilizan las relaciones entre las entidades de la ontología, como las relaciones jerárquicas o mereológicas.
- Técnicas basadas en ontologías de alto nivel. El uso de ontologías de alto nivel aporta una semántica compartida que puede ser aprovechada para la identificación de correspondencias.

Hoy en día los sistemas de *matching* ontológico no son aplicables a grandes ontologías debido a que se pierde precisión en los resultados conforme las ontologías crecen en tamaño, complejidad y heterogeneidad. Por este motivo el proceso de alineamiento no suele aplicarse de forma totalmente automática, sino que los usuarios interactúan en el proceso para controlar y mejorar la calidad del alineamiento. A continuación se presentan algunas herramientas destacadas que asisten en esta tarea:

- Prompt [141] es un entorno de trabajo que ofrece distintas técnicas de alineamiento y permite la visualización de las correspondencias. Está integrado en el entorno de edición de ontologías Protégé [142].

- Foam [49] es un marco de trabajo cuyo objetivo es la combinación de algoritmos de alineamiento. También puede ser integrado en los entornos de edición de ontologías como Protégé y KAON [23].
- Alignment Server. Esta herramienta está relacionada con Alignment API [39] que ofrece soporte para la ejecución de procesos de alineamiento y para la gestión y compartición de correspondencias.
- Web Service Modeling Toolkit (WSMT) [106] es un entorno de desarrollo para servicios web semánticos que proporciona funcionalidad para la gestión de ontologías, y en concreto, para la creación, edición y almacenamiento de alineamientos entre ontologías. Las correspondencias son representadas en formato AML (Abstract Mapping Language) [171], que es una sintaxis neutral para expresar alineamientos ontológicos. Asimismo, ofrece métodos y técnicas que asisten a los expertos del dominio como distintas vistas gráficas de las ontologías, sugerencias sobre entidades relacionadas, y en general proporciona ayuda a través del proceso de alineamiento [137].

Las correspondencias entre ontologías pueden ser expresadas de dos modos diferentes, utilizando axiomas de un lenguaje ontológico como OWL, o mediante un lenguaje específico para la definición de alineamientos. El uso de un lenguaje ontológico tiene la ventaja de que un alineamiento puede ser visto como una tercera ontología que importa las ontologías origen y destino y que contiene un conjunto de axiomas o reglas que se relacionan los términos de las ontologías importadas. Sin embargo, los lenguajes específicos para especificar alineamientos son más intuitivos que un lenguaje ontológico y, en general, las correspondencias pueden ser transformadas al lenguaje OWL. Los lenguajes de alineamiento más importantes son Alignment API [52], AML [171] y una extensión de Alignment API denominada EDOAL (Expressive and Declarative Ontology Alignment Language) [39].

#### 4.2.2. Ontologías en Ingeniería de Modelos

La semántica de los modelos tiene especial importancia en el desarrollo de la iniciativa MDA. La especificación MDA adopta el lenguaje UML como lenguaje de modelado cuya semántica no es completamente formal y en ciertos casos ambigua. Disponer de modelos UML con una semántica formal permite aplicar razonamiento para comprobar la consistencia del modelo, encontrar conocimiento implícito y, especialmente, tener la posibilidad de tratar con conocimiento incompleto, lo que es de interés en las actividades de modelado de alto nivel de abstracción.

OMG ha advertido la relevancia que para la plena realización de su propuesta MDA puede tener el disponer de un substrato formal basado en ontologías. Con este fin ha definido la especificación ODM [149] como respuesta a las necesidades de formalización en el ámbito de MDA, de modo que puedan superarse los inconvenientes para el modelado que presenta el lenguaje UML. ODM define metamodelos MOF para un conjunto de lenguajes formales de representación de conocimiento, entre los que se encuentra OWL. Asimismo, define las correspondencias entre estos lenguajes y OWL, lo que permite la interoperabilidad entre modelos expresados en esos lenguajes. En definitiva, la especificación ODM establece el puente entre el espacio tecnológico de los modelos y la web semántica.

Las correspondencias entre UML y los lenguajes de representación de conocimiento tienen carácter informativo. De este modo, las correspondencias pueden adaptarse a las necesidades concretas del dominio en el que van a ser utilizadas. Además, debe tenerse en cuenta que las correspondencias no son completas, es decir, existen casos en los que los elementos de un metamodelo no tienen una representación que preserve la semántica de las construcciones del metamodelo destino. Por ejemplo, las relaciones n-arias de UML no encuentran correspondencia en OWL [55].

### 4.3. Lógicas Descriptivas

Las lógicas descriptivas son una familia de lenguajes de representación del conocimiento que se caracterizan por la facilidad de uso como lenguaje de modelado y por su semántica formalmente definida [6]. La base formal de estos lenguajes corresponde con fragmentos de la Lógica de Primer Orden que determinan su expresividad. La expresividad de una lógica descriptiva equivale a la semántica de las expresiones que pueden construirse con ella. Por este motivo, las lógicas descriptivas se estructuran en niveles de expresividad que tienen asociados una cadena de letras mayúsculas que denotan las construcciones permitidas. Por lo tanto, disponer de más constructores en el lenguaje significa que es más expresivo, y en consecuencia, la complejidad computacional es mayor.

#### 4.3.1. Definiciones

Los símbolos utilizados en una lógica descriptiva se denominan *signatura* o vocabulario. Formalmente, la signatura es la unión disjunta  $\mathbf{S} = \mathbf{C} \uplus \mathbf{R} \uplus \mathbf{I}$  donde  $\mathbf{C} = \{A\}$  es el conjunto de conceptos atómicos o clases,  $\mathbf{R} = \{R_A\}$  el conjunto de roles atómicos o propiedades, e  $\mathbf{I} = \{a, b, \dots\}$  es el conjunto de individuos o instancias.

El vocabulario de una lógica descriptiva puede extenderse utilizando constructores para definir conceptos complejos  $\text{Con}(\mathbf{S}) = \{C_{(i)}, D_{(i)}, \dots\}$ , roles complejos  $\text{Rol}(\mathbf{S}) = \{R_{(i)}\}$  y axiomas  $\text{Ax}(\mathbf{S}) = \{O_{(i)}\}$ . En la notación empleada los subíndices pueden omitirse cuando no haya ambigüedad con roles y conceptos atómicos. La signatura de un concepto complejo  $\mathbf{S}(C_{(i)})$  se define como el conjunto de elementos de  $\mathbf{S}$  que están incluidos en  $C_{(i)}$ . La signatura de roles complejos y axiomas se definiría de forma análoga.

Los *axiomas* básicos de cualquier lógica descriptiva son la inclusión de conceptos ( $D \sqsubseteq C$ ), los axiomas de pertenencia de instancias ( $a : C$ ) y axiomas de pertenencia a roles ( $(a, b) : R$ ).

En lógica descriptiva, una TBox  $\mathcal{T}$  se define como un conjunto finito de axiomas de inclusión general de conceptos (GCI) de la forma  $D \sqsubseteq C$ , lo que implica que el concepto  $D$  es más específico que  $C$ , o que  $C$  subsume a  $D$ . Una definición de concepto  $C \equiv D$  ( $C$  y  $D$  son equivalentes) es una abreviación de la pareja de axiomas  $C \sqsubseteq D$  y  $D \sqsubseteq C$ . Una RBox  $\mathcal{R}$  consiste en un conjunto finito de axiomas de roles estableciendo propiedades de roles tales como la inclusión, transitividad, funcionalidad, etc. (véase tabla 4.4). Una ABox  $\mathcal{A}$  consiste en un conjunto finito de axiomas sobre individuos.

Una *ontología* o base de conocimiento en lógica descriptiva se define como la tripleta  $\mathcal{K} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$ , donde  $\mathcal{T}$  es una TBox,  $\mathcal{R}$  es una RBox y  $\mathcal{A}$  es una ABox. La signatura de una ontología  $\mathbf{S}(\mathcal{K})$  es la unión de todas las signaturas  $\mathbf{S}(O)$  de los axiomas de  $\mathcal{K}$ .

La *interpretación*  $\mathcal{I}$  de una ontología  $\mathcal{K}$  es una pareja  $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  donde  $\Delta^{\mathcal{I}}$  es un conjunto no vacío que se denomina *dominio de la interpretación*, y  $\cdot^{\mathcal{I}}$  es una función que mapea:

- Cada individuo  $a$  en  $\mathcal{K}$  con un elemento  $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$ .
- Cada concepto  $C$  en  $\mathcal{K}$  con un subconjunto  $C^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ .
- Cada rol  $R$  en  $\mathcal{K}$  con un subconjunto  $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ .

Se dice que una interpretación  $\mathcal{I}$  es un *modelo* de:

- $a : C$  sii  $a^{\mathcal{I}} \in C^{\mathcal{I}}$ ,
- $(a, b) : R$  sii  $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$ ,
- $C \sqsubseteq D$  sii  $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ ,
- una base de conocimiento  $\mathcal{K} = \langle \mathcal{T}, \mathcal{R}, \mathcal{A} \rangle$  sii es un modelo para cada elemento en  $\mathcal{T}$ ,  $\mathcal{R}$  y  $\mathcal{A}$ .

### 4.3.2. Lógicas Descriptivas Básicas

En este apartado se presenta un grupo de lógicas descriptivas que se utilizan como base para presentar otras lógicas más avanzadas. En primer lugar se presenta la lógica descriptiva  $\mathcal{AL}$ . La sintaxis y la semántica de los constructores de este lenguaje se muestra en la tabla 4.1. En  $\mathcal{AL}$  sólo se admiten roles atómicos, se pueden emplear dos conceptos especiales, el concepto superior ( $\top$ ), que representa a todos los individuos, y el concepto inferior ( $\perp$ ) que es el conjunto vacío, y la negación de conceptos sólo es aplicable a conceptos atómicos.

Constructor	Sintaxis	Semántica
<i>Constructores de conceptos</i>		
Concepto superior	$\top$	$\Delta^I$
Concepto inferior	$\perp$	$\emptyset$
Concepto atómico	$A$	$A^I \subseteq \Delta^I$
Negación de concepto atómico	$\neg A$	$\Delta^I \setminus A^I$
Intersección de conceptos	$C \sqcap D$	$C^I \cap D^I$
Cuantificación universal	$\forall R.C$	$\{x : \forall y, (x, y) \in R^I \Rightarrow y \in C^I\}$
Cuantificación existencial completa	$\exists R.\top$	$\{x : \exists y, (x, y) \in R^I\}$
<i>Constructores de roles</i>		
Rol atómico	$R_A$	$R_A^I \subseteq \Delta^I \times \Delta^I$

Tabla 4.1: Constructores de la lógica descriptiva  $\mathcal{AL}$

La lógica descriptiva  $\mathcal{ALC}$  extiende  $\mathcal{AL}$  con el constructor de negación general de conceptos (atómicos y complejos). Esta nueva construcción aumenta la expresividad del lenguaje puesto que  $C \cup D$  es equivalente a  $\neg(\neg C \sqcap \neg D)$  y  $\exists R.C$  equivale a  $\neg(\forall(\neg R.C))$  (véase tabla 4.2). Por tanto, la unión de conceptos y la restricción existencial pueden ser expresadas en  $\mathcal{ALC}$ , es decir, tiene expresividad  $\mathcal{UE}$ .

Constructor	Sintaxis	Semántica
Negación de concepto ( $C$ )	$\neg C$	$\Delta^I \setminus C^I$
Unión de conceptos ( $\mathcal{U}$ )	$C \sqcup D$	$C^I \cup D^I$
Cuantificación existencial ( $\mathcal{E}$ )	$\exists R.C$	$\{x : \exists y, (x, y) \in R^I \wedge y \in C^I\}$

Tabla 4.2: Extensión  $\mathcal{ALC}$  de la lógica descriptiva  $\mathcal{AL}$

### 4.3.3. Otras Lógicas Descriptivas

En la literatura se han definido varias lógicas descriptivas con diferentes niveles de expresividad. En este apartado se ofrece una visión general de la sintaxis y semántica de algunas extensiones de  $\mathcal{ALC}$ . Distintas combinaciones de extensiones pueden conducir al mismo lenguaje dado que algunas de ellas son mutuamente reducibles.

En primer lugar, se añaden constructores de conceptos y roles que extienden la expresividad de  $\mathcal{ALC}$ . La sintaxis y semántica de los nuevos constructores se muestra en la tabla 4.3. En las construcciones,  $S$  denota un rol simple que puede ser un rol atómico, la inversa de un rol simple, o un rol que sólo subsume roles simples y el operador  $\circ$  denota la composición de relaciones binarias. En segundo lugar, la tabla 4.4 presenta axiomas avanzados sobre roles e instancias donde  $\Delta_D$  denota un tipo de datos predefinido tal como `string`, `integer`, etc.

Constructor	Sintaxis	Semántica
<i>Constructores de conceptos</i>		
Nominales ( $\mathcal{O}$ )	$\{a, b, \dots\}$	$\{a^I, b^I, \dots\}$
Restricción numérica no cualificada ( $\mathcal{N}$ )	$\geq nR$ $\leq mR$	$\{x :  \{y : (x, y) \in R^I\}  \geq n\}$ $\{x :  \{y : (x, y) \in R^I\}  \leq m\}$
Restricción de cardinalidad cualificada ( $\mathcal{Q}$ )	$\geq nR.C$ $\leq mR.C$	$\{x :  \{y : (x, y) \in R^I \wedge y \in C^I\}  \geq n\}$ $\{x :  \{y : (x, y) \in R^I \wedge y \in C^I\}  \leq m\}$
Restricción de reflexividad ( $\mathcal{S}$ )	$\exists S.Self$	$\{x : (x, x) \in R^I\}$
<i>Constructores de roles</i>		
Rol inverso ( $\mathcal{I}$ )	$R^-$	$\{(y, x) \in \Delta^I \times \Delta^I \mid (x, y) \in R^I\}$
Cadena de roles	$R_1 R_2 \dots R_n$	$R_1^I \circ R_2^I \circ \dots \circ R_n^I$
Rol concreto ( $\mathcal{D}$ )	$T$	$T^I \subseteq \Delta^I \times \Delta_D$
Rol Universal	$U$	$\Delta^I \times \Delta^I$

Tabla 4.3: Constructores adicionales a lógica descriptiva  $\mathcal{ALC}$

A continuación se describen algunas de las lógicas descriptivas más utilizadas actualmente:

- $\mathcal{EL}$ , equivale a  $\mathcal{AL}$  sin el cuantificador universal ni el concepto inferior. Así pues, proporciona como constructores de conceptos el concepto superior, conjunción y la restricción existencial completa, y la equivalencia de conceptos como único axioma [4].
- $\mathcal{EL}++$  es una extensión de  $\mathcal{EL}$  que añade el concepto inferior, nominales, tipos de datos y GCI [5]. Es la lógica subyacente al perfil EL de OWL 2.
- $\mathcal{SH}$  extiende  $\mathcal{ALC}$  con roles transitivos y jerarquías de conceptos [84].



Constructor	Sintaxis	Semántica
<i>Axiomas de roles</i>		
Roles funcionales ( $\mathcal{F}$ )	Func( $R$ )	$(a^I, b^I) \in R^I \wedge (a^I, c^I) \in R^I \Rightarrow b^I = c^I$
Roles transitivos ( $\mathcal{S}$ )	Trans( $R$ )	$R^I$ es transitivo
Roles disjuntos	Dis( $S, S'$ )	$S^I \cap S'^I = \emptyset$
Roles reflexivos (s)	Ref( $R$ )	$(x, x) \in R^I, \forall x \in \Delta^I$
Roles irreflexivos (s)	Irr( $S$ )	$(x, x) \notin S^I, \forall x \in \Delta^I$
Roles simétricos	Sym( $R$ )	$(x, y) \in R^I \Rightarrow (y, x) \in R^I$
Roles asimétricos	Asy( $S$ )	$(x, y) \in S^I \Rightarrow (y, x) \notin S^I$
Jerarquías de roles ( $\mathcal{H}$ )	$R_1 \subseteq R_2$	$R_1^I \subseteq R_2^I$
<i>Axiomas de instancia</i>		
Aserciones negación de roles	$(a, b) : \neg R$	$(a^I, b^I) \notin R^I$
Aserciones de no igualdad	$a \neq b$	$a^I \neq b^I$
Aserciones de igualdad	$a = b$	$a^I = b^I$

Tabla 4.4: Axiomas adicionales a lógica descriptiva  $\mathcal{ALC}$ 

- $\mathcal{SHIF}$  extiende  $\mathcal{SH}$  con roles inversos y funcionales [82]. Esta lógica corresponde con el nivel *Lite* de OWL.
- $\mathcal{SHOIN}(D)$  extiende  $\mathcal{SH}$  con nominales, roles inversos, restricciones de cardinalidad y tipos de datos [82]. Esta lógica equivale al nivel DL de OWL.
- $\mathcal{SROIQ}(D)$  extiende  $\mathcal{SHOIN}(D)$  con restricciones de cardinalidad cualificadas, roles disjuntos, roles reflexivos e irreflexivos, cadenas de roles, inclusión de roles complejos, el rol universal, reflexividad local de los conceptos, asertos sobre negación de roles y asertos sobre igualdad y desigualdad [81]. Esta lógica corresponde con el nivel DL de OWL 2.

#### 4.3.4. Razonamiento con lógicas descriptivas

El razonamiento con una base de conocimiento es el procedimiento automático para inferir nuevos axiomas a partir de los axiomas ya representados. Habitualmente el razonamiento en lógica descriptiva puede llevarse a cabo con los conceptos en la TBox, los individuos en la ABox, o conjuntamente con ambas.

En lógica descriptiva, la tarea básica de razonamiento con los conceptos es determinar si pueden ser *satisfechos*. Intuitivamente, un concepto puede ser satisfecho (consistente) si no entra en contradicción con el resto de conocimiento en la ontología. Otra tarea importante es la subsunción de conceptos, que infiere si un concepto es más general que otro. La prueba de

equivalencia de conceptos, que determina si dos conceptos son el mismo, y la disyunción de conceptos, que determina si dos conceptos tienen individuos en común, son extensiones inmediatas del chequeo de la subsunción de conceptos.

Formalmente, las tareas de inferencia sobre conceptos se definen del siguiente modo:

- Un concepto  $C$  es consistente con respecto a una base de conocimiento  $\mathcal{K}$  si existe algún modelo  $\mathcal{I}$  de  $\mathcal{K}$  tal que  $C^{\mathcal{I}}$  es no vacío. Por extensión, una TBox  $\mathcal{T}$  es consistente si cada axioma en  $\mathcal{T}$  es consistente.
- Un concepto  $D$  es subsumido por un concepto  $C$  con respecto a una base de conocimiento  $\mathcal{K}$  si cada modelo  $\mathcal{I}$  de  $\mathcal{K}$  es un modelo de  $D \sqsubseteq C$ . Esto es denotado como  $\mathcal{K} \models D \sqsubseteq C$  ( $\mathcal{K}$  implica  $D \sqsubseteq C$ ).
- Dos conceptos  $C$  y  $D$  son equivalentes con respecto a una base de conocimiento  $\mathcal{K}$  si  $C$  es subsumido por  $D$  con respecto a  $\mathcal{K}$  y  $D$  es subsumido por  $C$  con respecto a  $\mathcal{K}$ . Esto es denotado como  $\mathcal{K} \models C \equiv D$  ( $\mathcal{K}$  implica  $C$  es equivalente a  $D$ ).
- Dos conceptos  $C$  y  $D$  son disjuntos con respecto a una base de conocimiento  $\mathcal{K}$  si  $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$  para cada modelo  $\mathcal{I}$  de  $\mathcal{K}$ .

En relación al razonamiento con individuos, la tarea de inferencia básica es comprobar si un aserto de la ABox no entra en contradicción con el resto de axiomas de la ontología. Otras posibles consultas son chequear si ciertas relaciones entre conceptos, roles e individuos se mantienen. Formalmente, las tareas básicas de inferencia con instancias se definen del siguiente modo:

- Un axioma de instancias  $O$  es consistente con respecto a una base de conocimiento  $\mathcal{K}$  si existe al menos una interpretación  $\mathcal{I}$  que sea un modelo de  $O$  y  $\mathcal{K}$ . Se dice que una ABox  $\mathcal{A}$  es consistente con respecto a  $\mathcal{K}$  si cada axioma de  $\mathcal{A}$  es consistente con respecto a  $\mathcal{K}$ .
- Un axioma de instancias  $O$  se dice que es implicado por una ABox  $\mathcal{A}$  si cada modelo  $\mathcal{I}$  de  $\mathcal{A}$  es también un modelo de  $O$ . Se denota como  $\mathcal{A} \models O$  ( $\mathcal{K}$  implica  $O$ ). Esta prueba puede ser extendida para ser realizada con respecto a una base de conocimiento  $\mathcal{K}$ . Si  $O$  es un axioma de pertenencia ( $a : C$ ), esta prueba se denomina *chequeo de instancia*.

En relación al razonamiento en una ABox en lógica descriptiva es importante señalar que se mantiene la *asunción de mundo abierto*. El razonamiento en mundo abierto debe entenderse como que el conjunto de axiomas en una base de conocimiento no es completo, y por lo tanto, no se puede deducir conocimiento inductivamente. Por contra, la asunción de mundo cerrado

supone que el conjunto de axiomas es completo y que las inducciones pueden realizarse de forma segura.

El conjunto de servicios de razonamiento puede ser extendido utilizando las tareas básicas de inferencia comentadas anteriormente. Por ejemplo, es habitual que los razonadores implementen un procedimiento de clasificación que encuentra la situación de un concepto en una jerarquía, es decir, las subclases y superclases directas.

Los algoritmos de razonamiento con lógicas descriptivas más comunes son los basados en *tableau*. La complejidad de los procedimientos que utilizan estos algoritmos depende de la expresividad del lenguaje considerado, que es alta incluso para las lógicas descriptivas más básicas. Afortunadamente, las inferencias en el peor caso son infrecuentes, y los procedimientos han sido altamente optimizados para ofrecer buenos tiempos de ejecución en los casos más frecuentes. Los razonadores que implementan esta técnica son Fact++ [190], Pellet [176] y RacerPro [70]. Para resolver las limitaciones de razonamiento de los algoritmos basados en *tableau*, se han propuesto alternativas como el sistema *hipertableau* de Hermit [175] o el algoritmo de resolución empleado en KAON2 [139] para tratar con grandes volúmenes de datos.

## 4.4. OWL

OWL (Web Ontology Language) es una familia de lenguajes de representación de conocimiento que se ha convertido actualmente en la opción más utilizada para definir ontologías en la Web Semántica. OWL está influenciado por el formalismo de las Lógicas Descriptivas, por lo que una ontología expresada en OWL contiene descripciones de clases, roles e individuos [83]. OWL comparte características con RDF, tales como el uso de URI para referenciar las entidades del dominio y el formato de serialización, mejora la expresividad de RDFS, introduce construcciones de metamodelado (*punning*), incluye claves al estilo de las bases de datos relacionales y soporta la mayoría de tipos de datos de XML Schema. Además, las ontologías OWL pueden organizarse modularmente en varios ficheros. OWL 1.0 es una recomendación del W3C desde 2004 [133] y en 2009 apareció la versión 2.0 [138].

La especificación OWL 1.0 tiene tres dialectos, cada uno de ellos con distintas propiedades computacionales:

- *OWL Lite* es el nivel menos expresivo y corresponde con la lógica descriptiva  $\mathcal{SHIF}(D)$ . Este dialecto de OWL es recomendado para la creación de tesauros y taxonomías sencillas. El razonamiento con *OWL Lite* es el más eficiente de los tres niveles.

- *OWL DL* es más expresivo que *OWL Lite* siendo computacionalmente completo y decidible. *OWL DL* es equivalente a lógica descriptiva *SHOIN(D)*.
- *OWL Full* es el dialecto más expresivo de OWL, aunque no es decidible. 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*.

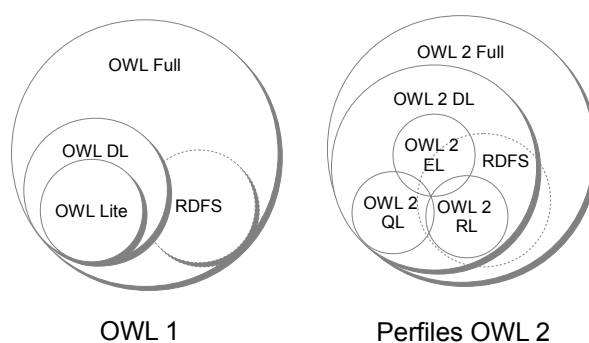


Figura 4.1: Relación entre los lenguajes y perfiles OWL

En la versión 2.0 de OWL, el dialecto DL ha sido enriquecido hasta alcanzar la expresividad de la lógica descriptiva *SROIQ(D)*. En esta versión se ha incluido el concepto de *perfil* [138] que hace referencia a las características de interés para el uso de OWL en cierto tipo de aplicaciones. En la figura 4.1 se muestra gráficamente una comparación entre las dos versiones de OWL [77]. Los perfiles que introduce OWL 2 son los siguientes:

- *OWL 2 EL*: es adecuado en aplicaciones que necesiten ontologías muy grandes, es decir, ontologías con un gran número de clases y propiedades. Este perfil corresponde aproximadamente con la lógica descriptiva  $\mathcal{EL}++$  que garantiza un tiempo polinomial para la resolución de problemas de razonamiento. SNOMED-CT es expresable en *OWL 2 EL*.
- *OWL 2 RL*: está diseñado para las aplicaciones que requieren razonamiento escalable sin sacrificar demasiada expresividad. Está orientado fundamentalmente a facilitar el razonamiento basado en reglas.

- *OWL 2 QL*: está destinado principalmente para aplicaciones que requieran responder a consultas de un modo formal y completo para un gran volumen de datos, a un coste computacional razonable. Asimismo, puede ser utilizada en tareas de integración en bases de datos debido a que proporciona las construcciones necesarias para expresar modelos conceptuales como diagramas de clases en UML o diagramas Entidad/Relación.

La sintaxis OWL puede ser expresada en diferentes formatos (RDF/XML, OWL/XML, Turtle, Manchester). La sintaxis normativa de OWL está basada en la sintaxis XML de RDF. En esta tesis utilizaremos la sintaxis Manchester [80] por ser más legible. En la tabla 4.5 se presenta la notación OWL Manchester para los constructores más relevantes de la lógica descriptiva. Además, esta notación incluye palabras clave para declarar clases, propiedades, anotaciones, etc. En el siguiente ejemplo se presenta la propiedad de objetos *tieneHijo* que declara el dominio y rango de la clase *Persona* y se define asimétrica y no reflexiva, la clase *PersonaMayorDeEdad* que utiliza una faceta sobre el tipo de datos entero para representar las personas con más de 18 años, y la clase *Abuelo*. Para referenciar entidades de otras ontologías se utilizan prefijos. En el ejemplo, la clase *Persona* pertenece a una ontología referida mediante el prefijo “familia”.

---

```
ObjectProperty: tieneHijo
  Domain: familia:Persona
  Range: familia:Persona
  Characteristics: Asymmetric, Irreflexive

Class: PersonaMayorDeEdad
  SubClassOf:
    familia:Persona,
    edad some int[>= 18]

Class: Abuelo
  Annotations: rdfs:comment 'Abuelo es una persona que tiene algún nieto'
  EquivalentTo:
    familia:Persona and tieneHijo some (tieneHijo some familia:Persona)
```

---

Constructor	Sintaxis DL	Notación Manchester
<i>Constructores de conceptos</i>		
Negación de conceptos	$\neg C$	<b>not</b> C
Intersección de conceptos	$C \sqcap D$	C <b>and</b> D
Unión de conceptos	$C \sqcup D$	C <b>or</b> D
<i>Restricciones de propiedades</i>		
Cuantificación universal	$\forall R.C$	R <b>only</b> C
Cuantificación existencial	$\exists R.C$	R <b>some</b> C
Valor nominal	$R : o$	R <b>value</b> o
Restricción numérica no cualificada	$\geq nR$ $\leq mR$ $= mR$	R <b>min</b> n R <b>max</b> m R <b>exactly</b> m
Restricción de cardinalidad cualificada	$\geq nR.C$	R <b>max</b> m C

Tabla 4.5: Ejemplos de construcciones en notación OWL Manchester

## 4.5. Modelado ontológico en la arquitectura dual

El lenguaje OWL es aceptado actualmente como un formalismo para la representación de modelos de contenido clínico [31]. Este reconocimiento ha sido obtenido principalmente gracias a varios proyectos de investigación que han promovido el uso de tecnologías semánticas en el ámbito sanitario, entre los que destaca el proyecto Artemis [42].

La representación en OWL de los modelos clínicos en las arquitecturas duales requiere que los modelos de referencia también sean expresados en este lenguaje. Así pues, esta sección se organiza en dos partes. En primer lugar, se presentan trabajos enfocados en la definición ontológica de los modelos de referencia en estándares de modelo dual. Seguidamente se analizan las propuestas de aplicación de ontologías OWL a modelos clínicos, en especial, las representaciones basadas en arquetipos.

### 4.5.1. Ontologías de los modelos de referencia

El proyecto de investigación que ha situado a OWL en el mapa de las tecnologías aplicadas al ámbito sanitario ha sido Artemis [42]. La motivación de este proyecto europeo fue la definición de una arquitectura de interoperabilidad semántica para sistemas de información sanitarios. La propuesta se articuló en torno al uso de un mediador encargado de la conversión de la información clínica entre los sistemas participantes, de modo que las instituciones sanitarias pudieran integrarse en la arquitectura Artemis manteniendo sus sistemas de información. En este proyecto las ontologías han sido aplicadas a la descripción de los servicios sanitarios, la información clí-

nica y para establecer las correspondencias semánticas entre las estructuras de datos clínicas.

Los modelos de información en Artemis son representados en OWL. En concreto, los modelos de referencia de las arquitecturas duales han sido representados de tal modo que se se pueden obtener automáticamente parte de las correspondencias entre distintos estándares. Para ello se eligió el modelo genérico HL7 RIM como ontología superior de la que fueron derivados los modelos de referencia siguiendo el proceso descrito en el estándar [110]. Las reglas de derivación se implementaron como axiomas en OWL, lo que permite descubrir el origen ontológico de cualquier entidad de los modelo de información. Por tanto, dos propiedades son alineadas si han sido obtenidas del mismo atributo de HL7 RIM. Esta estrategia fue aplicada a la interoperabilidad entre documentos HL7 CDA e ISO EN 13606, gracias a que éste último también ha sido definido como un modelo de información que puede ser derivado de HL7 RIM [45].

En HL7 ha habido otras propuestas para la definición ontológica de HL7 versión 3. La primera aproximación fue realizada en la Universidad de Stanford y se limitó a modelar las clases de alto nivel y tipos de datos de HL7 RIM [191]. Este trabajo fue completado en 2006 a través de una ontología en RDFS [155]. En 2008, el grupo de interés del W3C *Health and Life Sciences* publicó otra ontología de HL7 versión 3 en formato OWL [26]. La diferencia entre estas dos últimas ontologías reside en la versión representada de HL7 RIM, y la metodología y el lenguaje de modelado aplicados. Recientemente se ha publicado otra nueva propuesta que utiliza OWL DL para representar la versión actual de HL7 RIM. Los autores afirman que esta ontología soporta la instanciación de cualquier mensaje y documento del estándar y ha sido probada como soporte para la implementación de reglas clínicas [91]. Por último, en [167] se ha propuesto una metodología para la construcción de ontologías para las dos versiones del estándar HL7 y se ha implementado una herramienta que asiste en este proceso.

El estándar ISO EN 13606 y la especificación openEHR también han sido expresados con ontologías. En el marco de trabajo *Poseacle* se ha propuesto una arquitectura ontológica para modelos de información de arquitecturas duales obtenida a partir de la interpretación semántica de los modelos de información de los estándares ISO EN 13606 y openEHR [59]. Las ontologías obtenidas representan modelos de referencia *arquetipables* cuyas instancias definen arquetipos. Esta propuesta se describe con más detalle en la sección 5.3 de la memoria. Por otro lado, en [164] se ha propuesto una ontología para el modelo de referencia de openEHR que, a diferencia de las ontologías *Poseacle*, no se integra semánticamente con el modelo de arquetipos.

### 4.5.2. Ontologías aplicadas a modelos clínicos

La primera propuesta de representación de arquetipos en OWL fue desarrollada en el marco del proyecto Artemis [109]. La definición más precisa de las correspondencias entre modelos de información sanitarios motivó la introducción de arquetipos en la arquitectura de interoperabilidad semántica de Artemis. Los arquetipos definen conceptos clínicos con una semántica más específica que las estructuras de datos de los modelos de información. Así pues, la definición en OWL de los arquetipos esta justificada por la propia arquitectura de interoperabilidad en la que los modelos de referencia también han sido definidos en OWL con el propósito de establecer correspondencias a nivel ontológico y obtener parte de ellas a partir de un proceso de inferencia cuando los modelos de información son derivados de HL7 RIM.

La aproximación adoptada en Artemis se inspira en las recomendaciones de formalización de plantillas en HL7 CDA [71]. En primer lugar, todos los nodos objeto del arquetipo (términos) se declaran como clases en OWL. Las restricciones de atributos que se definen en cada término se traducen en axiomas OWL que se integran en la definición de la clase. Por tanto, un atributo restringido no da lugar a nueva propiedad en OWL. En segundo lugar, se establecen las correspondencias entre restricciones ADL y axiomas OWL. Sin embargo, son pocas las restricciones de atributos que son traducidas a OWL. Un ejemplo es la cardinalidad de un atributo que corresponde con un axioma de cardinalidad en OWL. En tercer lugar, la reutilización de términos a través de las referencias internas no implica ninguna construcción adicional en OWL, ya que todos los términos están representados por una clase. Por último, la composición de arquetipos ofrecida por la construcción *slot* se expresa como una propiedad en OWL que asocia términos con arquetipos. La restricción que define el *slot* se establece sobre el identificador del arquetipo.

Los arquetipos representados en OWL son utilizados en el proyecto Artemis para establecer correspondencias entre modelos clínicos. El lenguaje OWL ofrece la posibilidad de utilizar herramientas de alineamiento ontológico y aplicar inferencia para el descubrimiento de las correspondencias. Sin embargo, el principal inconveniente de la representación propuesta es que la semántica de los arquetipos no es definida de forma completa. Por un lado, las restricciones de tipos primitivos no son expresadas en OWL, como por ejemplo un intervalo de enteros. El motivo de esta limitación es que el lenguaje OWL no ha soportado axiomas sobre el dominio de los tipos primitivos hasta la versión 2 que fue publicada en 2009. Además, esto implica que la relación *slot* tampoco sea soportada adecuadamente por la representación. Por otro lado, la primera versión de OWL tampoco tenía expresividad suficiente para representar algunos tipos de restricciones como las ocurrencias de un término. Finalmente, la propuesta de Artemis



ignora las restricciones de orden y unicidad asociadas a la cardinalidad de atributos multivaluados.

La aparición de OWL 2 ha permitido completar la semántica del modelo de arquetipos. En [118] se motiva la representación de arquetipos en OWL como soporte para la definición de guías clínicas. Este trabajo extiende la propuesta de Artemis incluyendo la restricción de ocurrencias expresada con axiomas de cardinalidad cualificada en OWL. Sin embargo, cabe destacar que no se ha abordado la representación en OWL de las restricciones sobre tipos primitivos, a pesar de que la versión 2 del lenguaje ofrece soporte para ello, ni tampoco las restricciones de unicidad y orden. Por último, a diferencia de la propuesta de Artemis, la representación en OWL está ligada al modelo de referencia de openEHR y además sólo permite definir arquetipos para algunas clases del modelo de referencia.

Ninguna de las dos propuestas anteriores aborda de forma rigurosa la representación del modelo de referencia en OWL. En Artemis, la ontología del modelo de referencia es generada mediante un algoritmo básico que está limitado a las construcciones fundamentales de UML [71]. Asimismo, el algoritmo propuesto no está avalado por ningún estudio científico o recomendación de las entidades responsables de los lenguajes de modelado, esto es, OMG y W3C. Por otra parte, en [118] no se trata esta cuestión y reutilizan una ontología previamente definida para el estándar openEHR [164].

Sin embargo, el principal inconveniente de ambas propuestas para la validación técnica de arquetipos es que están soportadas por axiomas de inclusión en OWL. La restricción de cardinalidad ilustra el problema de la combinación de algunos axiomas de OWL con el axioma de inclusión que es empleado para relacionar los términos del arquetipo con las entidades del modelo de referencia. Por ejemplo, si una clase del modelo de referencia limita la cardinalidad máxima de un atributo a cinco instancias, un término de un arquetipo puede declararse subclase y establecer incorrectamente la cardinalidad máxima a un valor superior sin que sea detectado como una inconsistencia en OWL. El criterio de razonamiento de OWL es que mientras no haya instancias que violen la relación de subclase la ontología es consistente.

En definitiva, los trabajos que han abordado hasta ahora la definición de arquetipos en OWL presentan serias limitaciones para ser útiles como soporte para métodos de validación, ya sea por la omisión de restricciones (tipos primitivos, unicidad, orden, slot) o por la definición no adecuada de alguna de ellas (cardinalidad). Por otro lado, los arquetipos también pueden ser expresados como instancias, tal como propone el marco ontológico *Poseacle*. Sin embargo esta representación no permitiría aprovechar las posibilidades de razonamiento estructural de la lógica descriptiva en OWL.

La validación de documentos clínicos soportada por el lenguaje OWL ha sido estudiada en otros estándares. En [74] se propone una técnica para validar documentos HL7 CDA utilizando OWL. El objetivo del trabajo es la implementación mediante axiomas OWL de las reglas de uso de la terminología SNOMED-CT en HL7 [113]. Un ejemplo de regla de integridad es que el código de una observación en un documento CDA debe estar asociado a un concepto descendiente de "Entidad Observable" en SNOMED-CT. En este trabajo, el modelo R-MIM de HL7 CDA se ha definido como una ontología OWL y los documentos se representan como instancias de esa ontología. SNOMED-CT ha sido expresada también como una ontología en OWL utilizando las herramientas incluidas en la distribución de IHTSO, que implementan la propuesta presentada en [178, 177]. Los experimentos realizados en este trabajo muestran que el 73 % de los códigos de SNOMED-CT utilizados en los documentos CDA sobre observaciones violan las restricciones de integridad implementadas.

Además, un aspecto destacado del trabajo es la validación de los documentos aplicando razonamiento en mundo cerrado a través el razonador Pellet [176, 156]. Esto contrasta con el modelo de razonamiento propuesto en OWL basado en la asunción de mundo abierto. Los autores argumentan que el razonamiento en mundo abierto no es adecuado para la comprobación de restricciones de integridad, tal como se sugiere en [186]. Asimismo, también señalan que no es factible incluir la ontología completa de SNOMED-CT en un proceso de razonamiento, por lo que se ha construido un subconjunto mínimo para la tarea de validación.

Los trabajos anteriores han motivando que las tecnologías semánticas, y en especial el lenguaje de modelado OWL, hayan calado en otros modelos de información clínicos. Entre ellos destaca el trabajo de Tao *et al.* [185] en el que proponen una *metaontología* en OWL para la arquitectura de información CEM (Clinical Element Model) [35]. Las categorías básicas de modelado de contenido clínico, como un *panel* o una *sentencia*, se representan como clases en OWL y las relaciones semánticas que se establecen como propiedades. Las restricciones aplicadas para describir los elementos de información clínicos son representadas como axiomas OWL dando lugar a clases que definen los modelos clínicos. En general, la propuesta es similar a [109, 118], aunque la expresividad del lenguaje de modelado utilizado en CEM es más limitada que el paradigma de modelado con arquetipos.

Finalmente, la representación ontológica de los modelos clínicos ha sido abordada con una aproximación diferente a las anteriores. En [17] los arquetipos son representados como ontologías OWL donde las clases corresponden con los enlaces terminológicos asociados a los términos del arquetipo. El objetivo es favorecer la interoperabilidad semántica entre estándares a través de la alineación de ontologías basadas en terminologías clínicas. La propuesta fue presentada en 2009 y hasta ahora sólo se ha desarrollado la

parte del enlace automático de los términos de los arquetipos a SNOMED-CT [193]. En [169] se propone un método de obtención de la ontología de un arquetipo como subontología de la ontología SNOMED-CT. El objetivo del trabajo es asistir en el proceso de edición de plantillas openEHR seleccionando arquetipos relevantes semánticamente en las relaciones de *slot*.

## 4.6. Interoperabilidad de modelos clínicos

La iniciativa CIMI ha sacado a la luz la necesidad de que los modelos clínicos sean interoperables [31]. Hasta el momento la interoperabilidad se ha centrado en el intercambio de la información clínica, en vez de en el intercambio de modelos clínicos entre distintos sistemas de HCE. En este sentido, los arquetipos también tienen un papel importante en los sistemas de información sanitarios, controlando la entrada y visualización de la información clínica. En esta sección se describen principalmente propuestas de interoperabilidad a nivel de datos, aunque se destacan las principales aportaciones en relación a la interoperabilidad de modelos clínicos.

La arquitectura propuesta en el proyecto Artemis utiliza arquetipos para describir la semántica de los servicios sanitarios y de la información almacenada en los mensajes y documentos. La representación OWL de arquetipos eleva la arquitectura a un nivel semántico que es aprovechado para el descubrimiento de servicios y especialmente en el intercambio de datos. Sin embargo, los arquetipos no son transformados entre estándares, sino que se aprovechan las correspondencias establecidas entre ellos para transformar los datos. Por tanto, dos arquetipos pertenecientes a diferentes estándares que expresan el mismo concepto sanitario son alineados a nivel ontológico [110]. Parte de las correspondencias entre los arquetipos pueden ser inferidas si los modelos de referencia tienen el mismo origen ontológico, en este caso, HL7 RIM. También pueden aplicarse técnicas alineamiento ontológico para descubrir correspondencias entre los modelos de referencia. En una primera fase del proyecto las correspondencias entre ontologías fueron expresadas con la herramienta genérica MAFRA [121], y posteriormente se desarrolló una herramienta propia de alineamiento adaptada al uso de arquetipos [15].

En [13] se aborda el problema de la interoperabilidad semántica entre sistemas sanitarios utilizando un enfoque dirigido por ontologías y por las posibilidades de inferencia que proporciona OWL. La solución propuesta tiene como objetivo la integración de sistemas sanitarios heterogéneos que no necesariamente utilicen estándares. Por una parte, cada sistema está descrito por una *ontología de aplicación*. La obtención de esta ontología depende del tipo de sistema de información. En concreto, en el trabajo se presenta un módulo que genera una ontología de aplicación a partir de

un esquema de base de datos relacional, aunque la propuesta puede ser adaptada al uso de arquetipos. Por otro parte, se ha definido una ontología global, denominada *ontología canónica*, que expresa los aspectos semánticos del dominio y que está relacionada con terminologías clínicas estándares como SNOMED-CT. La ontología canónica actúa como mediadora en los procesos de transformación de la información clínica gracias a que las ontologías de aplicación son alineadas con la ontología canónica. Además, cabe destacar que en esta propuesta no se representan explícitamente los modelos de contenido clínico, sino que se manejan conceptos a nivel ontológico. La contribución más destacada del trabajo es el papel de OWL como técnica para expresar correspondencias con un alto nivel de expresividad, el uso de terminologías como referencia para alinear las ontologías y la aplicación razonamiento para el descubrimiento de correspondencias. Sin embargo, el principal inconveniente es el alto coste de construcción y mantenimiento de la ontología canónica.

En [108] se ha propuesto una arquitectura de interoperabilidad entre HL7 y openEHR que hace uso de ontologías OWL. Los modelos de referencia de los estándares son reconciliados utilizando técnicas de mapeo ontológico. El punto de partida son las ontologías previamente definidas para HL7 [155] y openEHR [164], y una propuesta de correspondencia entre tipos de datos entre ambos estándares [72]. En una primera fase aplican dos herramientas de mapeo ontológico, Falcon-AO [85] y Agreementmaker [37]. Las correspondencias obtenidas son refinadas y validadas por expertos en una segunda fase. Los autores argumentan que es necesario una etapa de refinamiento manual de las correspondencias debido a que los algoritmos de mapeo ontológico no consiguen la alineación completa de las ontologías y además se producen errores. Las correspondencias obtenidas son almacenadas en un formato propio en RDF. Esta aproximación ha sido desarrollada en [107] que introduce una *ontología puente* para relacionar las distintas ontologías participantes en la comunicación. Esta decisión se justifica por el hecho de que las herramientas de alineamiento ontológico sólo son capaces de relacionar dos ontologías y en el ámbito de las aplicaciones sanitarias son necesarias varias ontologías, como por ejemplo, cuando se relacionan dos modelos de referencia que hacen uso de terminologías clínicas. La herramienta utilizada para construir las ontologías puente es MAFRA [121], que también ha sido aplicada con el mismo propósito en el proyecto Artemis [42].

Dentro del estándar HL7 se ha estudiado la interoperabilidad de las versiones 2 y 3 utilizando OWL. En Artemis la interoperabilidad se consigue alineando los esquemas de los mensajes representados en OWL utilizando la herramienta OWLmt [15]. Por otro lado, en [18] se adopta una estrategia diferente basada en la construcción de las ontologías a partir de una ontología de alto nivel (BFO), lo que facilita el descubrimiento de las co-

rrespondencias entre los mensajes. Por último, en [167] se ha propuesto un enfoque metodológico para el desarrollo de ontologías para las dos versiones de HL7 y se han desarrollado herramientas semiautomáticas para la creación de las ontologías y para establecer los alineamientos.

Los organismos de estandarización han elaborado propuestas de armonización entre las distintas normas y especificaciones. La parte 3 del estándar ISO EN 13606 incluye una propuesta de arquetipos que corresponden con estructuras de datos para la representación de entradas clínicas de los modelos de referencia openEHR y HL7 versión 3. Asimismo, el modelo de referencia de ISO EN 13606 ha sido definido como un modelo R-MIM en el marco de modelado de HL7 versión 3 [45]. En [72] se realiza una propuesta de correspondencia de los tipos de HL7 y openEHR, y en [188] se propone la representación del modelo de referencia de openEHR en ISO EN 13606.

Por último, la herramienta Poseacle Converter define e implementa una metodológica para la transformación de modelos clínicos entre estándares de arquitectura dual [128]. El marco formal del proceso de transformación está definido por las ontologías *Poseacle* [59]. Dado el interés para esta tesis, la herramienta es descrita en la sección 7.2 de la memoria.



## Capítulo 5

# Representación ontológica de arquetipos

### 5.1. Introducción

Las arquitecturas de HCE basadas en el modelo dual utilizan arquetipos para expresar el conocimiento sobre la captura y almacenamiento de los modelos clínicos en las estructuras de datos proporcionadas por un modelo de información (modelo de referencia). En este capítulo se estudia la representación ontológica del conocimiento expresado en los arquetipos utilizando el lenguaje OWL. La definición formal de un arquetipo en OWL permite la implementación de tareas que aprovechan la semántica de los arquetipos como la validación de las restricciones, la transformación a otros modelos de información o la implementación de métricas de calidad. La expresividad del lenguaje OWL ofrece la posibilidad de representar los arquetipos de diferentes modos. En este capítulo se desarrollan tres representaciones de arquetipos que han sido utilizadas en distintas tareas de este trabajo de tesis.

Un arquetipo define un concepto clínico en base a restricciones sobre un modelo de información. La equivalencia formal entre el modelo de arquetipos y OWL consiste en establecer correspondencias entre las restricciones del modelo de arquetipos y las construcciones y axiomas que ofrece OWL. Además, el estudio de la representación de arquetipos en OWL no puede realizarse de forma aislada, sino que también debe incluir el modelo de información. En consecuencia, la formalización de la arquitectura dual en OWL requiere la representación del modelo de referencia y de los arquetipos.

La primera representación que se presenta en este capítulo tiene como principal objetivo la transformación de arquetipos y extractos de HCE entre estándares basados en la arquitectura dual. En el diseño del proceso de

transformación se utilizan varias ontologías en OWL que describen las características de los estándares de HCE basados en el modelo de arquetipos. La herramienta *Poseacle Converter* implementa el proceso de transformación en el que los arquetipos generados pueden ser obtenidos en ADL y OWL.

Las definiciones de un arquetipo deben ser correctas respecto al modelo de referencia en el que está definido y ha de establecer relaciones consistentes con otros arquetipos, en especial la relación de especialización. La segunda representación propuesta tiene como objetivo la validación de arquetipos y ha sido implementada en la herramienta *Archeck*. En esta representación los términos de un arquetipo corresponden con clases que deben ser consistentes con la ontología del modelo de referencia y, en caso de ser un arquetipo especializado, debe inferirse una relación de inclusión con la clase que representa el mismo concepto en el arquetipo base.

Finalmente, el marco de trabajo *Encorsetable* define un arquitectura para la conversión de modelos clínicos entre estándares de modelo dual. La arquitectura propuesta es una extensión de la herramienta *Poseacle Converter* orientada a incluir estándares de HCE que no utilicen el modelo de arquetipos. Además, la metodología refuerza el papel de los expertos en el dominio HCE en el proceso de conversión de modelos clínicos ofreciendo la posibilidad de definir las correspondencias entre los estándares. Estos requisitos han influido en la arquitectura *Encorsetable* que ha sido orientada al uso de ontologías OWL como soporte de todo el proceso de transformación, desde la definición de las correspondencias entre los modelos de información hasta la ejecución de la conversión de los modelos clínicos. En este capítulo se presentan dos representaciones de arquetipos en OWL que han sido empleadas en la implementación de la arquitectura.

## 5.2. Marco de comparación

La presentación de la correspondencia entre el modelo de arquetipos y OWL se realiza a través de algunos ejemplos de restricciones significativas en ADL. La figura 5.1 muestra un arquetipo de tipo `CLUSTER` que redefine el atributo `items` para establecer un rango formado por tres términos. La cardinalidad del atributo tiene un valor mínimo de 1 para ser compatible con las ocurrencias del término `at0002`. El término `at0003` declara una referencia interna al término `at0004` y un slot (`at0005`) que admite arquetipos “examen” de tipo `CLUSTER`.



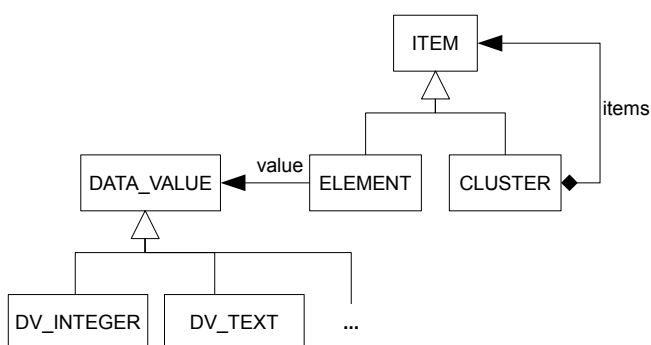
```

CLUSTER[at0000] matches {
  items cardinality matches {1..*; unordered; unique} matches {
    CLUSTER[at0001] occurrences matches {0..1} matches {
      items cardinality matches {0..*;unordered} matches {
        ELEMENT[at0004] occurrences matches {0..2} matches {...}
      }
    }
  }
ELEMENT[at0002] occurrences matches {1..*} matches { ... }
CLUSTER[at0003] occurrences matches {0..*} matches {
  items cardinality matches {0..*;unordered} matches {
    use_node ELEMENT occurrences matches {0..3}
    /items[at0001]/items[at0004]
  }
  allow_archetype CLUSTER[at0005] occurrences matches {0..1}
  matches {
    include archetype_id/value matches {
      /openEHR-EHR-CLUSTER\.examen(-[a-zA-Z0-9_]+)*\.v1/
    }
  }
}
...
}

```

Figura 5.1: Ejemplo de arquetipo

El modelo de referencia que se utiliza en los ejemplos es una parte del modelo de información de openEHR que corresponde con la jerarquía `ITEM` (véase figura 5.2).

Figura 5.2: Jerarquía `ITEM` del modelo de referencia de openEHR

Para ilustrar las restricciones sobre los tipos primitivos de openEHR se han seleccionado un conjunto de restricciones significativas. También se han incluido dos tipos de datos de interés en las HCE, como son los textos codificados y las cantidades.

- **Cadena conforme a una expresión regular:** atributo `nombre` de tipo `DV_TEXT` restringido a una cadena de caracteres que comienza por “a” seguida de tres dígitos.

---

```
nombre matches {/a\d{3}/}
```

---

- **Enumeración de cadenas:** atributo `unidad` de tipo `DV_TEXT` que sólo admite como valores las cadenas “cm” y “m”.

---

```
unidad matches {'cm', 'm'}
```

---

- **Intervalo numérico:** atributo `medida` de tipo `DV_INTEGER` restringido a un intervalo de enteros entre 3 y 5, ambos incluidos.

---

```
medida matches {3..5}
```

---

- **Patrón de fecha y tiempo.** Los datos temporales suelen ser representados en los modelos de referencia utilizando cadenas. Este es el caso de la clase `DV_DATE` que almacena la fecha en el atributo `value` de tipo `string`. La restricción se expresa como un patrón según el estándar ISO 8601 [93]. En el ejemplo se restringe el atributo `fecha` indicando que es obligatorio establecer los cuatro dígitos del año, opcional indicar el mes y no se admite la introducción de los días.

---

```
fecha matches {yyyy-?-XX}
```

---

- **Intervalo temporal:** atributo `fecha` es restringido para admitir fechas a partir del 1 de enero de 2000.

---

```
fecha matches {>= 2000-01-01}
```

---

- **Duración.** El tipo de datos para expresar duraciones admite valores como por ejemplo "5 días". En los modelos de referencia este tipo de datos suele ser definido como una cadena de acuerdo al patrón definido en ISO 8601 (por ejemplo 5 días se expresaría como P5d). Las restricciones sobre este tipo de datos son definidas de acuerdo a la siguiente expresión regular `P[Y|y][M|m][W|w][D|d][T[H|h][M|m][S|s]]`. En el siguiente ejemplo se restringe un atributo para aceptar duraciones que se expresan en horas y minutos (`PThm`).

---

```
atributo matches {PThm}
```

---

- **Texto codificado:** en el siguiente ejemplo se define un un texto codificado (`CODED_TEXT` en openEHR) restringido a tres códigos declarados en la ontología local del arquetipo (`at0010`, `at0011` y `at0012`).

---

```
ELEMENT[at0009] occurrences matches {0..1} matches {
  value matches {
    DV_CODED_TEXT matches {
      defining_code matches {
        [local::at0010, at0011, at0012]
      }
      ...
    }
  }
}
```

---

- **Cantidades:** en openEHR el tipo cantidad se representa con la clase `DV_QUANTITY`. En el ejemplo las restricciones indican que la unidad de medida se expresa en centímetros, en el intervalo -5,0 y 5,0, y con 0 decimales de precisión.

---

```
ELEMENT[at0006] occurrences matches {0..1} matches {
  value matches {
    C_DV_QUANTITY <
      list = <
        ["1"] = <
          units = <'cm'>
          magnitude = <-5.0..5.0>
          precision = <0>
        > ...
      >
    >
  }
}
```

---

### 5.3. Poseacle Converter

El marco de trabajo *Poseacle* está formado por un conjunto de ontologías que describen la arquitectura de modelo dual y que permiten la construcción de arquetipos como instancias de estas ontologías [59]. En esta sección se introduce la ontología que define los conceptos del modelo de arquetipos y a continuación se presenta una metodología para la construcción de modelos de referencia *arquetipables*.

#### 5.3.1. Ontología *Poseacle* del Modelo de Arquetipos

La ontología del modelo de arquetipos en *Poseacle* define los conceptos necesarios para representar las restricciones de los arquetipos y las categorías de estructuras de datos de la arquitectura dual. Esta ontología es la base para definir los modelos de información e instanciar los arquetipos. A continuación se presentan las clases más destacadas de esta ontología y en la figura 5.3 se muestran las propiedades que las relacionan:

- Clase `ARCHETYPE`. Esta clase representa a los arquetipos y se relaciona con las clases que describen las secciones que constituyen su declaración, esto es, la metainformación del arquetipo incluida en la cabecera (`ARCHETYPE_DESCRIPTION`), la definición del concepto raíz (`CLINICAL_TERM`) o la información de las terminologías clínicas con las que está enlazada.
- Clase `CONSTRAINT`. Las restricciones del modelo de arquetipos se organizan en torno a esta clase. De ella derivan restricciones específicas como las ocurrencias de un término o la cardinalidad de un atributo redefinido.
- Clase `CLINICAL_TERM`. Esta clase es la pieza que integra el modelo de información con el modelo de arquetipos. Su definición agrega la información y restricciones que caracterizan a los términos de un arquetipo.

#### 5.3.2. Ontología del Modelo de Referencia

El diseño de las ontologías de los modelos de información tiene como objetivo definir estructuras de datos *arquetipables*, esto es, que las instancias de la ontología definen arquetipos. Esta característica se consigue gracias a la integración del modelo de referencia y el modelo de arquetipos (véase figura 5.4). Las clases que definen las estructuras y tipos de datos del modelo de información se declaran subclases de `CLINICAL_TERM`. Esta relación lógica entre las dos ontologías hace que las clases del modelo de información puedan ser restringidas conforme al modelo de arquetipos.

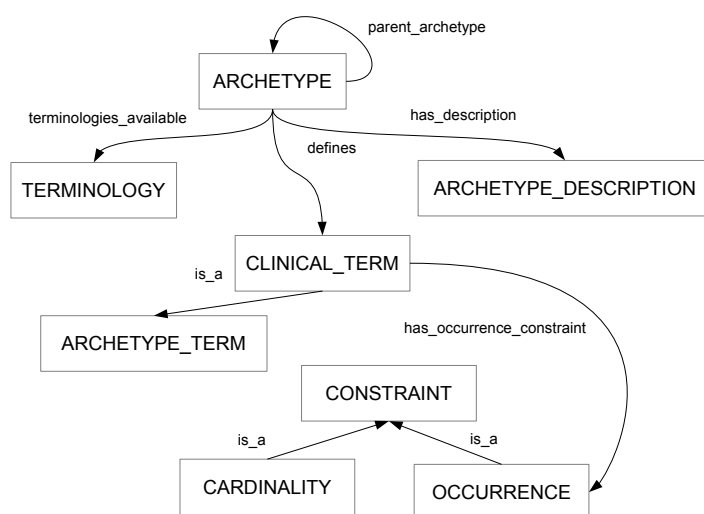


Figura 5.3: Extracto de la ontología del modelo de arquetipos en *Poseacle*

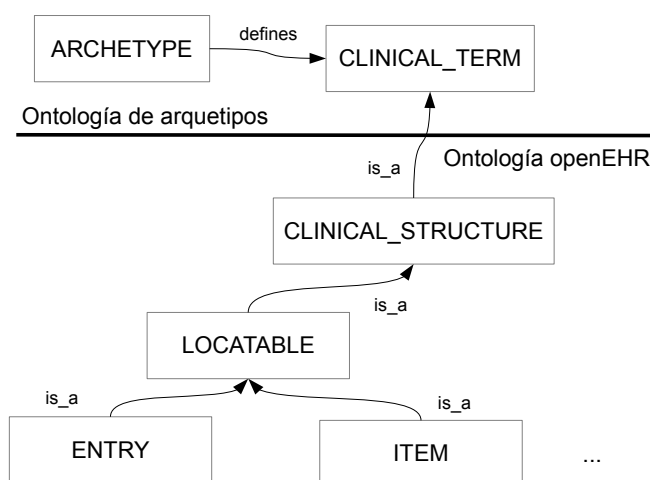


Figura 5.4: Integración de la ontología openEHR y la ontología del modelo de arquetipos en *Poseacle Converter*

La figura 5.5 ilustra la definición en OWL de la jerarquía `ITEM` de openEHR. Los criterios aplicados se describen a continuación:

- Una clase UML corresponde con una clase en OWL que es subclase directa o indirecta de `CLINICAL_TERM`. En el caso de openEHR la relación es indirecta a través de `CLINICAL_STRUCTURE` (véase figura 5.4)

---

```
Class: ITEM
  SubClassOf: CLINICAL_STRUCTURE
```

---

- Las relaciones de herencia entre clases UML se traducen en axiomas de inclusión.

---

```
Class: CLUSTER
  SubClassOf: ITEM

Class: ELEMENT
  SubClassOf: ITEM
```

---

- Las clases UML que heredan de la misma clase se declaran disjuntas.

---

```
DisjointClasses: CLUSTER, ELEMENT
```

---

- Los atributos de las clases UML se declaran como propiedades de objetos o datos en OWL (*ObjectProperty* o *DatatypeProperty*). El dominio de una propiedad OWL corresponde con la clase que declara el atributo y el rango de una propiedad OWL se define como el tipo del atributo. Si un atributo tiene el mismo nombre y rango en varias clases, el dominio de la propiedad es la unión de las clases en las que está declarado el atributo.

---

```
ObjectProperty: items
  Domain: CLUSTER or ITEM_TREE
  Range: ITEM

ObjectProperty: value_element
  Domain: ELEMENT,
  Range: DATAVALUE
```

---

- La definición del rango de una propiedad OWL se refuerza incluyendo en la clase que declara el atributo un axioma de inclusión basado en el cuantificador universal que sólo permite la asociación con instancias de la clase que representa el rango.

---

```

Class: CLUSTER
  SubClassOf: items only ITEM

Class: ELEMENT
  SubClassOf: value_element only DATAVALUE

```

---

```

Class: ITEM
  SubClassOf: CLINICAL_STRUCTURE

Class: CLUSTER
  SubClassOf:
    ITEM,
    items only ITEM

ObjectProperty: items
  Domain: CLUSTER or ITEM_TREE
  Range: ITEM

Class: ELEMENT
  SubClassOf:
    ITEM,
    value_element only DATAVALUE

ObjectProperty: value_element
  Domain: ELEMENT,
  Range: DATAVALUE

DisjointClasses: CLUSTER, ELEMENT

```

Figura 5.5: Jerarquía ITEM de openEHR en Poseacle Converter

### 5.3.3. Arquetipos *Poseacle*

La herramienta *Poseacle Converter* representa un arquetipo en OWL como instancias de la ontología del modelo de referencia. En este apartado se desarrolla parte de la definición del arquetipo de la figura 5.1 analizando los tipos de restricciones del modelo de arquetipos.

Las instancias OWL que describen al arquetipo son nombradas utilizando la clase a la que pertenecen y un identificador, que corresponde con el código *at* de los términos del arquetipo o con un número entero en el resto de casos. Nótese que la convención de nombrado utilizada tiene como pro-

pósito favorecer la legibilidad del código, puesto que la herramienta utiliza como identificador de los individuos un número generado aleatoriamente.

**Tipo restringido.** El término *at0000* del arquetipo declara que restringe la clase `CLUSTER`. La definición OWL corresponde con una instancia cuyo tipo es la clase del modelo de referencia que está restringiendo.

---

```

ADL
CLUSTER[at0000] matches { ...
OWL
Individual: CLUSTER_at0000
Types: CLUSTER

```

---

**Restricción de cardinalidad, orden y unicidad.** El término *at0000* del arquetipo redefine el atributo multivaluado `items`. La definición de cardinalidad de este atributo se expresa con una instancia de la clase `CARDINALITY` que se asocia al término que redefine el atributo mediante la propiedad `has_cardinality_constraint`. El intervalo de cardinalidad se expresa con la clase `INTEGER_INTERVAL` que se asocia con la definición de cardinalidad con la propiedad `interval`.

---

```

ADL
CLUSTER[at0000] matches {
  items cardinality matches {1..*; unordered; unique} matches ...
OWL
Individual: CLUSTER_at0000
Facts:
  has_cardinality_constraint CARDINALITY_CLUSTER_at0000_items
Individual: CARDINALITY_CLUSTER_at0000_items
Types: CARDINALITY
Facts:
  is_ordered false,
  is_unique true,
  interval INTERVAL_CARDINALITY_CLUSTER_at0000_items
Individual: INTERVAL_CARDINALITY_CLUSTER_at0000_items
Types: INTEGER_INTERVAL
Facts:
  lower_bound 1

```

---



**Restricción de rango.** El atributo `items` del término `at0000` se ha redefinido para permitir 3 términos arquetipados. Se utiliza la propiedad OWL correspondiente al atributo para asociar la instancia del término contenedor (`at0000`) con las instancias que representan a los términos que definen el rango.

---

ADL

---

```

CLUSTER[at0000] matches {
  items ... matches {
    CLUSTER[at0001] ...
    ELEMENT[at0002] ...
    CLUSTER[at0003] ...
  }
}

```

---

OWL

---

```

Individual: CLUSTER_at0000
Facts:
  items CLUSTER_at0001,
  items ELEMENT_at0002,
  items CLUSTER_at0003

```

---

**Restricción de ocurrencias.** El término `at0001` declara que como máximo una instancia puede estar asociada con `at0000` a través del atributo `items`. La restricción se representa como un individuo de la clase `OCCURRENCE` que contiene la información sobre el atributo y término contenedor en el que se declara, el intervalo de ocurrencias y la declaración de las restricciones de orden y unicidad que hereda del atributo donde está incluido el término.

---

ADL

---

```

CLUSTER[at0000] matches {
  items cardinality matches {1..*; unordered} matches {
    CLUSTER[at0001] occurrences matches {0..1} matches {...
  }
}

```

---

OWL

---

```

Individual: CLUSTER_at0001
Types: CLUSTER
Facts:
  has_occurrence_constraint OCCURRENCE_CLUSTER_at0000_items_at0001

Individual: OCCURRENCE_CLUSTER_at0000_items_at0001
Types: OCCURRENCE
Facts:
  container CLUSTER_at0000,
  attribute 'items'

```

```

is_ordered false,
is_unique false,
interval INTERVAL_CONSTRAINT_CLUSTER_at0000_items_at0001

Individual: INTERVAL_CONSTRAINT_CLUSTER_at0000_items_at0001
Types: INTEGER_INTERVAL
Facts:
  lower_bound 0,
  upper_bound 1

```

---

**Referencias internas.** El término *at0004* es reutilizado en la declaración del rango del atributo *items* del término *at0003*. Esta restricción se expresa con una instancia de la clase `ARCHETYPE_INTERNAL_REF` que se enlaza con el término contenedor utilizando la propiedad `has_constraint_object`. La instancia declara las ocurrencias que tiene el término (se omite en el código OWL del ejemplo), el atributo en el que se declara como rango y la ruta en el árbol ADL del término que es reutilizado.

---

```

ADL
CLUSTER[at0000] matches {
  items ... {
    CLUSTER[at0001] ... matches {
      items ... {
        ELEMENT[at0004] occurrences matches {0..2} matches {...}
      }
    }
    CLUSTER[at0003] ... matches {
      items ... {
        use_node ELEMENT occurrences matches {0..3}
        /items[at0001]/items[at0004]
      }
    }
  }
}

```

---

```

OWL

Individual: CLUSTER_at0005
Facts:
  has_constraint_object C_INTERNAL_REF_1

Individual: C_INTERNAL_REF_1
Types: ARCHETYPE_INTERNAL_REF
Facts:
  has_occurrence_constraint OCCURRENCE_C_INTERNAL_REF_1,
  restrict 'items',
  target_path '/items[at0001]/items[at0004]'

```

---

**Relación Slot.** El término *at0003* restringe el rango del atributo *items* declarando un término (*at0005*) que representa un slot. La restricción se expresa de forma análoga a una referencia interna. Un slot es una

instancia de la clase `ARCHETYPE_SLOT` que declara la clase del modelo de referencia que representa el slot (propiedad `archetype_type`), el nombre del atributo al que pertenece, la restricción de ocurrencias y un aserto ADL que especifica los arquetipos permitidos. La instancia se relaciona con el término contenedor utilizando la propiedad `has_constraint_object`.

---

ADL

---

```

CLUSTER[at0003] matches {
  items cardinality matches {1..*; unordered} matches {
    allow_archetype
    CLUSTER[at0005] occurrences matches {0..3} matches {
      include archetype_id/value matches {
        /openEHR-EHR-CLUSTER\.examen(-[a-zA-Z0-9_]+)*\.v1/
      }
    }
  }
}

```

---

OWL

---

```

Individual: CLUSTER_at0003
...
Facts:
  has_constraint_object SLOT_at0001

Individual: SLOT_at0005
Types: ARCHETYPE_SLOT
Facts:
  archetype_type 'CLUSTER',
  restrict 'items',
  has_occurrence_constraint OCCURRENCE_SLOT_at0001,
  includes ASSERTION_SLOT_at0005

Individual: ASSERTION_SLOT_at0005
Types: ASSERTION
Facts:
  string_expression 'openEHR-EHR-CLUSTER\.examen(-[a-zA-Z0-9_]+)*\.v1'

```

---

#### 5.3.4. Restricciones sobre tipos primitivos

La ontología *Poseacle* del modelo de arquetipos define una clase como raíz de los tipos de datos del modelo de referencia (`CLINICAL_DATATYPE`). Cada tipo de datos tiene asociada una clase que representa las restricciones que soporta, como por ejemplo `C_STRING` que restringe las cadenas de caracteres o `C_QUANTITY` que hace lo propio con los tipos de datos que representan cantidades.

A continuación se desarrollan los ejemplos de restricciones sobre tipos primitivos presentados en el apartado 5.2.

**Cadena conforme a una expresión regular.** La clase `C_STRING` hace uso de varias propiedades que permiten expresar las restricciones sobre cadenas del modelo de arquetipos. Entre ellas, la propiedad `pattern` se utiliza para establecer un patrón de expresión regular.

---

ADL

---

```
nombre matches {/a\d{3}/}
```

---

OWL

---

```
Individual: DV_TEXT_1
Types: DV_TEXT
Facts: has_constraint_datatype C_STRING_1

Individual: C_STRING_1
Types: C_STRING
Facts: pattern 'a\d{3}'
```

---

**Enumeración de cadenas.** La clase `C_STRING` también permite definir la restricción de una cadena basada en una enumeración de valores. La propiedad `list` asocia el individuo que representa la restricción con cada uno de los componentes de la enumeración.

---

ADL

---

```
unidad matches {'cm', 'm'}
```

---

OWL

---

```
Individual: DV_TEXT_1
Types: DV_TEXT
Facts: has_constraint_datatype C_STRING_1

Individual: C_STRING_1
Types: C_STRING
Facts:
  list 'cm',
  list 'm'
```

---

**Intervalo numérico.** Las restricciones sobre enteros se declaran como instancias de la clase `C_INTEGER`, y las de tipo real con la clase `C_REAL`. La restricción de intervalo se define utilizando una clase que representa un intervalo (`INTEGER_INTERVAL` para el tipo entero) y se asocia con la instancia que declara la restricción mediante la propiedad `range`.

---

ADL

---

unidad matches {3..5}

---

OWL

---

Individual: DV\_INTEGER\_1  
 Types: DV\_INTEGER  
 Facts: has\_constraint\_datatype C\_DV\_INTEGER\_1

Individual: C\_DV\_INTEGER\_1  
 Types: C\_INTEGER  
 Facts: range RANGE\_DV\_INTEGER\_1

Individual: RANGE\_DV\_INTEGER\_1  
 Types: INTEGER\_INTERVAL  
 Facts:  
   lower\_bound 3,  
   upper\_bound 5

---

**Patrón de fecha y tiempo.** Las instancias de la clase `C_TIME` se emplean para restringir los datos temporales y pueden ser asociadas con patrones utilizando la propiedad `pattern`.

---

ADL

---

atributo matches {yyyy-??-XX}

---

OWL

---

Individual: DV\_DATE\_TIME\_1  
 Types: DV\_DATE\_TIME  
 Facts: has\_constraint\_datatype C\_DV\_DATE\_TIME\_1

Individual: C\_DV\_DATE\_TIME\_1  
 Types: C\_TIME  
 Facts: pattern 'yyyy-??-XX'

---

**Intervalo temporal.** La clase `C_DATE_TIME` se utiliza para instanciar las restricciones sobre datos temporales. La propiedad `interval` permite asociar la restricción con una instancia que representa intervalos de tiempo (`TIME_INTERVAL`).

---

ADL

---

fecha matches {>= 2000-01-01}

---

OWL

---

```

Individual: DV_DATE_TIME_1
  Types: DV_DATE_TIME
  Facts: has_constraint_datatype C_DV_DATE_TIME_1

Individual: C_DV_DATE_TIME_1
  Types: C_TIME
  Facts: interval INTERVAL_DV_DATE_TIME_1

Individual: INTERVAL_DV_DATE_TIME_1
  Types: TIME_INTERVAL
  Facts: lower_bound '2000-01-01T00:00:00'^^xsd:dateTime

```

---

**Duración.** Las instancias de la clase `C_DURATION` representan las restricciones sobre duraciones y establecen el patrón utilizando la propiedad `pattern`.

---

ADL

---

```

atributo matches {PThm}

```

---

OWL

---

```

Individual: DV_DURATION_1
  Types: DV_DURATION
  Facts: has_constraint_datatype C_DV_DURATION_1

Individual: C_DV_DURATION_1
  Types: C_DURATION
  Facts: pattern 'PThm'

```

---

**Texto codificado.** Los códigos admitidos en un texto codificado se representan con la clase `C_CODED_TEXT`. La propiedad `code_list` conecta los códigos con la instancia que define la restricción y la propiedad `terminology` establece una instancia que especifica la terminología a la que pertenecen.

---

ADL

---

```

ELEMENT[at0009] occurrences matches {0..1} matches {
  value matches {
    DV_CODED_TEXT matches {
      defining_code matches {
        [local::at0010, at0011, at0012]
      }
    }
  }
  ...
}

```

---

OWL

---

```
Individual: C_CODED_TEXT_1
Types: C_CODED_TEXT
Facts:
  code_list 'at0010',
  code_list 'at0011',
  code_list 'at0012',
  terminology TERMINOLOGY_C_CODED_TEXT_1
```

```
Individual: TERMINOLOGY_C_CODED_TEXT_1
Types: TERMINOLOGY
Facts: id 'local'
```

---

## 5.4. Archeck

La herramienta *Archeck* implementa un método de validación de arquetipos. Por una parte, un arquetipo es *correcto* respecto al modelo de referencia si las restricciones que declara son compatibles con las definiciones de las clases del modelo de información. Por otro lado, los arquetipos establecen relaciones de especialización que deben ser *consistentes*, esto es, los extractos de HCE que representa un arquetipo especializado han de ser conformes con las restricciones del arquetipo base.

La técnica de validación del método *Archeck* se fundamenta en los servicios de razonamiento estructural (*TBox*) que ofrece la lógica descriptiva: identificación de clases inconsistentes e inferencia de relaciones de inclusión entre clases. Los arquetipos se declaran como clases OWL cuyas definiciones permiten inferir si son correctas respecto al modelo de referencia (clases consistentes) y si las relaciones de especialización entre arquetipos implican relaciones de inclusión de las clases que los representan en OWL. Sin embargo, los requisitos que tienen los dos tipos de validaciones no son los mismos, por lo que es necesario que ciertas restricciones tengan una doble representación.

### 5.4.1. Modelo de Referencia en *Archeck*

La representación OWL del modelo de referencia que utiliza *Archeck* implementa las recomendaciones de la propuesta ODM del OMG [149]. La especificación ODM tiene como fin establecer los puentes entre los lenguajes de modelado de sistemas de información y los lenguajes de representación del conocimiento. Para ello define las correspondencias entre los lenguajes UML y OWL, que son expresadas en el lenguaje de transformación de modelos QVT. A continuación se enumeran las reglas básicas que permiten transformar un modelo de referencia de UML a OWL:

- Las clases del modelo de referencia son transformadas en clases OWL.
- Los atributos de las clases son representados como propiedades OWL. Si el tipo del atributo es una clase, la propiedad será definida como una propiedad entre objetos (*ObjectProperty*), en caso contrario será expresada como una propiedad de datos (*DatatypeProperty*). El dominio de la propiedad corresponde con la clase a la que pertenece el atributo y el rango con su tipo.
- Las características de un atributo que puede restringir un arquetipo se declaran en la clase donde está definido mediante axiomas *SubClassOf*.
- El rango de un atributo es expresado como una restricción en la clase a la que pertenece y es definida utilizando el cuantificador universal de OWL.
- La cardinalidad de un atributo se define a través de una clase anónima que restringe la cardinalidad de la propiedad OWL correspondiente según los valores de cardinalidad del atributo.
- La cardinalidad de un atributo también se tiene en cuenta para declarar la propiedad como funcional. Si el atributo no es multivaluado, esto es, su cardinalidad máxima es uno, la propiedad OWL se declara como funcional.
- Las relaciones de herencia entre las clases del modelo de referencia se representan mediante axiomas *SubClassOf*.
- Las clases que comparten el mismo padre se declaran disjuntas.
- Los atributos heredados se declaran también como una propiedad que se relaciona con la propiedad OWL correspondiente declarada en el padre utilizando un axioma *SubPropertyOf*.
- Si un atributo es obligatorio, su definición es restringida con el cuantificador existencial de OWL.

La figura 5.6 muestra la representación OWL de la clase `CLUSTER` de `openEHR`. El primer axioma declara que un `CLUSTER` es un tipo de `ITEM`. Seguidamente, la clase incluye una restricción sobre el atributo `name` expresando que es obligatorio. Dado que `ELEMENT` también es un tipo de `ITEM`, se indica que la clase es disjunta con `ELEMENT` en el último axioma. El atributo multivaluado `items` declarado en la clase `CLUSTER` se define como una propiedad entre objetos (*ObjectProperty*) debido a que su tipo es la clase `ITEM`. El nombre de las propiedades se forma concatenando el nombre de la clase contenedora (por ejemplo, `CLUSTER`) y el nombre del atributo (por ejemplo, `items`). Nótese que la definición de la clase `CLUSTER` no incluye ninguna



```
Class: CLUSTER
SubClassOf:
  ITEM,
  CLUSTER_name some xsd:string,
  CLUSTER_name only xsd:string,
  CLUSTER_items only ITEM
DisjointWith: ELEMENT

ObjectProperty: CLUSTER_items
Domain: CLUSTER
Range: ITEM

DataProperty: CLUSTER_name
Domain: CLUSTER
Range: xsd:string
SubPropertyOf: ITEM_name
Characteristics: Functional
```

Figura 5.6: Representación en Archeck de la clase `CLUSTER` de openEHR

restricción para la cardinalidad de `items` ya que la cardinalidad de este atributo se define `0..*`. El atributo `name` se hereda de `ITEM` y es representado como una propiedad de datos funcional, ya que un `CLUSTER` sólo puede ser asociado como máximo con un valor, su rango es el tipo primitivo `string`, y se declara como subpropiedad de la propiedad asociada al atributo `name` de la clase `ITEM`. Cabe destacar que la cardinalidad del atributo `name` es `1..1` y que esta restricción se expresa, por una parte, en la clase (`CLUSTER`), y por otra, en la definición de la propiedad (`CLUSTER_name`). Por último, el tipo de ambos atributos es expresado en la clase `CLUSTER` utilizando restricciones basadas en el cuantificador universal. Estas restricciones no son redundantes con la definición del rango de las propiedades, ya que el rango de una propiedad sólo permite inferir axiomas de pertenencia de individuos a clases, mientras que la representación que se propone en Archeck pretende obtener inferencias a partir de las definiciones de clases.

#### 5.4.2. Modelo de arquetipos en *Archeck*

Archeck representa en OWL los arquetipos como clases. Las restricciones que forman la definición de un término de un arquetipo son traducidas en axiomas que se combinan para definir una clase OWL. La validación definida en Archeck tiene una doble vertiente, el modelo de referencia y la relación de especialización, cada una con requisitos de validación diferentes. Respecto al modelo de referencia interesa comprobar que las res-

tricciones de un atributo sean consistentes con su definición. En cambio, para validar la especialización de arquetipos se debe comprobar si la definición del arquetipo base subsume al arquetipo especializado. Esto da lugar a que una misma restricción pueda tener una doble representación en OWL. En la implementación de Archeck se ha optado por mantener estas dos representaciones juntas, aunque también podría haberse utilizado dos representaciones separadas. La única desventaja de la opción elegida es que si se detecta que un arquetipo es incorrecto respecto al modelo de referencia, la clase OWL asociada es inconsistente y por tanto no puede continuar la validación de la relación de especialización. Sin embargo, se ha elegido la representación conjunta por entender que la conformidad con el modelo de referencia es requisito para la validación de la especialización.

#### 5.4.2.1. Representación para la validación de la relación de especialización

En este apartado se presenta la representación de un arquetipo en OWL como soporte para la validación de la relación de especialización. Se ha optado por presentar primero esta definición por ser una aproximación más directa a la interpretación de las definiciones del modelo de arquetipos. La definición de la clase OWL asociada con el concepto raíz del arquetipo de ejemplo se muestra en la figura 5.7. El nombre de la clase resulta de la combinación del nombre de la clase que se restringe en el modelo de referencia (`CLUSTER`) y el identificador de concepto (`at0000`). La clase se define mediante un axioma de **equivalencia** que contienen los siguiente subaxiomas:

- Todas las clases se definen como subclases de `ARCHETYPED_CLASS`, la cual define todas las propiedades comunes a los términos arquetipados.
- La clase del modelo de referencia que es restringida (`CLUSTER`).
- La clase que representa el significado del concepto (`CONCEPT_at0000`).
- Las restricciones de los atributos. En el ejemplo de la figura 5.7, sólo es restringido el atributo `items`.

```

Class: CLUSTER_at0000
  EquivalentTo:
    ARCHETYPED_CLASS
    and CLUSTER
    and CONCEPT_at0000
    and CLUSTER_items only UNORDERED
    and CLUSTER_items only UNIQUE
    and CLUSTER_items min 1 ITEM
    and CLUSTER_items max 1 CLUSTER_at0001
    and CLUSTER_items min 1 ELEMENT_at0002

```

Figura 5.7: Representación Archeck del concepto raíz del arquetipo

La representación OWL propuesta no distingue entre un arquetipo especializado y otro normal. Dados los axiomas elegidos para definir las restricciones, el proceso de validación no necesita que se establezca ninguna relación entre los términos de arquetipo especializado y los términos del arquetipo base. El método de validación consiste en comprobar si las clases que definen los términos en el arquetipo base incluyen las clases que representan los términos en el arquetipo especializado.

A continuación se desarrolla la representación en OWL de las restricciones de los atributos:

**Definiciones de unicidad y orden.** Se propone el uso de dos clases disjuntas para cada una de las restricciones, `ORDERED` y `UNORDERED` para la restricción de orden, y `UNIQUE` y `NON_UNIQUE` para la restricción de unicidad. Las restricciones se expresan con el cuantificador universal sobre la propiedad asociada al atributo que restringen. Por tanto, dado que las clases son disjuntas, un arquetipo especializado define correctamente la restricción si establece el mismo valor que el arquetipo base.

---

ADL

---

```

CLUSTER[at0000] matches {
  items cardinality matches {1..*; unordered; unique} matches {
    ...

```

---

OWL

---

```

CLUSTER_items only UNORDERED
CLUSTER_items only UNIQUE

```

---

**Restricción de cardinalidad.** La definición de la cardinalidad de un atributo multivaluado, tanto en UML como en ADL, tiene su correspondencia en OWL utilizando axiomas de cardinalidad. El axioma de cardinalidad restringe la propiedad OWL que representa el atributo. Además, la restricción de cardinalidad puede ser expresada de forma cualificada, es decir, puede indicarse la clase sobre la que se establece la restricción. En el caso de la cardinalidad de un atributo no es necesario que la restricción sea cualificada, ya que en la representación del modelo de referencia en OWL se ha limitado el rango de la propiedad utilizando una restricción basada en el cuantificador universal. No obstante, se utiliza el axioma de cardinalidad cualificado para que la definición sea más legible.

---

ADL

---

```

CLUSTER[at0000] matches {
  items cardinality matches {1..*; unordered; unique} matches {

```

---

OWL

---

```

CLUSTER_items min 1 ITEM

```

---

**Restricción de ocurrencias.** Este tipo de restricción es propia de los atributos multivaluados y sólo es aplicable para la validación de la relación de especialización, ya que en UML no existe tal definición. De nuevo, el axioma de cardinalidad de OWL nos permite restringir el número de instancias que participan en una asociación. Si el valor mínimo de intervalo de ocurrencias es 0, no se introduce ningún axioma, ya que no tiene ninguna consecuencia lógica indicar que al menos habrá 0 instancias. Por otro lado, si el intervalo de ocurrencias no está acotado, tampoco se introduce un axioma de cardinalidad máxima.

---

ADL

---

```

CLUSTER[at0000] matches {
  items cardinality matches {1..*; unordered; unique} matches {
    CLUSTER[at0001] occurrences matches {0..1} matches { ... }
    ELEMENT[at0002] occurrences matches {1..*} matches { ... }
    CLUSTER[at0003] occurrences matches {0..*} matches { ... }

```

---

OWL

---

```

CLUSTER_items max 1 CLUSTER_at0001
CLUSTER_items min 1 ELEMENT_at0002

```

---

**Referencias internas.** Un término arquetipado puede formar parte del rango de varios atributos haciendo uso de las referencias internas. La

representación propuesta que asocia una clase OWL por cada término no distingue entre si el término está declarado como rango de un atributo o es reutilizado con una referencia interna. En ambos casos el término es independiente del atributo en el que se asocia como rango.

---

ADL

---

```

CLUSTER[at0000] matches {
  items ... {
    CLUSTER[at0001] ... matches {
      items ... {
        ELEMENT[at0004] occurrences matches {0..2} matches {...}
      }
    }
  }
  ...
  CLUSTER[at0003] ... matches {
    use_node ELEMENT occurrences matches {0..3}
    /items[at0001]/items[at0004]
  }
  ...

```

---

OWL

---

```

Class: CLUSTER_at0005
  EquivalentTo: ... and CLUSTER_items max 3 ELEMENT_at0004

```

---

**Relación slot.** La clase ARCHETYPED\_CLASS representa a cualquier término arquetipado. En el caso de que el término defina un slot, utiliza la propiedad `from_archetype` para asociar al término arquetipado la expresión regular que filtra los arquetipos permitidos en el slot. La expresión regular restringe el tipo `string` mediante el uso de una faceta de OWL 2.

---

ADL

---

```

CLUSTER[at0003] matches {
  items cardinality matches {1..*; unordered} matches {
    allow_archetype
      CLUSTER[at0005] occurrences matches {0..3} matches {
        include archetype_id/value matches {
          /openEHR-EHR-CLUSTER\.examen(-[a-zA-Z0-9_]+)*\.v1/
        }
      }
  }
}

```

---

OWL

---

```

Class: CLUSTER_at0005
  EquivalentTo: ... and from_archetype only string[pattern
    'openEHR-EHR-CLUSTER\.examen(-[a-zA-Z0-9_]+)*\.v1']

```

---

### 5.4.2.2. Restricciones sobre tipos primitivos

OWL 2 introduce axiomas para definir restricciones sobre los tipos de datos primitivos de XML. Estos axiomas son suficientes para expresar la semántica de las restricciones de datos de ADL en OWL. Por tanto, no es necesario introducir nuevas clases o propiedades OWL para representar las restricciones de tipos primitivos. Seguidamente se presentan los axiomas OWL que corresponden a los ejemplos introducidos en el apartado 5.2. La convención de nombrado de las clases y propiedades OWL se simplifica con el fin de favorecer la legibilidad de los ejemplos, es decir, los nombres de propiedades corresponden con el nombre del atributo que representan, por ejemplo, se utiliza *value* para identificar el atributo *value* de la clase *DV\_TEXT* de openEHR, en vez de *DV\_TEXT\_value*.

**Cadena conforme a una expresión regular.** En OWL 2 el dominio del tipo de datos *string* puede ser acotado utilizando una expresión regular. El axioma que establece esta restricción se utiliza para definir el rango de datos del atributo utilizando el cuantificador universal de OWL.

```

_____ ADL _____
nombre matches {/a\d{3}/}

_____ OWL _____
DV_TEXT and value only string[pattern 'a\d{3}']

```

**Enumeración de cadenas.** Esta restricción tiene una correspondencia exacta en OWL restringiendo el rango del atributo con una enumeración de valores.

```

_____ ADL _____
unidad matches {'cm', 'm'}

_____ OWL _____
DV_TEXT and value only {'cm', 'm'}

```

**Intervalo numérico.** La representación OWL de un intervalo numérico equivale a establecer el rango del atributo con una faceta que acote el conjunto de valores del tipo numérico correspondiente.

```

_____ ADL _____
medida matches {3..5}

```

---

 OWL
 

---

```
DV_ORDINAL and value only xsd:int[>= 3, <= 5]
```

---

**Patrón de fecha y tiempo.** La restricción se expresa como un patrón según el estándar ISO 8601. Sin embargo, estos patrones no son expresiones regulares que acotan el dominio de las cadenas que almacenan los datos temporales, sino que son patrones que guían la captura de los datos. La representación propuesta en OWL utiliza una restricción basada en un patrón de expresión regular sobre cadenas que sólo es de utilidad para la validación de la restricción en una relación de especialización. En otras palabras, el patrón no sirve para validar las cadenas que representan las fechas en los extractos HCE. En el ejemplo que se presenta a continuación, el patrón `yyyy-??-XX` es traducido a OWL con la expresión regular `/yyyy(-mm)?/`. De este modo, un arquetipo descendiente que defina un patrón que establezca el mes (patrón `yyyy-mm-XX`) es compatible con la definición anterior.

---

 ADL
 

---

```
fecha matches {yyyy-??-XX}
```

---



---

 OWL
 

---

```
DV_DATE and value only string[pattern 'yyyy(\\-mm)?']
```

---

**Intervalo temporal.** En OWL los datos temporales se expresan con el tipo XML `dateTime` y un intervalo temporal se define como un intervalo sobre este tipo. Nótese que en el caso anterior y en éste se restringe el atributo `value` de `DV_DATE` con dos facetas OWL sobre tipos diferentes: una expresión regular sobre el tipo `string` y un intervalo sobre el tipo `dateTime`. Para que el atributo `value` pueda soportar los dos tipos de restricciones, la propiedad OWL que lo representa se ha declarado sin un rango, es decir, admite cualquier valor literal. De este modo, un razonador OWL no encontrará estas definiciones contradictorias.

---

 ADL
 

---

```
fecha matches {>= 2000-01-01}
```

---



---

 OWL
 

---

```
DV_DATE and value only
  xsd:dateTime[>= '2000-01-01T00:00:00'^xsd:dateTime]
```

---

**Duración.** En los modelos de referencia un dato que expresa una duración se representa como una cadena. El formato de la cadena debe ser conforme con un patrón de acuerdo al estándar ISO 8601. Por tanto, el axioma OWL equivalente a la restricción de una duración en ADL es una expresión regular sobre el tipo `string`.

---

ADL

---

```
atributo matches {PThm}
```

---

OWL

---

```
DV_DURATION and value only string[pattern 'PThm']
```

---

**Texto codificado.** El conjunto de códigos aceptados por un texto codificado se expresa en OWL restringiendo las propiedades `code_string` y `terminology_id` de la clase `CODE_PHRASE`. Los códigos se representan mediante cadenas, por lo que la propiedad `code_string` puede acotarse con la enumeración de cadenas que corresponde a los códigos. La terminología asociada a los códigos se expresa utilizando una clase anónima que sólo permite cadenas (`DV_TEXT`) cuyo valor sea el identificador de la terminología.

---

ADL

---

```
ELEMENT[at0009] occurrences matches {0..1} matches {
  value matches {
    DV_CODED_TEXT matches {
      defining_code matches {
        [local::at0010, at0011, at0012]
      }
      ...
    }
  }
}
```

---

OWL

---

```
DV_CODED_TEXT
and defining_code only (
  (CODE_PHRASE
    and code_string only {'at0010', 'at0011', 'at0012'}
    and terminology_id only
      (DV_TEXT and value value 'local'))
  )
```

---

**Cantidades.** Las definiciones ADL sobre los tipos que representan las cantidades se expresan en OWL restringiendo cada atributo de la clase según su tipo de datos. En el ejemplo, la propiedad `units` es de tipo `string` y se define limitando la asociación al único valor permitido ("cm"). La propiedad `magnitude` es un número real que es acotado



con un intervalo numérico. La última propiedad es `precision` que se establece con el único valor aceptable. Las propiedades podrían haber admitido otras restricciones de acuerdo a su tipo primitivo, tal como restringir `magnitude` con una lista de valores.

---

```

ELEMENT[at0006] occurrences matches {0..1} matches {
  value matches {
    C_DV_QUANTITY <
      list = <["1"] = <
        units = <'cm'>
        magnitude = <-5.0..5.0>
        precision = <0> > ...
    >
  }
}

```

---

```

DV_QUANTITY
  and (units value 'cm')
  and (magnitude only xsd:double[>= -5.0, <= 5.0])
  and (precision value 0)

```

---

### 5.4.2.3. Extensión para la validación respecto al modelo de referencia

La validación de la consistencia de un arquetipo en relación al modelo de referencia tiene unos requisitos diferentes a la validación de la relación de especialización. Por una parte, ciertas definiciones del modelo de arquetipos no son aplicables al modelo de referencia, como por ejemplo la restricción de ocurrencias. Por otra parte, la definición del modelo de referencia haciendo uso de axiomas *SubClassOf* mientras que los arquetipos son representados con axiomas de equivalencia requiere que ciertas restricciones, como por ejemplo la cardinalidad, deban ser expresadas de distinto modo para cada tipo de validación.

En este apartado se analizan las definiciones del modelo de arquetipos que requieren un tratamiento especial para la validación respecto al modelo de referencia. El motivo es que la técnica para la validación de la relación de especialización se basa en la inferencia de axiomas de inclusión, mientras que la validación respecto al modelo de referencia consiste en detectar clases inconsistentes. En primer lugar, el axioma OWL que restringe la cardinalidad de una propiedad no es el adecuado si queremos que una clase sea inconsistente al declarar una restricción de cardinalidad incorrecta según el modelo de arquetipos. En segundo lugar, la declaración de los términos de un arquetipo implícitamente supone cambiar el rango de un atributo. Por tanto, es necesario validar que la definición de los términos arquetipados sean conformes con los tipos de los atributos en el modelo de referencia.

```
# Axiomas de inclusión

Class: CLASE
  SubClassOf: prop max 2

Class: ARQUETIPO
  SubClassOf: CLASE, prop max 3

-----

# Axiomas de equivalencia

Class: CLASE
  EquivalentTo: prop max 2

Class: ARQUETIPO
  EquivalentTo: prop max 3
```

Figura 5.8: Comparación axiomas de inclusión y equivalencia en OWL

### Restricción de cardinalidad

El axioma propuesto en el apartado anterior para representar la restricción de cardinalidad permite obtener las inferencias esperadas cuando forma parte de un axioma de equivalencia. Sin embargo, si la clase respecto a la que se realiza la validación no se expresa también con un axioma de equivalencia, la representación ya no sería útil. En otras palabras, el axioma OWL de cardinalidad expresa correctamente la definición de cardinalidad del modelo de arquetipos para validar la relación de especialización, pero no es la representación adecuada si se valida la consistencia respecto al modelo de referencia, ya que éste se define utilizando axiomas *SubClassOf*.

Este problema se ilustra en la figura 5.8 que muestra la definición de una clase OWL que restringe la cardinalidad máxima de una propiedad y una subclase que especializa esta restricción estableciendo una restricción incompatible con la definición de la clase padre. Un razonador OWL no podría clasificar la clase `ARQUETIPO` como inconsistente, ya que existen interpretaciones del modelo que permiten que la clase tenga instancias. En cambio, si ambas clases se expresaran con un axioma de equivalencia se obtendría que la clase `CLASE` no subsume a la clase `ARQUETIPO`, lo que se interpreta como que la restricción es incorrecta. Por tanto, no es posible utilizar la restricción de cardinalidad de OWL para validar la restricción de cardinalidad en ADL utilizando axiomas de inclusión.

La representación del modelo de referencia es extendida para satisfacer el requisito de validación de las restricciones de cardinalidad. Se define en OWL la clase `CARDINALITY`, que representa un intervalo de cardinalidad utilizando las propiedades, `attribute`, `min_value` y `max_value`, y la propiedad `cardinality` que asocia la definición de las clases del modelo de referencia con las cardinalidades de sus atributos. La figura 5.9 ilustra el uso de estas construcciones para definir la cardinalidad mínima de un atributo. Se emplea el cuantificador universal para restringir las cardinalidades de todos los atributos de `CLASE`. Cada definición de cardinalidad se expresa en OWL como una clase anónima que emplea la restricción de valor para asociar el nombre del atributo y el cuantificador universal para establecer el valor de la cardinalidad. En el arquetipo se utiliza un axioma con una estructura similar en el que la propiedad `cardinality` es restringida con el cuantificador existencial, mientras que la propiedad `min_value` se fija a un valor concreto.

```
Class: CLASE
  SubClassOf:
    cardinality only (
      (attribute value 'atributo1' and
        min_value only xsd:int[>= 1])
      or ...
      # otras restricciones de cardinalidad
    )
    ...

Class: ARQUETIPO
  EquivalentTo: ... CLASE and
    cardinality some (
      attribute value 'atributo1' and
        min_value value 2)
```

Figura 5.9: Axiomas OWL para validación cardinalidad en Archeck

### Restricción de rango

Una característica destacada del modelo de arquetipos es que permite restringir el tipo de un atributo de acuerdo a las definiciones de nuevos conceptos, esto es, los términos arquetipados. Desde el punto de vista de la consistencia con el modelo de referencia, cada uno de estos nuevos conceptos debe ser compatible con el tipo del atributo. El tipo de un atributo se define en el modelo de referencia utilizando un restricción basada en el

cuantificador universal de OWL. Así pues, se violaría la restricción de rango si indicamos que el atributo es redefinido para permitir un término arquetipado que no es compatible con el tipo del atributo (restricción basada en el cuantificador existencial, *some*).

La figura 5.10 muestra la definición OWL de la clase `CLUSTER_at0000` del ejemplo de arquetipo que combina las restricciones para la validación de la consistencia de la relación especialización en un axioma de equivalencia y para la validación respecto al modelo de referencia en axiomas subclass. Nótese que todas las declaraciones podrían formar parte del axioma de equivalencia. Sin embargo, se ha preferido separar las definiciones para diferenciar las dos representaciones.

```

Class: CLUSTER_at0000

# Validación especialización arquetipos
EquivalentTo:
  ARCHETYPED_CLASS
  and CLUSTER
  and CONCEPT_at0000
  and CLUSTER_items only UNORDERED
  and CLUSTER_items only UNIQUE
  and CLUSTER_items min 1 ITEM
  and CLUSTER_items max 1 CLUSTER_at0001
  and CLUSTER_items min 1 ELEMENT_at0002

# Validación respecto al modelo de referencia
SubClassOf:
  CLUSTER,
  CLUSTER_items some CLUSTER_at0001,
  CLUSTER_items some ELEMENT_at0002,
  CLUSTER_items some CLUSTER_at0003,
  cardinality some (attribute value 'items'
    and min_value value 1)

```

Figura 5.10: Definición completa del arquetipo en Archeck

## 5.5. Encorsetable

Encorsetable es una metodología para el intercambio de modelos clínicos entre estándares basados en la arquitectura dual. El marco ontológico de esta metodología está organizado en torno a una ontología denominada *Ontología de Restricciones*. Los modelos de información son definidos especializando las clases de la Ontología de Restricciones. El resultado es la obtención de ontologías que representan modelos de información *arquetipables*, es decir, ontologías cuyas instancias definen modelos clínicos o arquetipos.

La metodología Encorsetable puede considerarse una evolución del framework Poseacle. La diferencia más notable entre ambas propuestas es el modo en el que se representa el modelo de referencia. En Poseacle, la representación en OWL de los modelos de referencia es fiel a la semántica de las estructuras de datos. En cambio, los modelos de referencia definidos en la metodología Encorsetable son modelos de restricciones que no expresan la semántica de los datos, sino las restricciones que caracterizan las estructuras de datos del modelo de referencia. Esta diferencia permite que los arquetipos puedan definirse como instancias de las ontologías de los modelos de referencia o como ontologías que especializan la semántica de los modelos de información. Esta doble representación es útil en el proceso de transformación de modelos clínicos entre estándares.

### 5.5.1. Ontología de Restricciones

La Ontología de Restricciones define los conceptos necesarios para expresar los modelos de un dominio utilizando restricciones. Los conceptos fundamentales sobre los que se apoya la ontología son `C_TYPE`, `C_PROPERTY` y `C_RANGE`, que representan las restricciones aplicables a tipos de datos, propiedades y rangos, respectivamente. Los tipos de datos restringidos se clasifican en dos grupos: complejos y primitivos. Los primeros se definen restringiendo propiedades (`C_PROPERTY`), en concreto, la cardinalidad, unicidad y orden de las asociaciones. Los tipos primitivos corresponden a los tipos básicos de UML o XML, es decir, enteros, cadenas, etc. Por ejemplo, la clase `C_STRING` representa las restricciones sobre cadenas y puede ser asociada con expresiones regulares, con un valor por defecto, etc. Los rangos (`C_RANGE`) declaran los tipos de datos que pueden ser asociados a través de las propiedades y establecen el número de ocurrencias válidas para estas asociaciones. La figura 5.11 muestra un extracto de la Ontología de Restricciones. En los ejemplos que se desarrollan a continuación se utiliza el prefijo "co:" para hacer referencia a los conceptos de esta ontología.

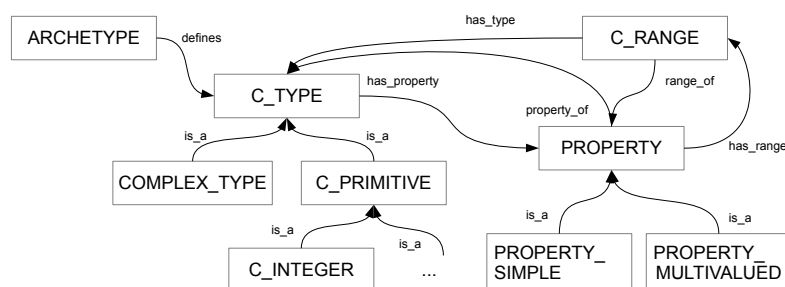


Figura 5.11: Ontología de Restricciones

### 5.5.2. Representación del Modelo de Referencia

El modelo de referencia es una ontología definida a partir de los conceptos proporcionados por la Ontología de Restricciones. La estructuras de datos del modelo de información corresponden con tipos de datos complejos y los atributos con propiedades restringidas. El resultado es un modelo de información *restringible*, que puede ser instanciado o especializado para definir arquetipos. La ontología es obtenida automáticamente aplicando las siguientes reglas que son ilustradas con las entidades de la jerarquía ITEM de openEHR y donde el prefijo “co:” hace referencia a la ontología de restricciones:

- Todas las clases del modelo de información se declaran subclases de C\_COMPLEX\_TYPE, es decir, son tipos de datos complejos.

---

```

Class: ITEM
  SubClassOf: co:COMPLEX_TYPE
  
```

```

Class: CLUSTER
  SubClassOf: co:COMPLEX_TYPE
  
```

```

Class: ELEMENT
  SubClassOf: co:COMPLEX_TYPE
  
```

---

- La relación de herencia en el modelo de información se traduce a OWL como axiomas de inclusión (*SubClassOf*).

---

```

Class: CLUSTER
  SubClassOf: ITEM
  
```

```

Class: ELEMENT
  SubClassOf: ITEM
  
```

---

- Un atributo se define como una clase que especializa `C_PROPERTY`, esto es, una propiedad restringible. Si el atributo es multivaluado, la clase deriva de `C_PROPERTY_MULTIVALUED`, en caso contrario, es subclase de `C_PROPERTY_SIMPLE`.

---

```
Class: ITEM_name
  SubClassOf: co:C_PROPERTY_SIMPLE

Class: CLUSTER_items
  SubClassOf: co:C_PROPERTY_MULTIVALUED
```

---

- La relación entre un tipo de datos y los atributos que contiene se establece con la propiedad OWL `has_property` restringida con el cuantificador existencial.

---

```
Class: ITEM
  SubClassOf: co:has_property some ITEM_name

Class: CLUSTER
  SubClassOf: co:has_property some CLUSTER_items
```

---

- El tipo de un atributo corresponde con una definición de rango (`C_RANGE`) que se relaciona con la declaración del atributo utilizando la propiedad OWL `has_range`. Así mismo, el rango se asocia con el tipo mediante la propiedad `has_type`. Si el tipo es primitivo, se establece la clase con las restricciones del tipo primitivo, como por ejemplo, `C_STRING` para las cadenas.

---

```
Class: ITEM_name
  SubClassOf: co:has_range only (co:has_type only co:C_STRING)

Class: CLUSTER_items
  SubClassOf: co:has_range only (co:has_type only ITEM)
```

---

- La cardinalidad de un atributo multivaluado se expresa restringiendo las propiedades de cardinalidad (`min_cardinality` y `max_cardinality`) de la clase `C_PROPERTY_MULTIVALUED`.

---

```
Class: CLUSTER_items
  SubClassOf: co:min_cardinality only xsd:int[>= 1]
```

---

- Los atributos heredados vuelven a declararse como nuevas clases que establecen una relación de subclase con la definición heredada y especializan la relación con el contenedor (propiedad `property_of`, inversa de `has_property`).

---

```
Class: CLUSTER_name
  SubClassOf:
    ITEM_name,
    co:property_of only CLUSTER

Class: ELEMENT_name
  SubClassOf:
    ITEM_name,
    co:property_of only ELEMENT
```

---

En la ontología resultante todas las entidades del modelo de referencia (tipos de datos y atributos) son definidas utilizando clases OWL. Los axiomas que definen a las clases emplean los cuantificadores universal y existencial para establecer las restricciones utilizando las propiedades de la ontología de restricciones (`has_property`, `has_range`, etc.). Se utiliza el cuantificador universal cuando la propiedad que se está restringido es funcional. Por ejemplo, la propiedad OWL `property_of` asocia una propiedad restringible con el tipo de datos en el que está definida. En este caso el axioma que establece la asociación restringe la propiedad con el cuantificador universal (*only*). En cambio, la propiedad OWL `has_property` relaciona tipos de datos complejos con propiedades restringibles. El cuantificador universal no sería apropiado ya que obligaría a declarar en el mismo axioma todas las propiedades de la clase y de sus descendientes. El cuantificador existencial (*some*) expresa la relación sin limitar que los tipos de datos descendientes puedan añadir nuevos atributos.



### 5.5.3. Representación de arquetipos con instancias

La ontología del modelo de referencia no expresa la semántica de las estructuras de información, tal como sucede en la representación Poseacle o Archeck, sino que define un modelo de información *arquetipable*. Esto permite que los arquetipos se declaren como instancias de las clases de esta ontología.

La convención de nombrado empleada es la siguiente: (1) el identificador de un término arquetipado corresponde con el código *at*; (2) el identificador de un atributo está formado por el nombre del término contenedor y el atributo (por ejemplo, *at0000\_items*); y (3) los rangos se nombran como la composición del término contenedor, atributo y término que actúa de rango (por ejemplo, *at0000\_items\_at0001*).

Las reglas de traducción de la definición de un arquetipo consisten en la instanciación del modelo de referencia del siguiente modo:

- La clase del modelo de referencia que restringe el término arquetipado corresponde con el tipo de la instancia.

---

ADL

---

```
CLUSTER[at0000] matches { ...
```

---

OWL

---

```
Individual: at0000
  Types: openehr:CLUSTER
```

---

- Los atributos restringidos son instancias de las clases que representan al atributo en el modelo de referencia. Los términos arquetipados y los atributos que restringen se relacionan a través de la propiedad *has\_property*.

---

ADL

---

```
CLUSTER[at0000] matches {
  items ... { ...
```

---

OWL

---

```
Individual: at0000_items
  Types: openehr:CLUSTER_items

Individual: at0000
  Facts: co:has_property at0000_items
```

---

- Las restricciones sobre atributos multivaluados como la cardinalidad, el orden y la unicidad se definen utilizando las propiedades asociadas a la clase `C_PROPERTY_MULTIVALUED`.

---

ADL

---

```
CLUSTER[at0000] matches {
  items cardinality matches {1..*; unordered; unique} matches {
    ...
  }
}
```

---

OWL

---

```
Individual: at0000_items
  Facts:
    co:min_cardinality 1,
    co:order false,
    co:unique true
```

---

- Los términos arquetipados indican los atributos a los que están vinculados mediante un rango en el que expresan el número de ocurrencias. La relación con su atributo contenedor se establece con la propiedad `range_of`, que es la inversa de la relación `has_range`.

---

ADL

---

```
CLUSTER[at0000] matches {
  items ... matches {
    CLUSTER[at0001] occurrences matches {0..1} matches { ...
  }
}
```

---

OWL

---

```
Individual: at0000_items_at0001
  Types: co:RANGE
  Facts:
    co:has_type at0001,
    co:min_occurrences 0,
    co:max_occurrences 1,
    co:range_of at0000_items
```

---

#### 5.5.4. Representación de arquetipos como clases

La semántica de las restricciones de un arquetipo puede expresarse a través de la declaración de una clase OWL que especialice la clase correspondiente del modelo de referencia. Todas las restricciones se definen mediante axiomas de inclusión (subclase). Las restricciones del modelo de arquetipos se traducen en los siguientes axiomas OWL:

- Un término arquetipado es subclase de la clase del modelo de referencia que especializa.

---

ADL

---

```
CLUSTER[at0000] matches { ...
```

---

OWL

---

```
Class: at0000
  SubClassOf: openehr:CLUSTER
```

---

- Los atributos redefinidos por un término se declaran como una clase que es subclase de la declaración del atributo en el modelo de referencia. Se relaciona con el término utilizando la propiedad `property_of` (inversa de `has_property`). La relación se define con el cuantificador universal.

---

ADL

---

```
CLUSTER[at0000] matches {
  items ... { ...
```

---

OWL

---

```
Class: at0000_items
  SubClassOf:
    openehr:CLUSTER_items,
    co:property_of only at0000
```

---

- La cardinalidad de los atributos multivaluados se declara acotando el rango de valores permitidos para la cardinalidad mínima y máxima. Las restricciones de orden y unicidad son cerradas, es decir, el valor establecido debe ser fijo y no permite que sea modificado por un arquetipo descendiente. Por tanto, son representadas con un axioma que fija el valor al que puede estar asociado.

---

ADL

---

```

CLUSTER[at0000] matches {
  items cardinality matches {1..*; unordered; unique} matches {
    ...
  }
}

```

---

OWL

---

```

Class: at0000_items
SubClassOf:
  co:min_cardinality only int[>= 1],
  co:order value false,
  co:unique value true

```

---

- Cada nuevo término declarado en el rango del atributo se declara con una clase que es subclases de `C_RANGE`. Al igual que en la construcción del modelo de referencia, la clase se liga a la declaración del atributo utilizando la propiedad OWL `range_of`. El tipo del rango se establece con la propiedad `has_type`. Por último, las ocurrencias de un término son definidas restringiendo los valores de las propiedades `min_occurrences` y `max_occurrences` en los rangos en los que es declarado.

---

ADL

---

```

CLUSTER[at0000] matches {
  items ... matches {
    CLUSTER[at0001] occurrences matches {0..1} matches { ...
  }
}

```

---

OWL

---

```

Class: at0000_items_at0001
SubClassOf:
  co:C_RANGE,
  co:has_type only at0001,
  co:range_of only at0000_items,
  co:min_occurrences only int[>= 0],
  co:max_occurrences only int[<= 1]

```

---

### 5.5.5. Restricciones sobre tipos primitivos

La ontología de restricciones contiene varias clases para restringir tipos primitivos, como por ejemplo `C_STRING` para el tipo cadena de caracteres. Se ha declarado la propiedad `assumed_value` para almacenar la restricción del valor por defecto. También se ha definido la propiedad `candidate` para establecer una lista de valores posibles. Estas dos propiedades son comunes a todos los tipos primitivos. Los tipos numéricos y los tipos temporales incluyen las propiedades `min_value` y `max_value` para establecer un rango de valores. Por último, los tipos cuya representación puede estar restringida por una expresión regular hacen uso de la propiedad `pattern`.

A continuación se incluye una selección de los ejemplos introducidos en la sección 5.2 que ilustra el uso de las propiedades que expresan las restricciones. Cada ejemplo muestra tanto la representación como instancias y como clases. Además, el código OWL está simplificado debido a que el uso de propiedades restringibles y rangos hace que sean necesarias varias instancias o clases anónimas para expresar una restricción.

**Cadena conforme a una expresión regular.** La clase `C_STRING` representa las restricciones sobre el tipo primitivo cadena de caracteres. La propiedad `pattern` se utiliza para establecer un patrón de expresión regular.

---

ADL

---

```
nombre matches {/a\d{3}/}
```

---

OWL instancias

---

```
Individual: C_STRING_1
  Types: co:C_STRING
  Facts: co:pattern 'a\d{3}'
```

---

OWL clases

---

```
Class: C_STRING_1
  SubClassOf:
    co:C_STRING,
    co:pattern only string[pattern 'a\d{3}']
```

---

**Enumeración de cadenas.** Todos los tipos primitivos pueden ser restringidos con una enumeración de valores. La propiedad `candidate` establece la asociación con los componentes de la enumeración.

---

ADL

---

```

unidad matches {'cm', 'm'}

```

---

OWL instancias

---

```

Individual: C_STRING_1
  Types: co:C_STRING
  Facts:
    co:candidate 'cm',
    co:candidate 'm'

```

---

OWL clases

---

```

Class: C_STRING_1
  SubClassOf:
    co:C_STRING,
    co:candidate only {'cm', 'm'}

```

---

**Intervalo numérico.** Los tipos numéricos aceptan las restricciones de intervalo que se establecen con las propiedades `min_value` y `max_value`. La ontología de restricciones declara el tipo `C_INTEGER` para definir las restricciones sobre valores enteros.

---

ADL

---

```

medida matches {3..5}

```

---

OWL instancias

---

```

Individual: C_INTEGER_1
  Types: co:C_INTEGER
  Facts:
    min_value 3,
    max_value 5

```

---

OWL clases

---

```

Class: C_INTEGER_1
  SubClassOf:
    co:C_INTEGER,
    min_value only int[>= 3],
    max_value only int[<= 5]

```

---

## 5.6. Comparación de las representaciones

En este capítulo se han presentado tres propuestas que utilizan OWL DL para representar el formalismo de arquetipos. Cada representación es utilizada en distintas tareas relativas a la gestión de arquetipos. Las ontologías de *Poseacle* han sido utilizadas para construir la herramienta *Poseacle Converter* y son explotadas en el gestor de arquetipos *ArchMS* para la clasificación de arquetipos, anotaciones de recursos formativos y búsquedas semánticas. La representación *Archeck* expresa la semántica formal del modelo de arquetipos y se ha utilizado como soporte de un método de validación basado en las inferencias obtenidas a partir de la representación. La propuesta *Encorsetable* ha sido empleada en la definición de un marco de interoperabilidad de modelos clínicos entre distintos estándares.

En este contexto las tres representaciones propuestas son semánticamente equivalentes. Es posible obtener una representación a partir de otra a través de su equivalencia con el modelo de arquetipos. Por ejemplo, un arquetipo OWL definido según las ontologías *Poseacle* puede ser transformado a la representación *Archeck* a través de la transformación del arquetipo OWL en ADL implementada en *Poseacle Converter*. En el contexto de *Encorsetable* también ha sido necesario intercambiar las representaciones de clases e instancias. En este caso, la equivalencia semántica se ha definido a través del modelo de arquetipos (AOM).

La definición del modelo de arquetipos en OWL requiere traducir el modelo de referencia también a OWL. Las tres representaciones definen de forma diferente los modelos de información y esto se debe a que estas definiciones dependen de la representación de los arquetipos. Por un lado, la propuesta de *Archeck* es la más fiel a la semántica del modelo de información, ya que ha sido definida siguiendo las recomendaciones recogidas en ODM. Por otro lado, aunque las ontologías de *Poseacle* fueron creadas antes de la publicación de ODM, cumplen en gran medida con las reglas de ODM. A pesar de ello, los modelos de referencia de *Archeck* y *Poseacle* tienen propósitos distintos. En *Poseacle* se definen clases *arquetipables*, es decir, los arquetipos son instancias de las clases del modelo de referencia. En cambio, en *Archeck* los arquetipos son clases que especializan el modelo de referencia. En cualquier caso, los modelos de referencia *Archeck* y *Poseacle* podrían ser utilizados para crear instancias de extractos de HCE, aunque con algunas limitaciones. Por último, la representación propuesta en *Encorsetable* es diferente a las otras dos, ya que no soporta la instanciación de extractos HCE. En la metodología *Encorsetable* los modelos de referencia se definen utilizando los constructores de la ontología de restricciones, cuyo propósito no es representar datos, sino restricciones de datos.





## Capítulo 6

# Calidad de arquetipos

### 6.1. Introducción

Los arquetipos han sido propuestos para tener un papel destacado en el funcionamiento de las aplicaciones sanitarias, en concreto, en relación a la interacción con los usuarios, ya que guían la introducción, presentación y validación de la información clínica. Por este motivo, los arquetipos deben ser considerados un activo de conocimiento que ha de ser incorporado en el diseño de las aplicaciones clínicas, al igual que otros recursos de conocimiento en el ámbito sanitario como los sistemas terminológicos. Sin embargo, para que los arquetipos sean aceptados y adoptados ampliamente en los sistemas de información sanitarios deben tener una calidad demostrable.

En este capítulo se describe el método *Archeck* de validación de arquetipos en las arquitecturas de HCE basadas en el modelo dual. El modelo de referencia y los arquetipos son representados como ontologías OWL que constituyen una base de conocimiento. El uso de razonadores OWL permite obtener inferencias a partir de la base de conocimiento que son analizadas para identificar definiciones de arquetipos incorrectas respecto al modelo de referencia y relaciones de especialización entre arquetipos inconsistentes.

El método de validación de arquetipos *Archeck* tiene un doble objetivo. En primer lugar, define un marco formal de representación de arquetipos que integra el modelo de información, los arquetipos y las terminologías clínicas. Los arquetipos se relacionan con el modelo de referencia y las terminologías a través de relaciones semánticas que permiten validar su consistencia. En segundo lugar, define formalmente la relación de conformidad entre arquetipos y propone una representación capaz de validar la relación de especialización del modelo de arquetipos.

La representación formal de arquetipos está basada en lógicas descriptivas, en particular, en la lógica descriptiva del lenguaje OWL 2 DL. La validación de un arquetipo se plantea desde dos perspectivas. Por un lado, es necesario comprobar que la definición de un arquetipo sea correcta

respecto al modelo de referencia, es decir, que los conceptos del arquetipo restringen clases correctas del modelo de referencia y que las restricciones que establecen sean consistentes. Por otro lado, la relación de conformidad de arquetipos exige que los extractos de HCE que representa un arquetipo especializado pueden ser tratados como extractos del arquetipo base. En este capítulo se utiliza la representación de arquetipos presentada en la sección 5.4 que integra las necesidades de validación respecto al modelo de referencia y de la relación de especialización.

El método de validación ha sido aplicado a los arquetipos de los dos repositorios más importantes en la actualidad: Clinical Knowledge Manager (CKM), soportado por la Fundación openEHR, y el repositorio desarrollado en el marco del programa NHS Connection for Health en Reino Unido. El resultado de la validación de ambos repositorios señala que una quinta parte de las relaciones de especialización entre arquetipos no son consistentes y que ciertos tipos de errores se distribuyen de distinto modo en ambos repositorios.

Finalmente, se introduce un marco de trabajo para la definición e implementación de métricas de calidad basado en las representaciones OWL de los arquetipos. La representación OWL del método Archeck es utilizada para validar la consistencia de los enlaces terminológicos en especializaciones. La representación de arquetipos como instancias propuesta por el framework Encorsetable se utiliza para cuantificar la calidad de un enlace terminológico en relación a los enlaces de otros términos definidos en su contexto.

## 6.2. Validación de las restricciones

La arquitectura de modelo dual propone que los modelos clínicos sean definidos como restricciones de las declaraciones de las entidades de un modelo de referencia. Un término de un arquetipo indica la clase del modelo de información que especializa y declara los atributos que va a redefinir. La figura 6.1 muestra la definición ADL del concepto raíz arquetipo *Examen* presentado en la sección 3.3. El concepto raíz especializa la clase `CLUSTER` redefiniendo el atributo `items` para incluir los conceptos: hechos normales, descripción clínica, hallazgos y una serie de imágenes.

La redefinición de los atributos consiste en modificar las características de un atributo del modelo de referencia. La validación de las nuevas definiciones depende de la característica que esté siendo modificada, por ejemplo, la sustitución del tipo de un atributo se valida comprobando que el nuevo tipo sea conforme al declarado en el modelo de referencia. Además, no todas las definiciones del modelo de arquetipos corresponden con características del modelo de referencia. Entre otras, la restricción de ocurrencias no está soportada por el lenguaje UML con el que se definen los modelos

```
-- Examen
CLUSTER[at0000] matches {
  items cardinality matches {1..*; unordered} matches {
    -- Hechos normales
    CLUSTER[at0001] occurrences matches {0..1} matches { ... }
    -- Descripción clínica
    ELEMENT[at0002] occurrences matches {0..1} matches { ... }
    -- Hallazgos
    CLUSTER[at0003] occurrences matches {0..1} matches { ... }
    -- Imágenes
    ELEMENT[at0004] occurrences matches {0..*} matches { ... }
  }
}
```

Figura 6.1: Fragmento del arquetipo *Examen* de openEHR

de referencia. Estas definiciones son comprobadas en la validación de la consistencia de la relación de especialización.

### 6.2.1. Uso correcto de las entidades del modelo de referencia

En la sección 3.5 se discuten algunas limitaciones del modelado de arquetipos en ADL. Entre ellas, la relación que establece un término de un arquetipo con el modelo de referencia está basada en cadenas, esto es, no hay ninguna relación semántica entre el modelo de información y el arquetipo. El método Archeck propone la representación de los términos de un arquetipo utilizando clases OWL que se relacionan con el modelo de referencia mediante un axioma de inclusión (*SubClassOf*) sobre la clase que está redefiniendo. Esta relación semántica es el soporte para la validación de las definiciones de un arquetipo, aunque no garantiza que la relación se establezca con una entidad correcta del modelo de referencia. La figura 6.2 muestra el concepto raíz de un arquetipo openEHR que referencia una clase incorrecta del modelo de información (*GRUPO* no es un tipo de datos válido en openEHR). Esto implica que la representación correspondiente en OWL hace uso de una clase que no existe en la ontología del modelo de referencia. Sin embargo, este error no es detectado como un error sintáctico en OWL (*clase no definida*) o una inconsistencia de la representación OWL. El modelo de razonamiento de OWL trabaja con la asunción de *mundo abierto*, lo que permite que el conocimiento definido sea incompleto. En otras palabras, que la ontología de un arquetipo declare una relación de inclusión con una clase que no está definida en la ontología del modelo de referencia no es interpretado como una inconsistencia en OWL.

---

ADL

---

```
GRUPO[at0000] matches {
  items cardinality matches ...
}
```

---

OWL

---

```
Class: GRUPO_at0000
SubClassOf:
  openehr:GRUPO,
  ...
```

---

Figura 6.2: Ejemplo de uso de entidades no definidas en el modelo de referencia

La solución a este problema puede ser abordada con dos técnicas diferentes:

- **Análisis de axiomas.** Las entidades no declaradas pueden ser identificadas procesando los axiomas de la ontología del modelo de referencia.
- **Inferencia.** Las entidades del modelo de referencia (clases y atributos) podrían ser clasificadas utilizando varias categorías. Por ejemplo, la categoría `RM_CLASS` podría representar las clases del modelo de información y todas las clases del modelo de referencia se declararían pertenecientes a esa categoría. Un término arquetipado que especifique una clase declarada en el modelo de referencia será clasificado como `RM_CLASS` utilizando un proceso de razonamiento. Por tanto, si no se infiere este axioma de inclusión, la clase del modelo de referencia no sería correcta.

Las dos propuestas son igualmente efectivas para detectar las entidades no declaradas. Se ha elegido la técnica basada en el procesamiento de inferencias debido a que es la que se utiliza en el resto de tareas del proceso de validación, y por tanto, se integra mejor en el método de validación Archeck.

La detección de atributos no declarados requiere introducir una modificación en la representación OWL de Archeck. En concreto, las restricciones de un atributo se declaran como clases anónimas que restringen la propiedad OWL que define el atributo en el modelo de referencia. La figura 6.3 muestra un extracto de la clase OWL que representa el concepto raíz

del arquetipo *Examen*. En ella podemos apreciar que las restricciones de orden y unicidad se definen restringiendo la propiedad `CLUSTER_items` que representa el atributo `items` de la clase `CLUSTER` de `openEHR`. Por tanto, la redefinición incorrecta de un atributo del modelo de referencia no es detectada utilizando la técnica de inferencia, ya que no se declaran nuevas propiedades para restringir los atributos de un término arquetipado.

```
Class: CLUSTER_at0000
SubClassOf:
  openehr:CLUSTER,
  openehr:CLUSTER_items only ORDERED,
  openehr:CLUSTER_items only NON_UNIQUE,
  ...
```

Figura 6.3: Extracto del concepto raíz arquetipo *Examen* en OWL

La representación OWL de Archeck es extendida para declarar una propiedad por cada atributo redefinido en los términos del arquetipo. La figura 6.4 presenta la propiedad asociada a la redefinición del atributo `items` del término `at0000`. La propiedad se declara subpropiedad de la propiedad que corresponde con el atributo que restringe el arquetipo (`CLUSTER_items`). En la ontología del modelo de referencia se declara la propiedad `OBJECT_RM_PROPERTY`, que representa a los atributos cuyo tipo sea una clase OWL, y la propiedad `DATA_RM_PROPERTY` que corresponde con la categoría de los atributos de tipo primitivo. De este modo, la técnica de inferencia detecta que un atributo no es declarado en el modelo de referencia si la propiedad no es clasificada como `OBJECT_RM_PROPERTY` o `DATA_RM_PROPERTY`, según corresponda. Por último, nótese que estas propiedades OWL se utilizan sólo como soporte para la identificación de atributos no definidos en el modelo de referencia y no se emplean en la definición OWL de los términos arquetipados.

```
ObjectProperty: CLUSTER_at0000_items
  SubPropertyOf: openehr:CLUSTER_items
  Domain: CLUSTER_at0000
  Range: openehr:ITEM
```

Figura 6.4: Propiedad de apoyo para detección de atributos no declarados

En resumen, la detección de entidades no declaradas en un arquetipo está basada en el procesamiento de las inferencias obtenidas de la ontología que define el arquetipo. Las clases y atributos del modelo de referencia son clasificados en categorías, según el tipo de declaración que representan, que ayudan a identificar el uso de entidades incorrectas en el arquetipo. Las clases y atributos restringidos en un arquetipo son correctos si se infiere que pertenecen a alguna de las categorías del modelo de referencia.

### 6.2.2. Redefinición de atributos

El método de validación de las redefiniciones de los atributos se basa en comprobar si las clases OWL que definen los términos arquetipados son consistentes con el modelo de referencia. Una clase OWL es inconsistente si el razonador infiere que la clase es clasificada como `owl:Nothing`. Esto es interpretado como que la definición de la clase es incorrecta, ya que las instancias de esa clase harían que la base de conocimiento fuera inconsistente. Por tanto, un arquetipo está correctamente definido respecto al modelo de referencia si todas las clases OWL de su ontología son consistentes.

La estrategia utilizada para validar las restricciones del arquetipo se basa en la combinación de la relación de inclusión (axioma *SubClassOf*), la restricción de propiedades con el cuantificador universal (*only*) y la restricción de propiedades con el cuantificador existencial (*some*). La clase que define el término arquetipado se relaciona con la clase del modelo de referencia mediante un axioma *SubClassOf*. En el modelo de referencia las definiciones de los atributos emplean el cuantificador universal y las redefiniciones en los arquetipos utilizan el cuantificador existencial. En la sección 5.4 se desarrolla en detalle la correspondencia de cada definición ADL en OWL.

En este apartado se desarrolla la técnica de validación de las restricciones de un atributo de un término arquetipado respecto al modelo de referencia. La figura 6.5 presenta un fragmento de la clase OWL que corresponde al tipo de datos `CLUSTER` de `openEHR`, que incluye la definición del atributo `items`. El primer axioma declara el tipo del atributo (`ITEM`) y el segundo axioma emplea una clase anónima para definir la cardinalidad

del atributo utilizando el cuantificador universal de OWL. Este tipo de datos será referido en los siguientes apartados para ilustrar los ejemplos de validación.

```
Class: CLUSTER
  SubClassOf: ITEM
  SubClassOf: CLUSTER_items only ITEM
  SubClassOf: cardinality only (
    (attribute value "items" and min_value only
     xsd:int[>= 0])
    # otros atributos
  )
  ...
```

Figura 6.5: Extracto de la representación OWL del tipo de datos CLUSTER de openEHR

#### 6.2.2.1. Restricción de cardinalidad

Desde el punto de vista del modelado de arquetipos, las restricciones que podemos establecer sobre la cardinalidad permiten:

- Eliminar un atributo opcional. Si el atributo se ha declarado en el modelo de referencia como opcional, es decir, tiene una cardinalidad mínima con valor 0, podemos indicar que el atributo no debe aparecer en el término arquetipado estableciendo la cardinalidad máxima en valor 0. De este modo, la cardinalidad del atributo quedaría en el rango 0..0, lo que en la práctica significa que no puede tomar valor.
- Establecer un atributo obligatorio. Un atributo opcional puede redefinirse para ser obligatorio aumentando la cardinalidad mínima. Por ejemplo, dado un atributo con cardinalidad 0..1, podemos hacer que sea obligatorio fijando la cardinalidad mínima en el valor 1.
- Modificar el intervalo de cardinalidad. En atributos multivaluados se puede restringir el rango de cardinalidad aumentando la cardinalidad mínima y/o disminuyendo la cardinalidad máxima.

La figura 6.6 muestra un fragmento de la clase OWL que representa el concepto raíz del arquetipo *Examen* que establece obligatoria la asociación a través del atributo `items`. La cardinalidad es declarada con una restricción existencial de la propiedad `cardinality` para establecer la cota inferior

de la cardinalidad del atributo “items”. La clase OWL se relaciona con el modelo de referencia utilizando un axioma *SubClassOf*.

```
Class: CLUSTER_at0000
  SubClassOf: openehr:CLUSTER,
  ...
  cardinality some (attribute value "items"
    and min_value value 1)
```

Figura 6.6: Representación OWL del concepto raíz del arquetipo *Examen*

La validación de la cardinalidad de un atributo asume que la cadena que identifica al atributo hace referencia a un atributo válido del término arquetipado, es decir, que el atributo es una entidad declarada en el modelo de referencia. La técnica para detectar si un arquetipo hace uso de entidades no declaradas es aplicada antes de la validación de las definiciones de un arquetipo. En el ejemplo de la figura 6.6, el atributo “items” es correcto, si la propiedad `CLUSTER_at0000_items` ha sido clasificada como `OBJECT_RM_PROPERTY`. Si es así, podemos asegurar que el axioma en el que se declara la cardinalidad del atributo `items` *redefine* correctamente el atributo del modelo de referencia.

La técnica de razonamiento que se aplica para validar la restricción de cardinalidad se basa en el uso del cuantificador universal de OWL en la declaración del atributo (véase el tipo de datos `CLUSTER` en la figura 6.5) y en el cuantificador existencial para la redefinición de la cardinalidad (véase figura 6.6). El cuantificador universal restringe los valores que puede tomar la cota mínima de la cardinalidad del atributo `items` (mayor o igual que 0), mientras que el cuantificador existencial declara que la cota mínima de la cardinalidad de ese atributo es 1. La clase OWL es consistente si el valor establecido para la cardinalidad no es contradictorio con la definición.

#### 6.2.2.2. Restricción de rango

La restricción de rango de un atributo consiste en establecer un nuevo tipo de datos para el atributo. El nuevo rango puede ser simple (un solo tipo) o múltiple, y se interpreta según la cardinalidad del atributo. A continuación se muestran varios casos:



**Sustitución de tipo.** Esta restricción se establece en atributos con cardinalidad máxima de 1, en los cuales se sustituye el tipo declarado por otro del modelo de referencia. El ejemplo que se muestra a continuación es la redefinición del rango del atributo `value` de la clase `ELEMENT` de `openEHR`. El tipo del atributo en el modelo de referencia es `DATA_VALUE`. En el arquetipo se sustituye este tipo por la clase `DV_TEXT`, que es subclase de `DATA_VALUE`.

---

```
ELEMENT[at0001] matches {
  value matches {
    DV_TEXT {*}
  }
}
```

---

**Tipos alternativos.** En atributos no multivaluados se puede definir el tipo de un atributo ofreciendo dos o más alternativas. Esta restricción se interpreta como que el atributo puede tomar valores de cualquiera de los dos tipos. Se muestra a continuación un ejemplo en el que se declaran dos tipos alternativos para el atributo `value` de la clase `ELEMENT` de `openEHR`. La única característica que tienen en común es que ambos deben ser compatibles con el tipo declarado del atributo en el modelo de referencia (`DATA_VALUE`).

---

```
ELEMENT[at0001] matches {
  value matches {
    DV_TEXT {*}
    DV_ORDINAL {*}
  }
}
```

---

**Rango arquetipado.** El rango de un atributo no tiene por qué corresponder con un tipo declarado en el modelo de referencia, sino que puede ser un término declarado en el arquetipo. Tanto en atributos simples como multivaluados es posible establecer como rango un término arquetipado. El ejemplo muestra la redefinición del rango del atributo simple `protocol` de la clase `OBSERVATION`. El rango establecido corresponde con el término arquetipado `at0002`. El tipo que restringe el término es `ITEM_TREE`, que es compatible con el tipo declarado del atributo `protocol` en el modelo de referencia (`ITEM_STRUCTURE`).

---

```
OBSERVATION[at0001] matches {
  protocol matches {
    ITEM_TREE[at0002] matches {
      items ...
    }
  }
}
```

---

**Rango múltiple.** Un término de un arquetipo puede restringir un atributo multivaluado de modo que pueda ser asociado con varios términos. Este caso corresponde con el ejemplo del concepto raíz del arquetipo *Examen*. El rango del atributo `items` está formado por 4 términos arquetipados. Una instancia del término `at0000` puede estar asociada a instancias de cualesquiera de los 4 términos. Una característica que distingue a los rangos múltiples es que cada término puede declarar el intervalo de instancias que pueden formar parte de la asociación (ocurrencias).

La redefinición del rango del concepto raíz del arquetipo *Examen* se muestra en la figura 6.7. El rango del atributo `items` está formado por 4 términos arquetipados. Para cada uno de ellos se incluye un axioma *SubClassOf* que establece una restricción existencial que relaciona el atributo (propiedad OWL `CLUSTER_items`) con las clases que representan a los términos arquetipados.

```
Class: CLUSTER_at0000
SubClassOf:
  CLUSTER,
  CLUSTER_items some CLUSTER_at0001,
  CLUSTER_items some ELEMENT_at0002,
  CLUSTER_items some CLUSTER_at0003,
  CLUSTER_items some CLUSTER_at0004
```

Figura 6.7: Restricción de rango del concepto raíz del arquetipo *Examen* en OWL

La definición propuesta para el rango permite validar los cuatro casos de redefinición comentados anteriormente. Para el ejemplo de la figura 6.7 el tipo del atributo `items` de la clase `CLUSTER` se declara con el cuantificador universal de OWL (véase figura 6.5), lo que se interpreta como que sólo se permiten instancias compatibles con la clase `ITEM`. La redefinición del rango en el arquetipo emplea una restricción existencial que indica que existirán instancias de cada clase que se declare en el rango. La restricción de rango es correcta si se obtiene la inferencia de que la clase OWL que declara el rango es subclase de `ITEM`.

Por último, nótese que la representación OWL propuesta debe interpretarse sólo para la validación de restricción de rango y no para la validación de datos. En otras palabras, la clase OWL que representa la raíz del arquetipo *Examen* no representa correctamente las instancias de este término arquetipado. En el ejemplo de la figura 6.7 se está diciendo que el término *at0000* se asocia obligatoriamente con instancias de cada uno de los rangos declarados para el atributo *items*. Esta interpretación no es correcta para realizar la validación de los datos, ya que, por ejemplo, el término *at0001* puede no tener instancias en esa asociación (valor mínimo de ocurrencia 0).

### 6.2.2.3. Restricción de ocurrencias

El lenguaje de modelado UML no incluye ninguna construcción para definir las ocurrencias de una clase como rango de un atributo. Ésta es una característica propia del modelo de arquetipos que no tiene equivalencia en un diagrama de clases UML (el tipo de un atributo es único). Sin embargo, la definición conjunta de ocurrencias de todos los términos que forman parte del rango de un atributo multivaluado debe ser consistente con la cardinalidad redefinida en el arquetipo, y a su vez, la cardinalidad del atributo debe ser consistente con el modelo de referencia. Así pues, la restricción de ocurrencias debe ser tenida en cuenta en esta etapa de validación.

La definición conjunta de las ocurrencias de un rango debe cumplir las siguientes restricciones:

1. La cardinalidad máxima del atributo debe ser mayor o igual que la suma de los valores mínimos de ocurrencias de los términos que definen el rango el atributo.
2. Si todos los términos del rango tienen acotado el número de ocurrencias máximo, la cardinalidad máxima del atributo debe ser mayor o igual que la suma de los valores máximos de ocurrencias de todos los términos.
3. La cardinalidad máxima de un atributo multivaluado no debe estar acotada si alguno de componentes que define el rango tiene un número de ocurrencias no acotado.

En el arquetipo *Examen* la cardinalidad máxima del atributo está bien definida (no está acotada), ya que el término *at0004* no limita el número de ocurrencias. Si ese término estuviera acotado con un valor máximo de 1, la cardinalidad máxima del atributo debería ser igual o mayor que 4 (suma de las ocurrencias máximas de los componentes del rango). La cardinalidad mínima del atributo también está bien definida debido a que es superior a la suma de las ocurrencias mínimas de todas las clases que definen el rango.

La representación OWL utilizada para validar la definición de ocurrencias está basada en la restricción de cardinalidad de las propiedades en OWL (véase figura 6.8). La cardinalidad del atributo se define mediante una restricción de cardinalidad no cualificada (no se indica el tipo), mientras que las ocurrencias de cada término son expresadas con una restricción de cardinalidad cualificada, es decir, se especifica la clase para la que se limitan las instancias en la asociación. La figura 6.8 corresponde a un fragmento del concepto raíz del arquetipo *Examen* con la restricción de cardinalidad mínima del atributo `items` (la cardinalidad máxima no se especifica porque no está acotada) y la definición de las ocurrencias máximas de los términos `at0001`, `at0002` y `at0003`. Nótese que no se incluye ningún axioma para declarar el valor 0 como cota inferior de las ocurrencias (no tiene sentido lógico) y que el término `at0004` no posee ningún axioma que declare sus ocurrencias, ya que el intervalo que declara es `0..*` (no acotado).

```

Class: CLUSTER_at0000
SubClassOf:
  CLUSTER_items min 1,
  CLUSTER_items max 1 CLUSTER_at0001,
  CLUSTER_items max 1 ELEMENT_at0002,
  CLUSTER_items max 1 CLUSTER_at0003,
  ...

```

Figura 6.8: Representación OWL para la validación conjunta de la restricción de ocurrencias

El uso de restricciones de cardinalidad en OWL no permite comprobar completamente las reglas de validación de las ocurrencias debido al modelo formal de la semántica de OWL. Si suponemos, por ejemplo, que el arquetipo que se muestra en la figura 6.1 establece una cota superior para la cardinalidad con un valor de 5, cumpliría con la regla 1 de la validación de ocurrencias (es superior a la suma de las ocurrencias mínimas de todos los componentes), pero no cumpliría la regla 3, ya que uno de los componentes tiene cardinalidad no acotada (`at0004`). Esto significa que se podrían construir extractos inconsistentes, y por tanto, el arquetipo sería incorrecto. En estas circunstancias, un razonador no detectaría la clase OWL `CLUSTER_at0000` como inconsistente, ya que existiría alguna interpretación en lógica descriptiva en la que la clase tendría instancias.

Para tratar esta limitación de OWL se introducen dos clases que representan dos categorías incompatibles, `BOUNDED` y `UNBOUNDED`, es decir, las clases son disjuntas y representan un rango acotado y no acotado, respec-

tivamente. En el caso en el que la cardinalidad máxima de un atributo sea acotada se introduce un axioma que establezca una restricción basada en el cuantificador universal sobre el atributo que permita sólo instancias *acotadas* (BOUNDED). Por otro lado, si alguno de los componentes no establece una cota superior de ocurrencias, se introduce un axioma basado también en el cuantificador universal que diga que el atributo es *no acotado* (UNBOUNDED). Así pues, se produciría una contradicción si la cardinalidad máxima está acotada y algún componente no acota el máximo número de ocurrencias. En la figura 6.9 se modifica la definición OWL de la figura 6.8 estableciendo que el atributo *items* tiene cardinalidad máxima 5. Se observa además en la figura que se han introducido dos axiomas incompatibles, por un lado la cardinalidad está acotada (BOUNDED) y, por otro lado, uno de los componentes tiene ocurrencias no acotadas (UNBOUNDED). La presencia de estos dos axiomas en la clase provoca que un razonador la clasifique como inconsistente.

```
Class: CLUSTER_at0000
SubClassOf:
  CLUSTER_items min 1,
  CLUSTER_items max 5,
  CLUSTER_items only BOUNDED,
  CLUSTER_items max 1 CLUSTER_at0001,
  CLUSTER_items max 1 ELEMENT_at0002,
  CLUSTER_items max 1 CLUSTER_at0003,
  CLUSTER_items only UNBOUNDED,
  ...
```

Figura 6.9: Representación OWL extendida para la validación conjunta de la restricción de ocurrencias

Por último, nótese que el tratamiento de la regla de validación de las ocurrencias está basado en un axioma para representar la cardinalidad distinto al que se ha presentado en el apartado anterior (uso de la clase *CARDINALITY*). El motivo es que el axioma de cardinalidad de OWL permite validar las ocurrencias de los términos de un atributo respecto a la definición de cardinalidad, siendo esta validación local a la clase en la que se define. En cambio, el uso de la clase *CARDINALITY* pretende validar la restricción de cardinalidad respecto al modelo de referencia. En la sección 5.4 que describe las reglas de representación de arquetipos OWL discute que los axiomas de cardinalidad OWL no son correctos para validar las restricciones de cardinalidad entre dos clases que se relacionan entre sí utilizando el axioma *SubClassOf*. Sin embargo, la validación de las ocurrencias se realiza en el

contexto de la clase y por tanto el uso de este tipo de axiomas es correcto. Asimismo, para la validación de la relación de especialización también puede utilizarse el axioma de cardinalidad OWL, ya que se emplea en clases definidas mediante un axioma de equivalencia.

### 6.2.3. Referencias internas y slots

Las dos construcciones proporcionadas por el modelo de arquetipos para reutilizar definiciones, esto es, referencias internas y slots, deben tenerse en cuenta para la validación de un arquetipo respecto al modelo de referencia. A pesar de que ambas definiciones no redefinen ninguna característica del modelo de referencia, son declaradas como rango de algún atributo, y por tanto, es necesario validar si esta definición es consistente.

En primer lugar, la noción de referencia interna no es propia de los modelos de clases en UML, ya que todas las clases tienen entidad propia y pueden asociarse con cualquier otra clase. Sin embargo, en la estructura jerárquica de un arquetipo, un término se define como componente (rango) de un atributo y las referencias internas pretenden reutilizar su definición en otros atributos. En OWL las clases son independientes y no dependen de ninguna propiedad. Por tanto, desde el punto de vista de la representación en OWL, la reutilización de un término a través de una referencia interna es equivalente a declarar el término, ya que en ambos casos el término se declara como una clase y se liga al atributo a través de un axioma OWL. La figura 6.10 incluye un fragmento de un arquetipo que declara una referencia interna (declaración `use_node`). El término arquetipado se indica con la ruta que lo identifica (`/items[at0001]/items[at0002]`). En la representación OWL se sustituye la ruta por el término arquetipado al que referencia. El primer axioma de la clase `CLUSTER_at0005` indica que es rango del atributo `items` y el segundo axioma expresa la restricción de ocurrencias.

Un slot no define un término arquetipado, sino que representa un punto de enlace con otro arquetipo. La definición de un slot incluye la clase del modelo de referencia con la que deben ser compatibles los arquetipos, la expresión regular que filtra los identificadores de los arquetipos y un intervalo de ocurrencias. La figura 6.11 contiene un ejemplo de un arquetipo que declara un slot de tipo `CLUSTER` y su representación en OWL. El primer axioma de la clase `CLUSTER_at0000` introduce el slot como rango del atributo. El rango será consistente si la clase que representa al slot (`CLUSTER_at0001`) es compatible con el tipo del atributo `items`. La restricción de ocurrencias del slot es introducida en el segundo axioma de la clase `CLUSTER_at0000`. Este axioma y el resto de restricciones de ocurrencias y cardinalidad se tendrán en cuenta en la validación de la regla de las ocurrencias. Por tanto, un slot se declara en una clase OWL que contiene los axiomas necesarios para validarlo como rango de un atributo y proporciona la información suficiente para validar la regla de las ocurrencias del atributo al que pertenece.

---

```

ADL
CLUSTER[at0000] matches {
  items ... {
    CLUSTER[at0005] ... matches {
      use_node ELEMENT occurrences matches {0..5}
      /items[at0001]/items[at0002]
    }
  }
}

```

---

```

OWL
Class: CLUSTER_at0005
SubClassOf:
  CLUSTER_items some ELEMENT_at0002
  CLUSTER_items max 5 ELEMENT_at0002

```

---

Figura 6.10: Representación OWL de una referencia interna

---

```

ADL
CLUSTER[at0000] matches {
  items cardinality matches {1..*; unordered} matches {
    allow_archetype CLUSTER[at0001] occurrences matches
    {0..3} matches {
      include archetype_id/value matches { ... }
    }
  }
}

```

---

```

OWL
Class: CLUSTER_at0000
SubClassOf:
  CLUSTER_items some CLUSTER_at0001
  CLUSTER_items max 3 CLUSTER_at0001
  ...

Class: CLUSTER_at0001
SubClassOf:
  CLUSTER,
  from_archetype only ...

```

---

Figura 6.11: Representación OWL de un slot

### 6.3. Detección de inconsistencias en especializaciones

El formalismo de arquetipos define la relación de especialización en términos de conformidad de extractos de HCE. Un arquetipo es especialización de otro si cualquier extracto del arquetipo es *conforme* con el arquetipo base que especializa. De esta definición cabe destacar que no implica herencia de declaraciones, tal como sucede en el paradigma de modelado orientado a objetos (UML). Por ejemplo, sea un atributo definido como opcional en el modelo de referencia y restringido en un arquetipo para que sea obligatorio, cualquier arquetipo que lo especialice debe volver a declarar este atributo como obligatorio, es decir, al menos con cardinalidad mínima 1, para ser conforme con el arquetipo base.

El modelo de arquetipos carece de una definición formal de la semántica de la especialización. Esta deficiencia ha motivado que la mayoría de editores de arquetipos no validen la relación de especialización, siendo el editor LinkEHR-Ed una excepción. Esta situación explica en parte el elevado número de especializaciones inconsistentes que han sido encontradas en el análisis de dos repositorios de arquetipos (véase sección 6.5). El método Archeck define la semántica de la especialización de arquetipos como la relación de inclusión de las clases OWL que definen a los términos de los arquetipos. Esta definición es el soporte para la detección de inconsistencias en las especializaciones, tal como se describirá en este apartado.

La relación de especialización entre arquetipos no distingue entre términos nuevos, términos redefinidos y términos repetidos. Se utiliza una convención para asignar identificadores a los términos que es interpretada por las herramientas de edición: un nuevo concepto tiene un identificador que comienza por "at0.", un término redefinido mantiene el identificador del concepto que redefine y añade ".1", mientras que un término repetido mantiene el mismo identificador. Por ejemplo, la figura 6.12 muestra el mapa conceptual del arquetipo *Examen del Feto* en el que se puede apreciar que repite la declaración del término *at0002* (Descripción clínica), redefine el término *at0003* (Hallazgos) utilizando el identificador *at0003.1* (Hallazgos abdominales) e introduce nuevos términos como *at0.1* (Posición del feto).

Un arquetipo no define un esquema para la representación y validación de datos, sino que debe entenderse como una guía para la captura y visualización de los datos de un extracto. Este papel que juegan los arquetipos en las arquitecturas de HCE basadas en el modelo dual se pone de manifiesto en la redefinición del rango de un atributo en la relación de especialización. Un arquetipo especializado puede extender el rango de un atributo, esto es, añadir nuevos términos. La figura 6.12 muestra un ejemplo de extensión del rango del concepto *at0003* (Hallazgos) del arquetipo *Examen* y la redefinición de ese concepto en el arquetipo *Examen del Feto* (*at0003.1*, Hallazgos Abdominales) que introduce un nuevo concepto (*at0.1*, Posición del feto).



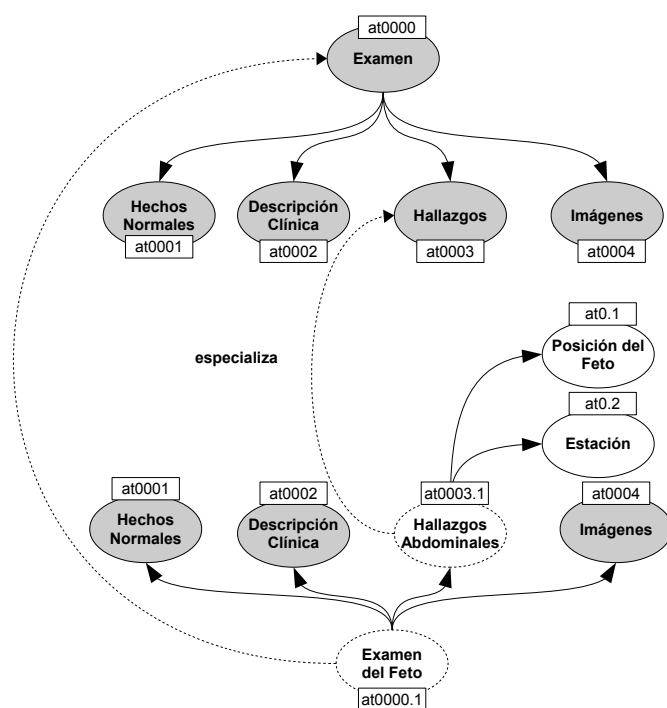


Figura 6.12: Asignación de identificadores al arquetipo *Examen del Feto*

Las introducción de nuevos términos en el rango de un atributo implica que los extractos conformes al arquetipo *Examen del Feto* tendrán más información que los extractos del arquetipo *Examen*. Sin embargo, un extracto del arquetipo *Examen del Feto* puede ser visto como un arquetipo *Examen*, ya que las definiciones que no forman parte del arquetipo base no son tenidas en cuenta en la visualización. En cambio, la especialización exige que si un componente es obligatorio en el arquetipo base, los extractos del arquetipo especializado deben tener esa información. Por tanto, la relación de especialización no debe entenderse como la conformidad entre dos esquemas de datos, sino como una relación de conformidad de restricciones.

La sección 5.4 trata la representación de arquetipos en OWL utilizada en el método Archeck. La propuesta está basada en la definición de los términos arquetipados como clases OWL utilizando un axioma de equivalencia que combina todas las restricciones que establece el término sobre sus atributos. La figura 5.1 corresponde con la clase OWL que representa el concepto raíz del arquetipo *Examen*. Los tres primeros axiomas se utilizan para: 1) clasificar a la clase como un término arquetipado (`ARCHETYPED_CLASS`), 2) indicar la clase del modelo de referencia que especializa el término (`CLUSTER`) y 3) identificar el concepto que está definiendo (concepto `at0000`). El resto de axiomas corresponden con las restricciones establecidas sobre los atributos de la clase `CLUSTER`.

```
Class: CLUSTER_at0000
EquivalentTo:
  ARCHETYPED_CLASS
  and CLUSTER
  and CONCEPT_at0000
  and CLUSTER_items only UNORDERED
  and CLUSTER_items only NON_UNIQUE
  and CLUSTER_items min 1 ITEM
  and CLUSTER_items max 1 CLUSTER_at0001
  and CLUSTER_items max 1 ELEMENT_at0002
  and CLUSTER_items max 1 CLUSTER_at0003
```

Figura 6.13: Representación OWL del concepto *Examen* para validación de la relación de especialización

La representación elegida para los términos arquetipados no discrimina a los arquetipos especializados, es decir, que la relación de especialización no afecta a la definición en OWL de los términos que contiene. Esto significa que las ontologías de los arquetipos son independientes entre sí. Esta decisión contrasta con la relación que se establece con el modelo de referencia que supone que todos los términos arquetipados son subclases de la clase del modelo de referencia que restringen. Esta decisión está basada en la distinción entre corrección e inconsistencia. La representación para la validación respecto al modelo de referencia trata de identificar los términos arquetipados incorrectos, los cuales son clasificados como clases inconsistentes en OWL. Por el contrario, un arquetipo puede ser correcto desde el punto del vista del modelo de arquetipos, aunque no sea una especialización consistente.

En esta sección se describe el método de validación de la relación de especialización apoyado en la representación OWL presentada en la sección 5.4.2. La representación será extendida con el fin de que los errores de modelado puedan ser identificados de forma precisa. Además, se abordará el tratamiento de la propagación de errores, debido a la estructura jerárquica de los arquetipos, y el tratamiento de los conceptos opcionales. Finalmente, se enumeran los tipos de inconsistencias que son identificados por el método Archeck.

### 6.3.1. Método de validación

La relación de conformidad que establece la especialización de arquetipos sólo afecta a los términos que ambos arquetipos tienen en común, quedando fuera de la validación los nuevos términos introducidos por el arquetipo especializado. Así pues, la relación de especialización debe ser validada desde el arquetipo base, es decir, desde el arquetipo que es especializado. En otras palabras, la regla de conformidad se cumple si todos los términos del arquetipo base subsumen algún término del arquetipo especializado.

El método Archeck valida una relación de especialización a partir de las inferencias de axiomas de inclusión entre los términos del arquetipo especializado y los términos del arquetipo base. A pesar de que entre los términos de ambos arquetipos no se establece ninguna relación lógica, la representación OWL propuesta garantiza, gracias al uso de las clases *CONCEPT*, que la relación de inclusión entre dos términos podrá ser inferida y que un término del arquetipo base podrá subsumir como mucho a un término del arquetipo especializado.

Un término de un arquetipo base se ha especializado correctamente si:

1. El término subsume a otro término del arquetipo especializado. Si la relación de inclusión también existe en sentido contrario, entonces se entiende que los términos son equivalentes, y por tanto, se ha mantenido su definición.
2. El término no subsume a ningún otro término y la clase que representa el concepto sólo subsume a este término. Dado que los términos arquetipados son subclases de las clases que representan a los conceptos, y éstas son disjuntas, un término no puede subsumir a ningún otro de su propia ontología. En este caso se interpreta que el término define un *concepto opcional* que no ha sido declarado en el arquetipo hijo.

El proceso de validación de la consistencia de la relación de especialización entre dos arquetipos se desarrolla en las siguientes fases:

- **Preparación.** Se crea una base de conocimiento formada por el modelo de referencia y las ontologías del arquetipo base y la especialización.
- **Detección de conceptos inconsistentes.** Se obtienen las inferencias de la base de conocimiento y se procesan los términos del arquetipo base en busca de relaciones de subsunción con los términos del arquetipo hijo, es decir, se consulta si el término ha sido correctamente especializado.

El resultado del proceso es el conjunto de conceptos inconsistentes en la relación de especialización. Por tanto, un arquetipo especializa correctamente a otro si en el proceso de validación no se obtiene ningún concepto inconsistente.

### 6.3.2. Detección precisa de inconsistencias

La representación propuesta en OWL para los términos arquetipados es eficaz para detectar si una especialización es consistente, pero presenta dos inconvenientes que dificultan la interpretación de los errores de modelado. En primer lugar, el método informa sobre los conceptos inconsistentes, pero no indica las restricciones que han causado los errores. Por ejemplo, el término que presenta la figura 6.13 podría ser inconsistente por una o varias de las restricciones que declara. En segundo lugar, la representación de la restricción de ocurrencia basada en el axioma de cardinalidad de OWL puede provocar que un término inconsistente haga también inconsistente al concepto que lo contiene. La figura 6.14 presenta un ejemplo de restricción de ocurrencia mínima del término `CLUSTER_at0001` dentro del atributo `items` del término `CLUSTER_at0000` en dos arquetipos relacionados por especialización (el prefijo “padre” identifica al arquetipo base e “hijo” al arquetipo especializado). Si el concepto `at0001` no está correctamente especializado, el axioma de cardinalidad mínima de OWL hace que el concepto `at0000` tampoco lo esté. El motivo es que la clase `padre:CLUSTER_at0000` no subsume a la clase `hijo:CLUSTER_at0000`, es decir, la inconsistencia del concepto `at0001` se propaga al concepto `at0000`.

```
Class: padre:CLUSTER_at0000
  EquivalentTo:
    ...
    and CONCEPT_at0000,
    and CLUSTER_items min 1 padre:CLUSTER_at0001

Class: hijo:CLUSTER_at0000
  EquivalentTo:
    ...
    and CONCEPT_at0000,
    and CLUSTER_items min 1 hijo:CLUSTER_at0001
```

Figura 6.14: Ejemplo de propagación de inconsistencias

En este apartado se actualiza la representación de arquetipos en OWL presentada en la sección 5.4.2 para la detección precisa de errores y para evitar la propagación de inconsistencias entre conceptos. Esta actualización debe ser entendida como una mejora de cara a la usabilidad del método cuando el resultado de la validación debe ser interpretada por usuarios.

### 6.3.2.1. Restricciones atómicas

La identificación precisa de las restricciones que causan los errores de modelado requiere extender la representación OWL propuesta en el apartado anterior. La extensión propuesta consiste en añadir una clase OWL por cada restricción del arquetipo (clases *restricción*). Estas clases se relacionan con el concepto que definen a través de la propiedad `constraint`. A su vez, los tipos de restricciones son organizados en categorías disjuntas (orden, unicidad, etc.) de modo que cada clase *restricción* se define como una especialización de alguna de estas categorías. Por último, la representación de las clases que declaran las restricciones también está basada en axiomas de equivalencia, ya que el método de validación buscará inferencias que las relacionen por inclusión.

La figura 6.15 muestra la restricción de orden del atributo `items` del concepto raíz del arquetipo *Examen*. El primer axioma indica que es una restricción de orden (`ORDER`) y el segundo relaciona la restricción con en el concepto que la define (`at0000`). El último axioma corresponde con la representación en OWL de la restricción del orden, que es igual que el axioma incluido en el término arquetipado (véase figura 6.13).

```
Class: ORDER_CLUSTER_at0000_items
  EquivalentTo:
    ORDER
    and constraint only CONCEPT_at0000
    and CLUSTER_items only UNORDERED
```

Figura 6.15: Restricción de orden del concepto raíz arquetipo *Examen* en OWL

Las clases *restricción* se asocian a los conceptos (`CONCEPT_at0000`) en lugar de estar vinculadas a los términos arquetipados (`CLUSTER_at0000`). Esto es así debido a que estas clases son de utilidad para analizar las declaraciones que causan la inconsistencia en un término arquetipado. Si se ligara la clase restricción con el término arquetipado (inconsistente), ya no sería posible comprobar si la restricción del arquetipo especializado es con-

forme con el arquetipo base. Por tanto, utilizando la clase que representa al concepto, que es única y común al arquetipo base y especializado, es posible inferir la relación de inclusión entre las clases.

El resto de restricciones se representan de forma análoga. Para facilitar la interpretación de los errores de modelado se introducen anotaciones en la declaración de las clases que no tienen ningún significado lógico. La figura 6.16 define la restricción de ocurrencia del término *at0001* del arquetipo *Examen*. Las anotaciones incluyen información que ayuda a interpretar el error de modelado.

```
Class:
    OCCURRENCE_CLUSTER_at0000_items_CLUSTER_at0001
Annotations:
    concept 'at0000',
    attribute 'items',
    range 'at0001'
EquivalentTo:
    OCCURRENCE
    and constraint only CONCEPT_at0000
    and CLUSTER_items max 1 CLUSTER_at0001
```

Figura 6.16: Restricción de ocurrencias del término *at0001* del arquetipo *Examen*

### 6.3.2.2. Propagación de errores

La figura 6.14 muestra un ejemplo de propagación de errores en el que un concepto que no se ha especializado correctamente hace que el término que lo contiene también sea erróneo. Las clases *restricción* presentadas en el apartado anterior también se enfrentan a un problema similar que se ha resuelto desvinculando las restricciones de los términos arquetipados, es decir, las clases *restricción* se relacionan con la clase que representa al concepto. La misma estrategia puede ser aplicada para evitar la propagación de errores. La figura 6.17 adapta el ejemplo presentado en la figura 6.14 para utilizar clases *concepto* en lugar de términos arquetipados. De este modo, la definición incorrecta del concepto *at0001* en el arquetipo hijo ya no afecta a la consistencia del concepto *at0000*.

```
Class: padre:CLUSTER_at0000
  EquivalentTo:
    ...
    and CONCEPT_at0000,
    and CLUSTER_items min 1 CONCEPT_at0001

Class: hijo:CLUSTER_at0000
  EquivalentTo:
    ...
    and CONCEPT_at0000,
    and CLUSTER_items min 1 CONCEPT_at0001
```

Figura 6.17: Uso de las clases *concepto* para evitar la propagación de inconsistencias

En definitiva, en la representación OWL utilizada en el método Archeck los términos arquetipados no se relacionan entre sí, y por tanto, no propagan los errores de modelado. Las referencias internas y los slots son tratados igual que el resto de términos arquetipados, ya que en ambos casos también están relacionadas con la definición de un concepto.

### 6.3.2.3. Proceso de validación con detección precisa de errores

El proceso de validación de la relación de especialización introduce una etapa final para la detección precisa de inconsistencias que se apoya en el uso de las clases *restricción*. Los conceptos inconsistentes identificados en la etapa anterior son analizados para la identificación de los errores de modelado.

El análisis de los conceptos inconsistentes se realiza desde el arquetipo base en los siguientes pasos:

- Obtención de las restricciones del concepto inconsistente en el arquetipo base. Dado que las clases *restricción* son subclases del concepto que definen, éstas pueden ser identificadas en la base de conocimiento.
- Análisis de la restricción. Una restricción es correcta si subsume alguna restricción del arquetipo especializado. Tal como ocurre con los términos arquetipados, una restricción puede subsumir como mucho a otra restricción.
- Interpretación del error de modelado. Una restricción incorrecta significa la presencia de un error de modelado que es interpretado según

el tipo de la restricción (orden, cardinalidad, etc.) y las anotaciones que incluye la clase que declara la restricción.

La figura 6.18 muestra las dos fases del proceso de validación aplicadas a los arquetipos *Examen* y *Examen del Feto*. Supongamos que tras la aplicación de la primera fase se marca como inconsistente el concepto “hallazgos abdominales”, pues el razonador no infiere que sea subsumido en la declaración del concepto “hallazgo” definido en el arquetipo base. En la segunda fase se detecta que la restricción de orden no es consistente, debido a que los conceptos establecen restricciones de orden contradictorias para el atributo *items* (ordenado y no ordenado). Por tanto, el proceso de validación identifica al concepto “hallazgos abdominales” como inconsistente y aporta como explicación la contradicción en la definición de orden del atributo *items*.

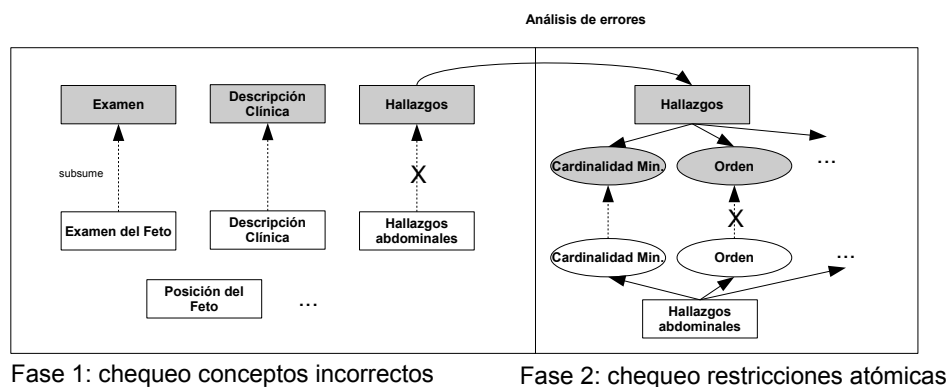


Figura 6.18: Fases del proceso de validación de la relación de especialización

### 6.3.3. Control de conceptos opcionales

Un concepto es opcional si el número de ocurrencias que define en los atributos a los que pertenece permite no declararlo en un arquetipo especializado. El valor 0 como cota mínima del intervalo de ocurrencias significa que el concepto es opcional, y por tanto, admite que el concepto pueda no ser declarado en una especialización. Sin embargo, la representación OWL de los conceptos opcionales plantea un problema debido a que el modelo de arquetipos no exige en estos casos que se redefina el término para establecer un valor 0 como cota máxima del número de ocurrencias. En otras palabras, es suficiente con no declarar un concepto opcional en un arquetipo especializado.

El problema de los conceptos opcionales se ilustra en la figura 6.19 que muestra el término *at0000* que es redefinido en una relación de especialización (ontologías *padre* e *hijo*). Por un lado, el término del arquetipo *padre*



declara 3 términos como rango del atributo `items: at0001` y `at0002`, con ocurrencias entre 0 y 1, y `at0003` con al menos una ocurrencia. Los conceptos `at0001` y `at0002` son opcionales, ya que el intervalo de ocurrencias admite el valor 0. Por otro lado, el término en el arquetipo *hijo* no declara el concepto opcional `at0002`. Con ambos términos el método *Archeck* detectaría que el concepto `at0000` es una especialización inconsistente debido a que no se ha inferido que la clase `padre:CLUSTER_at0000` subsuma a la clase `hijo:CLUSTER_at0000`. La explicación es que el modelo de razonamiento en mundo abierto de OWL no interpreta la ausencia de un axioma de cardinalidad para el concepto `at0002` como que el concepto no pueda existir.

```
Class: padre:CLUSTER_at0000
  EquivalentTo:
    CLUSTER
    and CONCEPT_at0000,
    and CLUSTER_items max 1 CONCEPT_at0001
    and CLUSTER_items max 1 CONCEPT_at0002
    and CLUSTER_items min 1 CONCEPT_at0003

Class: hijo:CLUSTER_at0000
  EquivalentTo:
    CLUSTER
    and CONCEPT_at0000
    and CLUSTER_items max 1 CONCEPT_at0001
    and CLUSTER_items min 1 CONCEPT_at0003
```

Figura 6.19: Ejemplo de inconsistencia provocada por conceptos opcionales

La solución a este problema consiste en introducir una restricción de cardinalidad cualificada en el arquetipo *hijo* que establezca la cota superior de la cardinalidad del concepto `at0002` a 0, es decir, evita que el concepto aparezca como rango del atributo. No obstante, ésta sería una solución particular para este caso, pero no es una solución genérica que permita obtener la representación OWL de un arquetipo especializado sin tener que procesar el arquetipo base.

La generalización de la solución al problema de los conceptos opcionales pasa por extender de nuevo la representación OWL. En esta ocasión se añaden dos clases OWL por cada atributo redefinido en el arquetipo, una de ellas para representar los conceptos opcionales del rango y otra para los conceptos obligatorios. La figura 6.20 muestra la declaración de estas clases para el atributo `items` representado en la figura 6.19. Se aplica una convención de nombrado para identificar el conjunto de conceptos opcionales

para estas clases que utiliza el prefijo “OPTIONAL” y “MANDATORY” para los conceptos obligatorios, seguido del nombre de la clase OWL que contiene la restricción (“CLUSTER\_at0000”) y por último el atributo restringido (“items”).

```

Class: padre:OPTIONAL_CLUSTER_at0000_items
  EquivalentTo: CONCEPT_at0001 or CONCEPT_at0002

Class: padre:MANDATORY_CLUSTER_at0000_items
  EquivalentTo: CONCEPT_at0003

Class: hijo:OPTIONAL_CLUSTER_at0000_items
  EquivalentTo: CONCEPT_at0001

Class: padre:MANDATORY_CLUSTER_at0000_items
  EquivalentTo: CONCEPT_at0003

```

Figura 6.20: Clases de apoyo para el tratamiento de conceptos opcionales

Las clases que representan los componentes *opcionales* y *obligatorios* definen dos conjuntos disjuntos cuya unión es el conjunto de conceptos que definen el rango de un atributo. Estas clases permiten determinar los *conceptos opcionales no declarados* en un arquetipo especializado como los conceptos opcionales en el arquetipo base que no se han declarado en el arquetipo especializado (no son opcionales ni obligatorios). En el ejemplo de la figura 6.20 se calcularía como:

```

Opcional(padre)={at0001, at0002}
Obligatorio(padre)={at0003}
Opcional(hijo)={at0001}
Obligatorio(hijo)={at0003}

NoDeclarados(hijo)={at0002}

```

Por tanto, los atributos multivaluados deben declarar un axioma que establece la cardinalidad máxima a 0 para los conceptos opcionales en el arquetipo base y no declarados en la especialización (véase figura 6.21).

```
Class: hijo:UNDEFINED_CLUSTER_at0000_items
EquivalentTo:
  ((padre:MANDATORY_CLUSTER_at0000_items and
    not (padre:OPTIONAL_CLUSTER_at0000_items and
      hijo:OPTIONAL_CLUSTER_at0000_items))
    and not hijo:MANDATORY_CLUSTER_at0000_items)

Class: hijo:CLUSTER_at0000
EquivalentTo:
  CLUSTER
  and CONCEPT_at0000
  and CLUSTER_items max 1 CONCEPT_at0001
  and CLUSTER_items min 1 CONCEPT_at0003
  and CLUSTER_items max 0
    hijo:UNDEFINED_CLUSTER_at0000_items
```

Figura 6.21: Solución al problema de los conceptos opcionales

#### 6.3.4. Tipos de inconsistencias en especializaciones

En este apartado presenta un resumen de los errores de modelado en la especialización de arquetipos identificados por el método Archeck. Estos errores se organizan según los tipos de restricciones del modelo de arquetipos:

Cardinalidad de los atributos. Este error es detectado cuando se declaran intervalos de cardinalidad incompatibles. Por ejemplo, si el arquetipo *Examen* hubiera establecido la cardinalidad del atributo *items* del concepto raíz al rango 1..\*, provocaría que la definición del arquetipo hijo fuera inconsistente debido a que establece la cardinalidad en 0..\*.

Ocurrencias de un concepto. La violación del intervalo de ocurrencias es motivo de inconsistencia. Un ejemplo de este tipo de error sería permitir más de una ocurrencia del concepto “Hallazgos abdominales” en el arquetipo *Examen del feto* debido a que en la definición del arquetipo base se indica que el concepto *at0003* puede tener como mucho una instancia.

Orden de los conceptos en un atributo. Este tipo de error corresponde con el ejemplo desarrollado en la figura 6.18. El orden de un atributo no debe ser entendido como una restricción, ya que no admite posibilidad de cambio en otra redefinición. Por tanto, la conformidad de la restricción

de orden sucede cuando los valores establecidos en el arquetipo base y en la especialización son el mismo.

Unicidad de las instancias en un atributo. Esta categoría de error tiene una interpretación similar a la restricción de orden, es decir, la conformidad de una especialización consiste en asignar el mismo valor.

Rango de un atributo. El error de rango es común en atributos simples en los que se sustituye el tipo declarado modelo de referencia, como por ejemplo el atributo `value` de la clase `ELEMENT` de `openEHR`. El análisis de los repositorios de arquetipos que se presenta en la sección 6.1 señala que este tipo de error es el más común.

Relación de slot. Este error sucede cuando un slot de un arquetipo especializado permite más arquetipos que la definición del arquetipo base. La detección de este error se basa en comprobar si el patrón de expresión regular que define el arquetipo base subsume el patrón de la especialización.

## 6.4. Métricas de calidad

La consistencia a nivel técnico de los arquetipos es un requisito para que sean usados en los sistemas de información sanitarios. La representación en OWL de los arquetipos permite la validación técnica basada en el análisis de las inferencias obtenidas de los razonadores, es decir, que la corrección de la representación es el soporte del método de validación. Siguiendo este principio, en esta sección se extiende la representación OWL de los arquetipos para la definición e implementación de métricas de calidad en el modelado de arquetipos. La propuesta de definición de métricas se motiva con la evaluación de enlaces terminológicos, aunque podrían ser aplicadas a cualquier otra características de los arquetipos.

### 6.4.1. Consistencia de enlaces terminológicos

En este apartado se desarrolla un método para evaluar la calidad de los enlaces terminológicos en las relaciones de especialización basado en la representación de arquetipos en OWL del método `Archeck`. Esta métrica no forma parte de la semántica del modelo de arquetipos y se basa en el supuesto de que un enlace terminológico es correcto si es compatible con el enlace establecido en el padre. La métrica ha sido implementada para la terminología `SNOMED-CT` y ha sido añadida al método de validación de `Archeck`.

El problema de la consistencia de los enlaces terminológicos en las especializaciones se motiva con un ejemplo tomado del repositorio de arquetipos

CKM (abril de 2013). El concepto raíz del arquetipo “Heart rate” está enlazado con el concepto SNOMED-CT 364075005 (*Heart rate*). Este arquetipo está especializado por el arquetipo “Heart rate pulse” cuyo concepto raíz está enlazado al concepto de SNOMED-CT 248627000 (*Pulse characteristics*). Los enlaces terminológicos establecidos por ambos arquetipos hacen que la especialización sea inconsistente. El motivo de la inconsistencia es que las relaciones de los dos conceptos de SNOMED-CT son contradictorias con la relación de los arquetipos, ya que el concepto 248627000 es un ancestro de 364075005 (véase figura 6.22).

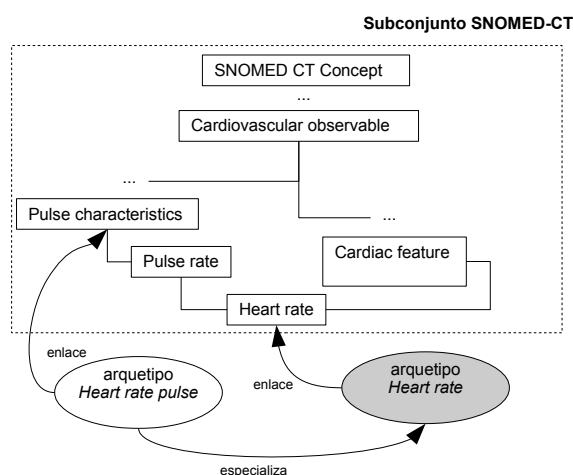


Figura 6.22: Enlace terminológico de los arquetipos *Heart Rate* y *Heart Rate Pulse* con la terminología SNOMED-CT

La sección *ontología* de los arquetipos permite añadir enlaces terminológicos a los identificadores de los términos con conceptos de terminologías clínicas tales como SNOMED-CT o LOINC. Dada la representación OWL presentada en la sección 5.4, los enlaces terminológicos pueden ser incluidos en la definición de la clase OWL que define los términos arquetipados. La representación de los enlaces terminológicos requiere que la terminología, o al menos una parte de ella, esté disponible en OWL. La figura 6.23 muestra el concepto raíz del arquetipo *Examen* que es enlazado con el concepto 425044008 de SNOMED-CT (*Sección examen físico*). La única diferencia con la definición presentada en la figura 6.1 es la introducción del concepto de la terminología como parte del axioma de equivalencia. Por tanto, los enlaces terminológicos son procesados como cualquier otra restricción del arquetipo, es decir, un enlace terminológico incorrecto (no consistente) haría que el razonador no infiera que el término del arquetipo base subsume la definición del término en el arquetipo especializado, y el error preciso sería detectado en la segunda fase del proceso de validación.

```
Class: CLUSTER_at0000
Annotations:
  rdfs:label 'Examination'
EquivalentTo:
  ARCHETYPED_CLASS
  and CLUSTER
  and CONCEPT_at0000
  and snomed-ct:425044008
  ...
```

Figura 6.23: Representación OWL del enlace terminológico del concepto *Examen* en SNOMED-CT

El proceso de validación de Archeck se ve seriamente penalizado por la introducción de la ontología de SNOMED-CT en la base de conocimiento. La mayoría de los razonadores OWL-DL tienen problemas para tratar con grandes terminologías como SNOMED-CT que tiene más de 300.000 conceptos. Por tanto, para mejorar el rendimiento del proceso, la base de conocimiento incluye sólo el subconjunto mínimo de conceptos de SNOMED-CT que son necesarios para la detección de inconsistencias. Este subconjunto está formado por: 1) los conceptos de SNOMED-CT enlazados a los términos del arquetipo que está siendo validado, 2) los conceptos SNOMED-CT enlazados al arquetipo base y 3) los conceptos que son ancestros de los conceptos añadidos en 1) y 2). El uso de este subconjunto es suficiente para soportar la búsqueda de relaciones de subsunción entre los términos arquetipados. La figura 6.22 muestra el subconjunto de conceptos de SNOMED-CT seleccionados para la validación del arquetipo “Pulso cardíaco”, que especializa al arquetipo “Ritmo cardíaco”.

#### 6.4.2. Calidad de enlaces terminológicos

En el apartado anterior se ha presentado una técnica para validar la consistencia de enlaces terminológicos en relaciones de especialización. La solución propuesta aprovecha la integración del modelo de arquetipos y las terminologías a través de su representación común en OWL y los servicios de razonamiento ofrecidos por el formalismo de la lógica descriptiva. En este apartado se desarrolla esta estrategia como soporte para la definición e implementación de métricas de calidad de arquetipos basadas en enlaces terminológicos. Estas métricas utilizan la representación OWL de arquetipos como instancias propuesta en la metodología Encorsetable (véase sección 5.5).

La implementación de las métricas de calidad para enlaces terminológicos se fundamenta en la representación de arquetipos como instancias OWL. El enlace con los conceptos de una terminología puede ser representado como una relación entre instancias o como una clasificación de instancias. La primera opción utiliza una propiedad para relacionar los términos del arquetipo con los conceptos de la terminología, ambos instancias de las ontologías correspondientes. La segunda alternativa se basa en clasificar una instancia como perteneciente a la clase que representa el concepto de la terminología. Ambas alternativas son igual de eficaces para enlazar términos de un arquetipo con una terminología. Sin embargo, la relación basada en propiedades permite aprovechar las posibilidades de razonamiento sobre propiedades: transitividad, inversa, asimetría, cadenas de propiedades, etc. Por tanto, se propone que las terminologías que van a ser enlazadas tengan una representación como instancias.

#### 6.4.2.1. Integración de terminologías

La representación de arquetipos como instancias OWL en la metodología Encorsetable define la propiedad `term_binding` para conectar los términos de un arquetipo con los conceptos de una terminología. Es requisito de la representación que la terminología también esté definida como instancias en OWL. Para la integración de las terminologías en la representación propuesta se ha definido la clase `TERM_CONCEPT` como rango de la propiedad `term_binding`. Los conceptos de una terminología deben ser instancias de esa clase para poder ser enlazados con los términos del arquetipo.

La definición de las métricas de calidad aprovecha las relaciones semánticas presentes en las terminologías. Se han definido dos propiedades que representan las relaciones taxonómicas y lógicas entre los conceptos de una terminología. La propiedad `is_a` declara la relación de inclusión entre dos conceptos. Esta propiedad se ha definido como *transitiva* para que sea semánticamente equivalente el axioma *SubClassOf* de OWL. Nótese que no empleamos el axioma *SubClassOf* para construir la taxonomía de conceptos debido a que estamos representando arquetipos y terminologías como instancias. Las relaciones lógicas entre los conceptos, como por ejemplo *parte de*, se establecen utilizando la propiedad `attribute`, que puede ser especializada para representar la semántica precisa de la relación. Finalmente, tanto `is_a` como `attribute` se declaran subpropiedades de `semantic_relation` que representa las relaciones semánticas entre dos conceptos.

Seguidamente se ilustra el proceso de integración de una terminología utilizando el caso de SNOMED-CT, ya que esta terminología es suficientemente expresiva para la definición de métricas de calidad. La integración de una terminología no exige que las instancias de los conceptos se relacionen

explícitamente con los conceptos y propiedades introducidos anteriormente. Por ejemplo, podemos utilizar las relaciones lógicas definidas en la terminología (*parte de, causa de, etc.*) y posteriormente integrar esas relaciones con la representación Encorsetable.

La terminología SNOMED-CT se ha definido utilizando los siguientes constructores:

- Clase `SNOMED_CONCEPT`. Esta clase representa a los conceptos de la terminología. Todos los conceptos, salvo la jerarquía *Linkage concept*, son instancias de esta clase.
- Propiedad `linkage_concept`. Esta propiedad representa la raíz de las relaciones entre conceptos en la terminología. Todos los conceptos de la jerarquía *Linkage concept* se emplean para relacionar conceptos de la terminología. Así pues, la relación taxonómica de los conceptos de esta jerarquía corresponde con una jerarquía de propiedades cuya raíz es `linkage_concept`.

La integración de la terminología SNOMED-CT se consigue definiendo axiomas que relacionan los dos constructores anteriores con la representación Encorsetable. En primer lugar, se declara la clase `SNOMED_CONCEPT` como subclase de `TERM_CONCEPT`. En segundo lugar, la propiedad que define las relaciones en SNOMED-CT (`linkage_concept`) se declara como subpropiedad de `attribute`. Por último, el concepto 116680003 (*is a*) es un relación lógica al ser descendiente de *Linkage concept* y además se define como subpropiedad de la propiedad `is_a` de la Ontología de Restricciones para indicar que es utilizado para declarar la relación taxonómica de los conceptos en SNOMED-CT.

#### 6.4.2.2. Extensión de la Ontología de Restricciones

La Ontología de Restricciones es la ontología superior de la metodología Encorsetable. Las clases y propiedades que se presentaron en la sección 5.5 son suficientes para representar semánticamente el modelo de arquetipos. Sin embargo, la implementación de las métricas se basa en consultas OWL en Lógica Descriptiva cuya definición se simplifica incluyendo algunas propiedades inversas y cadenas de propiedades en la ontología. Estas propiedades son:

- `type_of` es utilizada por asociar los términos del arquetipo con el rango que declara sus ocurrencias e indirectamente con las propiedades que hacen uso de ellos como rangos. Esta propiedad es inversa de `has_type`.



- `binding_from` es la propiedad inversa de `term_binding` y permite identificar los términos arquetipados con los que está asociado un concepto de una terminología.
- `component_of` relaciona un concepto arquetipado con las propiedades donde es declarado como rango. Se define como la cadena de propiedades `type_of` o `range_of`. También se declara la propiedad inversa `component`.
- `part_of` establece la relación mereológica entre los términos de un arquetipo. Dado que la estructura de un arquetipo es jerárquica, la propiedad se define como `component_of` o `property_of`.
- `sibling` relaciona dos conceptos que se definen como rango de una misma propiedad. Esta definida como `component_of` o `component`. Esta propiedad se define como *simétrica*.

#### 6.4.2.3. Definición de métricas

Una métrica define un criterio cuantificable que mide la calidad de un arquetipo. Dado que los arquetipos se expresan utilizando los constructores de la Ontología de Restricciones, podemos utilizar estos mismos constructores para la especificación de las métricas. La definición de las métricas en OWL nos permite aprovechar las posibilidades de razonamiento de la lógica descriptiva, de manera que la definición pueda ser utilizada para obtener el valor asociado a la métrica.

La definición de una métrica consiste en la declaración de clases OWL que representan las colecciones de individuos que queremos cuantificar. Por ejemplo, la clase `term_binding some TERM_CONCEPT` equivale a todos los términos de un arquetipo que tienen algún enlace terminológico. El dato cuantificable que interesa de estas clases es el número de individuos que pertenecen a ella. Así pues, una métrica se expresa como una fórmula aritmética que combina el número de individuos de estas clases.

Las clases sobre las que se apoya el cálculo de las métricas también pueden expresarse como consultas parametrizadas. Por ejemplo, los términos que agrupa jerárquicamente otro término pueden consultarse utilizando la propiedad `part_of`. Sea `$t` el símbolo que identifica el parámetro, la consulta se podría expresar como `part_of value $t`.

Algunas métricas pueden requerir introducir nuevas propiedades a la ontología que mejoren la expresividad de la métrica. En general, estas propiedades se declaran como cadenas de propiedades y permiten inferir relaciones entre las instancias. Por ejemplo, podemos declarar la propiedad `local_binding` que conecte dos términos del arquetipo cuyos enlaces terminológicos tengan una relación lógica en la terminología. El axioma

que definiría esta propiedad sería: `term_binding` o `semantic_relation` o `binding_from`.

A continuación se define una métrica que mide el grado de *cohesión terminológica* entre un término y los términos que lo definen a través de sus atributos (componentes). Esta métrica es de interés, por ejemplo, para medir la conexión entre el enlace terminológico de un término que declara un elemento de tipo *texto codificado* y los enlaces de los códigos permitidos para el elemento. La definición de esta métrica se apoya en las siguientes clases:

- C1 - Componentes de un término *t* con enlaces terminológicos :  
`part_of value $t and term_binding some TERM_CONCEPT`
- C2 - Términos con conexión terminológica a *t*. Esta clase utiliza la propiedad `local_binding`, descrita anteriormente, que conecta dos términos con enlaces terminológicos relacionados en la terminología.  
`part_of value $t and local_binding value $t`

La métrica se expresa con la fórmula:  $M1 = \frac{C2}{C1}$

## 6.5. Validación de los repositorios

El método de validación se ha aplicado a los dos repositorios públicos de arquetipos openEHR más importantes en la actualidad, el repositorio gestionado en la herramienta Clinical Knowledge Manager (CKM) de la Fundación openEHR y el repositorio de programa National Health Service (NHS) del Reino Unido. La evaluación de los repositorios se ha centrado en los arquetipos que definen relaciones de especialización, 81 en CKM y 212 en NHS. El repositorio de NEHTA no ha sido incluido en el estudio debido al reducido número de arquetipos especializados que contiene.

La tabla 6.1 presenta un resumen de los resultados de la evaluación. Un informe preciso con los errores encontrados en los repositorios se encuentra en <http://miuras.inf.um.es/archeck>. En el análisis se ha incluido también la métrica de calidad sobre la consistencia de enlaces terminológicos en las relaciones de especialización que ha sido descrita en el apartado 6.4.1. Todos los arquetipos analizados declaran las restricciones correctas respecto al modelo de referencia (openEHR). Éste ha sido un resultado esperado, ya que los editores de arquetipos comprueban este tipo de error. Por otro lado, el análisis de los arquetipos especializados indica que un elevado número de ellos no definen una relación de especialización consistente. En concreto, la mayor parte de los errores son debidos a la violación de la restricción de rango del atributo `value` del tipo `ELEMENT`.

Para facilitar el análisis de los arquetipos, los errores de rango en la clase `ELEMENT` han sido clasificados en varias categorías:

**Reemplazo de texto codificado.** El arquetipo base declara el rango de tipo `CODED_TEXT` y en la redefinición se sustituye el tipo por `DV_TEXT`. Esta sustitución es incorrecta porque `CODED_TEXT` no es un tipo compatible con `DV_TEXT`.

**Códigos incompatibles.** Este error se produce en una especialización que redefine un elemento de tipo `CODED_TEXT` estableciendo un conjunto de códigos que no está incluido en el conjunto de códigos que define el arquetipo base. Por ejemplo, el arquetipo *Examen del feto* define el concepto "Posición del feto" como un texto codificado que permite el conjunto de códigos *Longitudinal, Transversal y Oblícuo*. Una especialización del arquetipo no puede extender este conjunto de códigos ni tampoco eliminar la restricción para permitir cualquier código.

**Cantidades.** Esta categoría representa la aplicación de restricciones incompatibles a los atributos del tipo cantidad (`DV_QUANTITY` en openEHR).

**Sustitución de tipo incorrecta.** Este grupo representa el resto de errores que corresponde con otros reemplazos de tipo incorrectos, tales como sustituir el tipo de datos `DV_TEXT` por `DV_COUNT`.

Se ha identificado otra categoría de error que está asociada con errores de redefinición de atributos, pero que es debida a errores en la estructura del arquetipo. Esta categoría se denomina *concepto desubicado*. En el ejemplo del arquetipo *Examen* se declara el concepto "Detalle" (at0006) bajo el encabezamiento "Hallazgos" (at0003). En el arquetipo especializado *Examen del feto* el concepto "Detalle" estaría *desubicado* si fuera declarado, por ejemplo, bajo el encabezamiento "Descripción clínica" (at0002).

La validación de los repositorios ha encontrado que un 22,2% de los arquetipos especializados en el repositorio CKM y un 21,2% en NHS contienen errores. El análisis de los resultados señala que el error más común es la definición incorrecta del tipo de datos de un `ELEMENT`, especialmente en el repositorio NHS. Además, los errores relacionados con la codificación (reemplazo de texto codificado, códigos incompatibles y enlace terminológico con SNOMED-CT) representan en torno al 49,2% de todos los errores. Los errores de codificación son más frecuentes en el repositorio de NHS (61,5%) que en el repositorio CKM (15,1%). Por otro lado, los errores debidos a la definición de la estructura del arquetipo (*conceptos desubicados*) son bastante más frecuentes en el repositorio de CKM. Por último, el único tipo de error que no se ha encontrado en ningún repositorio ha sido la restricción de unicidad. El análisis de los arquetipos revela que en muy pocas ocasiones esta restricción se establece explícitamente al definir la cardinalidad de un arquetipo, ya que toma el valor por defecto *no ordenado*.

	<b>CKM</b>	<b>NHS</b>	<b>Global</b>
<b>Atributo</b>	21,2 %	15,4 %	16,9 %
Cardinalidad	0,0 %	1,1 %	0,8 %
Ocurrencia	15,2 %	9,9 %	11,3 %
Orden	6,1 %	4,4 %	4,8 %
Unicidad	0,0 %	0,0 %	0,0 %
<b>Tipo de datos</b>	42,4 %	79,1 %	69,4 %
Reemplazo texto codificado	9,1 %	28,6 %	23,4 %
Códigos incompatibles	3,0 %	33,0 %	25,0 %
Cantidad	27,3 %	5,5 %	11,3 %
Sustitución de tipo incorrecta	3,0 %	12,1 %	9,7 %
<b>Enlace terminológico</b>	3,0 %	0,0 %	0,8 %
SNOMED-CT	3,0 %	0,0 %	0,8 %
<b>Otros</b>	33,3 %	5,5 %	12,9 %
Slot	15,2 %	0,0 %	4,0 %
Conceptos desubicados	18,2 %	5,5 %	8,9 %

Tabla 6.1: Distribución de los errores de modelado en los repositorios CKM y NHS

## 6.6. Herramienta de validación

El método de validación ha sido implementado en la herramienta web *Archeck* que está disponible en <http://miuras.inf.um.es/archeck>. Los errores de consistencia son identificados de forma precisa de acuerdo al identificador del término y el atributo donde se ha producido el error. La herramienta ha sido desarrollada en Java y utiliza el analizador de arquetipos de openEHR [29]. Las ontologías son procesadas con OWL API [78] y se utiliza Hermit [175] como razonador, ya que es el único razonador disponible actualmente que soporta la definición de axiomas sobre tipos de datos necesaria para la validación de los arquetipos. La figura 6.24 muestra la interfaz web de *Archeck*. Para realizar la validación de un arquetipo es necesario que sea proporcionado en formato ADL (parte izquierda de la figura). Además, si el arquetipo es una especialización de otro, también deben proporcionarse todos los ancestros del arquetipo. El informe de validación muestra los errores de modelado y proporciona enlaces a la representación de los arquetipos en OWL.

El rendimiento global del proceso de validación es aceptable. El tiempo medio del método de validación es de 1,3 segundos por arquetipo y es prácticamente el mismo en ambos repositorios. Este tiempo depende en gran medida del rendimiento del razonador y en los axiomas elegidos para la representación OWL. El tiempo medio de procesamiento de las definiciones ADL y la generación de la representación OWL es de unos 0,5 segundos por arquetipo. La generación de los arquetipos en OWL utiliza un proceso dirigido por modelos en el que los artefactos software (archivos ADL, ontologías OWL, esquemas XML y terminologías) son representados como modelos y el proceso de transformación está guiado por reglas de transformación. Esta arquitectura penaliza ligeramente el rendimiento del proceso, aunque supone una mejora en el mantenimiento del software.

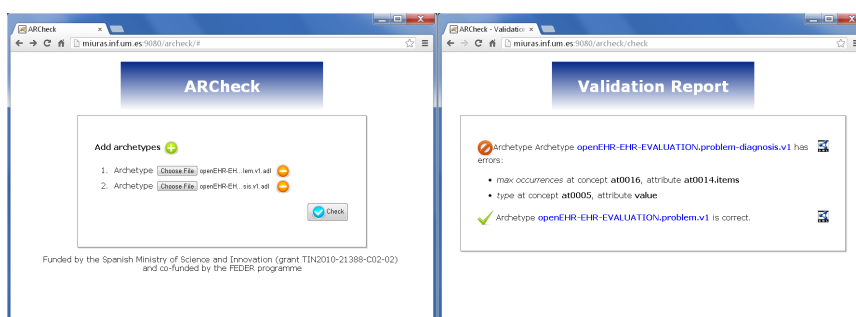


Figura 6.24: Interfaz web de la herramienta de validación *Archeck*



## Capítulo 7

# Interoperabilidad de Modelos Clínicos

### 7.1. Introducción

La iniciativa CIMI tiene como objetivo definir una infraestructura que permita la interoperabilidad de modelos clínicos [31]. En la comunidad científica la mayor parte de las propuestas realizadas hasta el momento están centradas en la interoperabilidad de la información clínica. Sin embargo, la motivación de CIMI está fundada en que los modelos clínicos (arquetipos) cada vez tienen un papel más importante en los sistemas de información sanitarios, especialmente controlando la entrada y visualización de la información. Por tanto, la reutilización de los modelos clínicos ha despertado el interés de los principales organismos e instituciones en el área de la Informática Médica.

En este capítulo se describe el transformador de arquetipos *Poseacle Converter*. Esta herramienta no sólo es capaz de convertir arquetipos entre los estándares ISO EN 13606 y openEHR, sino que también realiza la transformación de extractos de HCE. La arquitectura de interoperabilidad propuesta se ha construido sobre las ontologías *Poseacle*, que se definen en torno a una Ontología Común que agrupa el conocimiento consensuado en las arquitecturas basadas en el modelo dual. Asimismo, la herramienta se caracteriza por implementar el proceso de transformación utilizando técnicas de Ingeniería de Modelos.

A continuación se presenta el marco de trabajo *Enconsertable* cuyo objetivo es definir una infraestructura de interoperabilidad de modelos clínicos basados en el modelo dual. La propuesta es una evolución de *Poseacle Converter* que se caracteriza por el uso de la tecnología OWL en todas las etapas del proceso de transformación. Además, la metodología *Encorsetable* promueve la colaboración de los expertos en el dominio HCE en la definición de las correspondencias entre estándares.

## 7.2. Poseacle Converter

Poseacle Converter es una herramienta de conversión de arquetipos y extractos de HCE entre estándares basados en la arquitectura de modelo dual. El proceso de conversión ha sido diseñado utilizando ontologías que representan el modelo de arquetipos y los modelos de información. Esta característica permite que la herramienta pueda transformar arquetipos entre distintos formalismos, concretamente entre ADL y OWL, siendo los arquetipos OWL instancias de las ontologías. En la sección 5.3 puede consultarse la representación *Poseacle* de arquetipos y modelos de referencia en OWL.

La herramienta ha sido desarrollada como soporte de la metodología Poseacle para la interoperabilidad semántica de sistemas sanitarios [59]. El marco de trabajo está constituido por varias ontologías que caracterizan el dominio de la HCE basada en el modelo dual. La semántica de las estructuras y tipos de datos de un modelo de información se expresa en una ontología que se integra con otra que define las restricciones del modelo de arquetipos. El resultado es una ontología que representa un modelo de referencia en OWL cuyas instancias representan a los arquetipos. Por último, en el marco ontológico definido en Poseacle juega un papel fundamental la denominada *Ontología Común*, que contiene el conocimiento global en el dominio de HCE y actúa como representación intermedia en los procesos de transformación.

Los ontologías *Poseacle* constituyen el soporte formal del proceso de interoperabilidad entre estándares de modelo dual, aunque no forman parte propiamente de la implementación de la herramienta de conversión. Poseacle Converter ha sido implementado gracias al uso de técnicas de Desarrollo de Software Dirigido por Modelos (véase sección 2.4) que han permitido la definición de correspondencias entre estándares y la ejecución de los procesos de transformación.

### 7.2.1. Ontología Común

La Ontología Común ha sido introducida en la arquitectura de Poseacle para facilitar la definición de correspondencias entre estándares de modelo dual. Este objetivo se consigue gracias a que esta ontología define los conceptos suficientes para representar cualquier estructura de datos de los modelos de información de HCE. La versión actual de la ontología es fruto del proceso de integración de las ontologías de los modelos de referencia de ISO EN 13606 y openEHR aplicando la metodología propuesta en [57].

El beneficio esperado del uso de la Ontología Común es la reducción del esfuerzo de integración de nuevos estándares. Por un lado, es suficiente con alinear las entidades del nuevo estándar y la Ontología Común. Por otro lado, fomenta la reutilización de conocimiento consensuado en el dominio HCE, que puede ser aprovechado en iniciativas como CIMI [31].



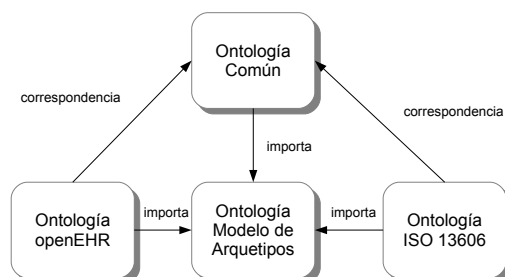


Figura 7.1: Marco ontológico de la arquitectura *Poseacle*

El marco ontológico de Poseacle está constituido por cuatro ontologías: la ontología del modelo de arquetipos, las ontologías de los modelos de referencia ISO EN 13606 y openEHR, y la Ontología Común. La figura 7.1 muestra la relación entre estas ontologías. La ontología de arquetipos es utilizada por las otras tres ontologías, dando lugar a modelos de referencia *arquetipables*. Entre las ontologías de cada estándar y la Ontología Común se definen correspondencias que permiten la transformación de arquetipos y extractos de HCE.

### 7.2.2. Contexto tecnológico

La herramienta Poseacle Converter ha sido desarrollada en tres etapas. El proyecto se inició en 2007 y tuvo como objetivo representar arquetipos ADL como instancias de una ontología OWL [130]. Seguidamente se definió e implementó la infraestructura para la transformación de arquetipos entre los estándares ISO EN 13606 y openEHR [128]. Esta infraestructura fue adaptada en la última etapa para la transformación de extractos de HCE entre ambos estándares [129].

El marco de trabajo Poseacle está soportado conceptualmente por el uso de ontologías definidas en OWL que representan el conocimiento de los estándares de HCE basados en el modelo dual, en particular, de ISO EN 13606 y openEHR. La interoperabilidad semántica entre estos estándares requiere establecer correspondencias entre sus ontologías como soporte para la transformación de arquetipos y extractos. La comunidad de la Web Semántica que respalda el desarrollo de OWL no ofrecía en 2007 ninguna recomendación de lenguaje para la definición de los alineamientos entre ontologías OWL. Además, la representación de arquetipos ADL en OWL impuso unos requerimientos técnicos que no pudieron ser abordados por la tecnología OWL. Por estos motivos, la arquitectura e implementación de Poseacle Converter están ligadas a la Ingeniería de Modelos que ha proporcionado soluciones a algunas carencias técnicas de la Web Semántica.

La comunidad de Ingeniería de Modelos ha sido muy activa en el desarrollo de herramientas y en la integración con otros espacios tecnológicos como XML y la Web Semántica (véase sección 2.4). El consorcio OMG ha liderado la iniciativa MDA (Model Driven Architecture) que define un marco conceptual y una serie de estándares como soporte de la Ingeniería de Modelos. Sin embargo, el proceso de elaboración y consolidación de estos estándares ha sido lento, lo que ha favorecido la aparición de herramientas y lenguajes que han sido acogidos como estándares de facto por la comunidad. La implementación de Poseacle Converter se ha realizado utilizando herramientas de la comunidad que no forman parte del marco MDA, como son el framework EMF, el lenguaje RubyTL para la transformación de modelos y el lenguaje MOFScript para obtener representaciones textuales a partir de modelos.

La rápida madurez del paradigma del desarrollo de software dirigido por modelos ha motivado el establecimiento de puentes con otros espacios tecnológicos [114]. Gracias a ello otras comunidades de usuarios han podido aprovecharse del marco conceptual y las herramientas desarrolladas en la comunidad de Ingeniería de Modelos. En este sentido, la especificación ODM (Ontology Definition Metamodel) [149] del OMG ha sido la que más ha influenciado el desarrollo de Poseacle Converter. ODM prescribe una serie de metamodelos para varios formalismos de representación de conocimiento (RDFS, OWL) y una propuesta de correspondencias entre UML y estos metamodelos. Las recomendaciones de ODM han sido implementadas por algunas herramientas (por ejemplo, Protégé), lo que ha permitido expresar las ontologías que utiliza Poseacle Converter como metamodelos procesables en Ingeniería de Modelos.

### 7.2.3. Representación de arquetipos en OWL

Poseacle Converter define una metodología genérica para la representación del conocimiento expresado en arquetipos. La metodología está respaldada por la capacidad de la Ingeniería de Modelos de actuar como representación intermedia en procesos de transformación entre distintos formalismos. La implementación actual está orientada a los estándares ISO EN 13606 y openEHR, y a la representación del conocimiento utilizando el lenguaje OWL, aunque puede ser aplicada a cualquier otra representación ontológica y estándar basado en la arquitectura de modelo dual.

ADL es el lenguaje utilizado para la definición de arquetipos. Este lenguaje es una sintaxis concreta del modelo de arquetipos (AOM) expresado en UML. La Fundación openEHR ofrece un conjunto de herramientas para trabajar con el lenguaje ADL que incluyen un analizador sintáctico que instancia un modelo AOM a partir de la definición de un arquetipo y un generador de documentos XML conformes al esquema del modelo de arquetipos.

El framework EMF implementa el puente entre el espacio tecnológico XML y la Ingeniería de Modelos. EMF ofrece varias herramientas de importación de metamodelos que están expresados en artefactos software tales como interfaces en el lenguaje de programación Java o esquemas XML. Esta funcionalidad permite obtener una representación del esquema XML del modelo de arquetipos (AOM) como metamodelo en Ingeniería de Modelos. Además, EMF ofrece el soporte necesario para instanciar modelos AOM a partir de documentos XML. Por lo tanto, gracias al framework EMF podemos tratar a los arquetipos como modelos. En definitiva, la combinación de las herramientas de openEHR y EMF nos permite establecer un puente entre los espacios tecnológicos de los arquetipos y los modelos.

El marco ontológico Poseacle expresa los modelos de referencia como ontologías OWL de modo que los arquetipos son instancias de esas ontologías. Para poder aprovechar las herramientas de transformación de modelos disponibles en la comunidad DSDM es necesario que las ontologías Poseacle sean también expresadas como metamodelos. El puente que conecta el lenguaje OWL con la Ingeniería de Modelos está definido en la propuesta ODM que recoge las correspondencias entre los lenguajes UML y OWL. La definición de estas correspondencias está expresada en el lenguaje de transformación QVT, que actualmente no ha sido completamente implementado. Afortunadamente el editor de ontologías Protégé ha implementado la exportación de ontologías OWL como interfaces en el lenguaje de programación Java. El proceso de exportación hace uso de las reglas de transformación definidas en la propuesta ODM. Por lo tanto, aprovechando la posibilidad de importación de interfaces Java como metamodelos ofrecida por EMF, conseguimos representar las ontologías de Poseacle como metamodelos sobre los que podemos definir las correspondencias entre los modelos de referencia de los estándares.

La transformación de arquetipos ADL en OWL comienza con la conversión del modelo que representa al arquetipo en un modelo conforme a la ontología del estándar al que pertenece. Las correspondencias entre el modelo de arquetipos y la ontología se definen utilizando el lenguaje de transformación RubyTL. El resultado de la transformación es el modelo *semántico* del arquetipo, es decir, se obtiene la representación del arquetipo enriquecida semánticamente gracias a la ontología del modelo de referencia.

La última etapa del proceso de conversión consiste en representar el arquetipo *semántico* en OWL. Esta tarea se ha implementado como una transformación modelo a texto definida en el lenguaje MOFScript. Las reglas de transformación procesan las declaraciones del arquetipo semántico y generan su representación equivalente en OWL.

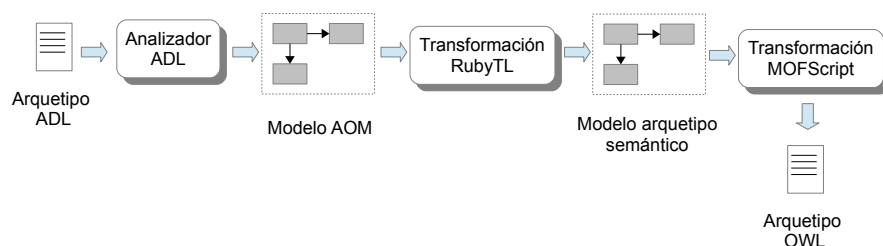


Figura 7.2: Proceso de conversión de un arquetipo en OWL

En resumen, la figura 7.2 ilustra las fases del proceso de transformación de arquetipos ADL en OWL que son las siguientes:

- Representación de un arquetipo ADL como modelo, en concreto, un modelo conforme al metamodelo de arquetipos (AOM).
- Transformación del modelo AOM en un arquetipo *semántico*, es decir, un modelo de la ontología correspondiente al estándar del arquetipo.
- Generación de la representación en OWL del arquetipo *semántico*.

El proceso de conversión de arquetipos en OWL se ha implementado para los estándares ISO EN 13606 y openEHR. Para cada estándar ha sido necesario definir las correspondencias entre el modelo de arquetipos y la ontología del estándar, y además las reglas de generación de los arquetipos *semánticos* al lenguaje OWL. Estos son los requisitos para la introducción de un nuevo estándar en la arquitectura de Poseacle Converter.

#### 7.2.4. Transformación de arquetipos

La transformación de arquetipos entre estándares basados en la arquitectura dual tiene un interés doble. Por una parte, la comunidad openEHR es pionera en el desarrollo de arquetipos, disponiendo de repositorios públicos con cientos de ellos. En cambio, el estándar ISO EN 13606 ha adoptado la arquitectura dual más tarde y el número de arquetipos disponibles es escaso. Por este motivo, interesa disponer del conocimiento expresado en los arquetipos openEHR para el estándar ISO EN 13606. Por otro lado, el intercambio de extractos entre sistemas sanitarios implementados con diferentes estándares requiere que el arquetipo en el que está expresado el extracto en el sistema origen sea transformado según el estándar del sistema destino para que el extracto pueda ser interpretado en la recepción. Así mismo, las correspondencias aplicadas en la transformación del arquetipo son la guía para transformación de los extractos.

El framework Poseacle Converter implementa la transformación de arquetipos entre los estándares ISO EN 13606 y openEHR. Para la definición de las correspondencias entre estos estándares no ha sido necesario utilizar herramientas de alineamiento ontológico, ya que el estándar ISO EN 13606 puede considerarse, en parte, un subconjunto del modelo de referencia de openEHR. Las correspondencias implementadas pueden consultarse en [125].

La definición de correspondencias se ve favorecida por dos características que están presentes en ambos modelos de información: estructuras de datos *estructuras* y atributos para preservar la semántica. Los estándares ISO EN 13606 y openEHR contienen las clases `CLUSTER` y `ELEMENT`, que pueden ser utilizadas estructuras de datos *constructoras*, es decir, pueden representar prácticamente cualquier clase de un modelo de información: los datos de la HCE tales como una medida o texto son almacenados en contenedores de tipo `ELEMENT` y éstos se organizan en estructuras de datos flexibles formadas por `CLUSTERS`. Por otro lado, el estándar ISO EN 13606 declara el atributo `meaning` en todas las estructuras de datos del modelo de información. Este atributo permite identificar el significado de la información en el sistema origen cuando se intercambian extractos de HCE. Esta característica del estándar ISO EN 13606 es clave para preservar la semántica en el proceso de transformación de arquetipos y extractos. Sin embargo, el modelo de referencia de openEHR no incluye un atributo con el mismo propósito, de modo que para la integración en la arquitectura de Poseacle Converter ha sido necesario utilizar otro atributo común que no es utilizado actualmente (`uid`).

El proceso de alineamiento ontológico aplicado a los estándares ISO EN 13606 y openEHR se ha basado en la aplicación de los siguientes criterios:

- Dos clases que representan el mismo concepto en el dominio de HCE son alineadas. Un ejemplo sería la correspondencia de las clases `DV_TEXT` de openEHR y `TEXT` del estándar ISO EN 13606, ya que ambas clases representan datos de tipo textual.
- Correspondencia con un concepto general. Si en la ontología destino no existe un concepto semánticamente equivalente, se procede a establecer la correspondencia con otro concepto más general cuyo significado englobe al concepto que queremos hacer corresponder. A este grupo de correspondencias pertenece el alineamiento de la clase `OBSERVATION` de openEHR con la clase `ENTRY` de ISO EN 13606.
- Las propiedades de los conceptos alineados se hacen corresponder del siguiente modo:
  - Dos propiedades son alineadas si tienen el mismo significado y sus tipos (rango de la propiedad) son compatibles, es decir, existe

una alineación entre los tipos de datos. Un ejemplo sería la alineación de la propiedad `items` de la clase `CLUSTER` de openEHR con la propiedad `parts` de la clase `CLUSTER` de ISO EN 13606.

- El concepto destino es una estructura de datos *extensible*. Por ejemplo, la propiedad `encoding` de la clase `OBSERVATION` de openEHR no tiene correspondencia en la clase `ENTRY` de ISO EN 13606. Sin embargo, la clase `ENTRY` contiene la propiedad `data` que permite albergar cualquier tipo de estructura `CLUSTER/ELEMENT`. En particular, la información asociada por la propiedad `encoding` puede ser almacenada en un `ELEMENT` y se establece en su propiedad `meaning` el significado original de la información (“openehr:OBSERVATION:encoding”).
- En otro caso, no se establece la correspondencia de la propiedad. Esto implica que en la transformación habrá pérdida de información. Un ejemplo de este caso sería la propiedad `property` de la clase `PQ` (Physical Quantity) de ISO EN 13606. La clase `PQ` corresponde con la clase `DV_QUANTITY` de openEHR, que no tiene un atributo para almacenar la información de `property` ni tampoco es una estructura de datos extensible. Así pues, las restricciones aplicadas al atributo `property` de `PQ` y la información asociada al mismo almacenada en los extractos se perderían en el proceso de transformación.

El proceso de definición de correspondencias necesario para la transformación de arquetipos es bidireccional, es decir, debe establecerse en cada sentido de la transformación. Además, las correspondencias no son implementadas directamente, sino que han de ser interpretadas de acuerdo a la Ontología Común. En otras palabras, el alineamiento de las ontologías de los estándares ISO EN 13606 y openEHR requiere definir 4 grupos de correspondencias: (1) de openEHR a la Ontología Común, (2) de la Ontología Común a ISO EN 13606, (3) de ISO EN 13606 a la Ontología Común y (4) de la Ontología Común a openEHR. Las correspondencias entre las ontologías son implementadas utilizando el lenguaje de transformación de modelos RubyTL.

El framework Poseacle Converter definido para la conversión de arquetipos a OWL es extendido para introducir la Ontología Común necesaria para el proceso de transformación de arquetipos. Además, se incorpora un generador de arquetipos ADL a partir de modelos implementado con el lenguaje de transformación MOFScript.

La figura 7.3 muestra las etapas del proceso de transformación de un arquetipo entre estándares de arquitectura dual que se desarrolla en las siguientes etapas:

- Representación del arquetipo origen en ADL como *modelo AOM* y posteriormente en un modelo conforme al metamodelo del estándar origen (*modelo arquetipo origen*).
- Transformación del *modelo arquetipo origen* en un modelo conforme al metamodelo de la Ontología Común (*modelo arquetipo común*).
- Transformación del *modelo arquetipo común* en un modelo conforme al metamodelo de la ontología destino (*modelo arquetipo destino*).
- Obtención del arquetipo destino a partir de la generación de la representación en ADL/OWL del *modelo arquetipo destino*.

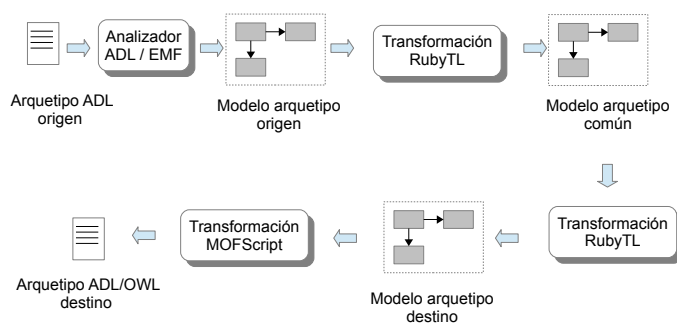


Figura 7.3: Proceso de transformación de arquetipos

### 7.2.5. Limitaciones

Poseacle Converter ha demostrado ser una herramienta efectiva para la conversión de arquetipos entre los estándares ISO EN 13606 y openEHR [126]. El principal inconveniente comunicado por los usuarios de la herramienta ha sido no ofrecer la posibilidad de ajustar las correspondencias establecidas entre los estándares. La implementación de esta funcionalidad tendría un gran impacto en la arquitectura y en el marco conceptual. Por un lado, las correspondencias no se expresan utilizando un lenguaje de transformación estandarizado (por ejemplo, QVT), ni tampoco se definen en una notación de alineamiento de ontologías. Esto dificulta notablemente que un usuario pueda editar las reglas de transformación en lenguajes específicos como RubyTL. Por otro lado, las correspondencias no se establecen directamente entre estándares, sino a través de la Ontología Común, lo que obliga a conocer bien esta ontología para expresar correctamente las reglas de transformación.

Otro inconveniente de la arquitectura de la herramienta es el coste de mantenimiento de la Ontología Común al introducir un nuevo modelo de referencia. La versión actual de la Ontología Común es el resultado del proceso de integración de dos estándares relacionados como son ISO EN 13606 y openEHR. La introducción de un modelo de información con características que no puedan expresarse en la Ontología Común, como por ejemplo HL7 CDA, supondría la necesidad de adaptar y extender la Ontología Común.

Por último, la introducción de un nuevo modelo de información no sólo supondría cambios en la Ontología Común, sino que además apenas se podría reutilizar la implementación de las etapas comunes del proceso de transformación. El único módulo que es común a cualquier estándar es el encargado de la transformación de un arquetipo ADL en un modelo AOM. La generación de arquetipos semánticos en OWL o la generación de modelos en ADL es específica de cada modelo de referencia y no puede ser reutilizada.

### 7.3. Encorsetable

El marco de trabajo Encorsetable surge como respuesta a las limitaciones de Poseacle Converter. En primer lugar, se ha eliminado la Ontología Común como componente intermediario de los procesos de transformación. A pesar del beneficio que aporta utilizar una ontología que representa el conocimiento común y consensuado en el dominio HCE, el coste de mantenimiento de esta ontología es alto y además dificulta notablemente la definición de correspondencias entre estándares. En segundo lugar, la introducción de un modelo de referencia en la arquitectura no supone ningún esfuerzo de implementación, ya que la representación en OWL se obtiene automáticamente, tanto del modelo de información como de los arquetipos, y también la generación de arquetipos ADL a partir de su representación en OWL. Por último, la característica más destacada de la metodología Encorsetable es la promoción de la colaboración de los usuarios (expertos del dominio HCE) en la definición de las correspondencias entre estándares. El resultado es un marco de trabajo que permite el intercambio de modelos clínicos entre estándares de HCE basados en la arquitectura dual.

El marco formal que soporta la metodología Encorsetable es el lenguaje OWL-DL. Por una parte, el modelo ontológico gira en torno una ontología denominada *Ontología de Restricciones* que ofrece las construcciones para expresar los modelos de información y los modelos clínicos (véase sección 5.5). Por otro lado, la transformación de modelos clínicos es el resultado de un proceso de inferencia que se apoya en la definición de las correspondencias entre los modelos de información en OWL.



La transformación de modelos clínicos entre estándares se apoya en el uso de modelos de referencia *arquetipables*. En la metodología Encorsetable la representación de los modelos de información en OWL es generada automáticamente en base a los conceptos de la Ontología de Restricciones. El resultado es una ontología que define las restricciones que caracterizan a las entidades del modelo de información, pero no la semántica de su modelo de datos, es decir, las instancias de esta ontología no representan extractos, sino modelos clínicos. Dado que todos los modelos de referencia son ontologías definidas utilizando las construcciones de la Ontología de Restricciones y los modelos clínicos son instancias de esas ontologías, la transformación de modelos clínicos consiste en clasificar esos modelos en la ontología del modelo de información destino. Por lo tanto, la metodología Encorsetable para la transformación de modelos clínicos consiste en la definición de correspondencias entre los modelos de información para que pueda realizarse la clasificación.

Los expertos del dominio HCE están a cargo del alineamiento de los modelos de referencia en la metodología Encorsetable. Las correspondencias deben expresarse en formato OWL de acuerdo a las ontologías de los modelos de información generadas automáticamente. Esta tarea puede ser asistida por herramientas de alineamiento ontológico que sean capaces de exportar las correspondencias como axiomas OWL (véase apartado 4.2.1). En general, los formatos de alineamiento ontológico que utilizan estas herramientas tienen una traducción directa en el lenguaje OWL, aunque en algunos casos el lenguaje OWL-DL no es suficientemente expresivo para representar cierto tipo de correspondencias.

Las organizaciones responsables de la estandarización de los estándares HCE están colaborando en la tarea de armonizar los modelos de información de la HCE. El estándar ISO EN 13606 incluye arquetipos de referencia que ilustran cómo representar las estructuras de datos de estándares como HL7 utilizando arquetipos y el modelo de información de ISO EN 13606. Por su parte, la Fundación openEHR ha definido una serie de reglas informales de transformación entre el modelo de referencia openEHR y el estándar ISO EN 13606 [188]. Asimismo, las correspondencias implementadas en la herramienta Poseacle Converter también pueden ser reutilizadas [125]. En definitiva, el esfuerzo realizado por la comunidad HCE para la armonización de estándares de HCE podría ser implementado en el marco común de transformación de modelos clínicos proporcionado por el framework Encorsetable.

La tarea de alineamiento de los modelos de información entre estándares HCE no es sencilla, a pesar de la ayuda que ofrecen las herramientas de alineamiento ontológico y las propuestas de armonización entre estándares. El problema es debido, en parte, a que los modelos de referencia expresan el dominio HCE con distinto nivel de abstracción. Por ejemplo, el estándar

ISO EN 13606 define estructuras de datos bastante genéricas, ya que está orientado al intercambio de extractos HCE, y en cambio, openEHR utiliza un modelo de información más preciso que ha sido diseñado para el desarrollo de sistemas de información sanitarios. La diferencia de expresividad entre los modelos de información dificulta notablemente la definición declarativa de correspondencias entre los estándares, y en particular, la implementación como axiomas OWL.

La metodología Encorsetable propone el uso de arquetipos como herramienta facilitadora de la definición de correspondencias. La idea es emplear el concepto de arquetipo en su sentido formal (modelo de restricciones) en vez de utilizarlo para declarar un modelo clínico. En otras palabras, la propuesta es usar arquetipos para extender los modelos de información genéricos con el fin de definir estructuras de datos especializadas que puedan alinearse más fácilmente con otros modelos de información. Por ejemplo, el estándar ISO EN 13606 ofrece únicamente las clases `CLUSTER` y `ELEMENT` para definir estructuras de datos multiparte como tablas o árboles. Por ejemplo, un arquetipo podría especializar el modelo de información restringiendo la clase `CLUSTER` para definir una estructura de datos tabla. El resultado es una nueva estructura de datos que puede ser alineada, por ejemplo, con la clase `ITEM_TABLE` de openEHR.

La metodología propuesta ha sido aplicada a los estándares basados en el modelo de arquetipos ISO EN 13606 y openEHR. Otros estándares definidos según los principios de la arquitectura dual también pueden ser incorporados al framework Encorsetable, como el modelo de *templates* de HL7 CDA.

### 7.3.1. Introducción al proceso de transformación

En este apartado se desarrolla un ejemplo de transformación de un arquetipo openEHR al estándar ISO EN 13606 (véase figura 7.4). A través del ejemplo se introducen las etapas de la transformación de un modelo clínico y el papel del lenguaje OWL en el proceso. Este primer ejemplo utiliza estructuras de datos que son prácticamente equivalentes en ambos estándares, por lo que la correspondencia no requiere del uso de otros arquetipos como *intermediarios* en el proceso de transformación.

La representación en OWL de las entidades de los modelos de información se caracteriza por definir los tipos de datos y atributos como clases OWL (véase 5.5.2). La generación automática de la ontología de los modelos de referencia nombra a las clases que representan a los atributos utilizando como prefijo el identificador de la clase que lo contiene, por ejemplo, la clase `CLUSTER_items` corresponde con el atributo `items` de la estructura de datos `CLUSTER`. Por último, se han introducido espacios de nombres para distinguir las clases de las ontologías de cada estándar, "13606" para el estándar ISO EN 13606 y "openehr" para la ontología openEHR.

```
CLUSTER[at0000] matches { -- Arquetipo openEHR
  items cardinality matches {1..*; ordered; unique}
  matches {
    ELEMENT[at0001] occurrences matches {1..2} {
      value matches {
        DV_TEXT matches { * }
      }
    }
  }
}
```

Figura 7.4: Arquetipo a transformar con la metodología Encorsetable

La definición de correspondencias entre estándares consiste en el alineamiento de las clases de las ontologías que definen a los modelos de información. La metodología Encorsetable propone el uso de dos tipos de relaciones entre clases: equivalencia ( $\equiv$ ) e inclusión ( $\subseteq$ ). Esta propuesta simplifica el proceso de alineamiento y además tiene el beneficio de que todas las herramientas y lenguajes de alineamiento ontológico soportan estos dos tipos de relaciones, y que la semántica de estas correspondencias puede ser expresada directamente en OWL. La figura 7.5 incluye un extracto de las correspondencias entre los estándares ISO EN 13606 y openEHR que son aplicadas en el proceso de transformación del arquetipo de la figura 7.4. Como puede apreciarse en las correspondencias, las estructuras de datos son prácticamente equivalentes, salvo por el nombrado de algunas propiedades y clases. Por ejemplo, el atributo `items` de la clase `CLUSTER` de openEHR corresponde con el atributo `parts` de la clase `CLUSTER` en ISO EN 13606.

El proceso de transformación comienza con la conversión del arquetipo ADL en formato OWL. El framework Encorsetable representa a los arquetipos como clases o instancias OWL. Al inicio del proceso es necesario que los arquetipos sean tratados como instancias de las ontologías de los modelos de referencia para poder así ejecutar las correspondencias entre los estándares. Un término arquetipado se define como una instancia que se relaciona con las instancias que declaran a los atributos que redefine. La representación OWL del arquetipo de la figura 7.4 se muestra en la figura 7.6. Las entidades de la Ontología de Restricciones se distinguen utilizando el espacio de nombres `co`. El resultado de esta etapa es un documento OWL con la declaración como instancias de todos los términos del arquetipo.

```

_____ Correspondencias _____
openehr:CLUSTER ≡ 13606:CLUSTER
openehr:CLUSTER_items ≡ 13606:CLUSTER_parts
openehr:ELEMENT ≡ 13606:ELEMENT
openehr:ELEMENT_value ≡ 13606:ELEMENT_value
openehr:DV_TEXT ≡ 13606:TEXT

_____ Axiomas OWL _____
EquivalentClasses: openehr:CLUSTER, 13606:CLUSTER
EquivalentClasses: openehr:CLUSTER_items, 13606:
    CLUSTER_parts
EquivalentClasses: openehr:ELEMENT, 13606:ELEMENT
EquivalentClasses: openehr:ELEMENT_value, 13606:
    ELEMENT_value
EquivalentClasses: openehr:DV_TEXT, 13606:TEXT

```

Figura 7.5: Extracto de las correspondencias entre los modelos de referencia ISO EN 13606 y openEHR

```

Individual: at0000
  Types: openehr:CLUSTER
  Facts: co:has_property at0000_items

Individual: at0000_items
  Types: openehr:CLUSTER_items
  Facts: co:min_cardinality 1, co:is_ordered true,
        co:is_unique true, co:has_range at0000_items_at0001

Individual: at0000_items_at0001
  Types: co:RANGE
  Facts: co:has_type at0001, co:min_occurrences 1,
        co:max_occurrences 2

Individual: at0001
  Types: openehr:ELEMENT
  Facts: co:has_property at0001_value

Individual: at0001_value
  Types: openehr:ELEMENT_value
  Facts: co:has_range at0002_value_range

Individual: at0001_value_range
  Types: co:RANGE_SIMPLE
  Facts: co:has_type at0002_value_type

Individual: at0001_value_range_type
  Types: openEHR:DV_TEXT

```

Figura 7.6: Representación como instancias OWL del arquetipo

La transformación del arquetipo consiste en un proceso de inferencia ejecutado por un razonador OWL. La base de conocimiento sobre la que se realiza el razonamiento está formada por: (1) el documento OWL que contiene la definición del arquetipo, (2) la ontología de openEHR, (3) la ontología de ISO EN 13606, (4) la Ontología de Restricciones y (4) el documento que define las correspondencias en OWL. Las inferencias obtenidas en esta etapa clasifican las instancias del arquetipo en la ontología del modelo de referencia destino, esto es, ISO EN 13606 (véase figura 7.7).

```
Individual: at0000
Types: 13606:CLUSTER

Individual: at0000_items
Types: 13606:CLUSTER_parts

Individual: at0001
Types: 13606:ELEMENT

Individual: at0001_value
Types: 13606:ELEMENT_value

Individual: at0001_value_range_type
Types: 13606:TEXT
```

Figura 7.7: Inferencias obtenidas tras la clasificación del arquetipo OWL de la figura 7.6

El proceso de transformación finaliza con la conversión de las instancias del arquetipo de nuevo al lenguaje ADL. Los axiomas de la base de conocimiento son filtrados para seleccionar las relaciones que establecen las instancias del arquetipo con la Ontología de Restricciones y la ontología modelo de referencia destino. De este modo, las restricciones originales establecidas en el arquetipo son preservadas, ya que se utilizan las entidades de la Ontología de Restricciones, y el arquetipo resultante restringe las estructuras de datos y atributos del modelo de información destino (véase figura 7.8).

```

CLUSTER[at0000] matches { -- Arquetipo 13606
  parts cardinality matches {1..*; ordered; unique} matches {
    ELEMENT[at0002] occurrences matches {1..2} {
      value matches {
        TEXT matches { * }
      }
    }
  }
}

```

Figura 7.8: Resultado de la transformación del arquetipo de la figura 7.4

### 7.3.2. Modelos de referencia con distinta expresividad

La metodología Encorsetable simplifica el proceso de alineación de modelos de referencia gracias al uso en exclusiva de las relaciones de equivalencia e inclusión. En el ejemplo presentado en el apartado anterior estas dos relaciones son suficientes para expresar las correspondencias necesarias en el proceso de transformación debido a que las definiciones de las estructuras de datos `CLUSTER` y `ELEMENT` en los modelos de referencia ISO EN 13606 y openEHR son prácticamente equivalentes. Sin embargo, el modelo de referencia openEHR define estructuras de datos que no tienen una correspondencia directa con ISO EN 13606, tales como `ITEM_TABLE` o `ITEM_TREE`.

La clase `ITEM_TABLE` de openEHR representa una estructura de datos para la organización de los componentes de la HCE en forma de tabla. El modelo de información de openEHR declara la clase `ITEM_TABLE` con un atributo multivaluado (`rows`) de tipo `CLUSTER` que representa las filas de la tabla (véase la definición XML de la clase en la figura 7.9), es decir, define una estructura de datos multiparte cuyo componente en el primer nivel es de tipo `CLUSTER`. La estructura de datos podría ser representada en ISO EN 13606 aplicando la siguiente regla informal de transformación: un `ITEM_TABLE` de openEHR equivale a un `CLUSTER` de ISO EN 13606 formado sólo por otros `CLUSTERS` que representan las filas de la tabla.

```

<xs:complexType name="ITEM_TABLE">
  <xs:complexContent>
    <xs:extension base="ITEM_STRUCTURE">
      <xs:sequence>
        <xs:element name="rows" type="CLUSTER"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

Figura 7.9: Definición XML de la estructura de datos `ITEM_TABLE` de openEHR

La regla de transformación de un `ITEM_TABLE` de openEHR a ISO EN 13606 no puede ser representada utilizando relaciones de inclusión y equivalencia entre las entidades de los modelos de referencia. La propuesta de la metodología Encorsetable para facilitar la definición de la transformación consiste en extender el modelo de referencia destino (ISO EN 13606) utilizando arquetipos. La idea es definir arquetipos que enriquezcan las estructuras de datos del modelo de referencia destino con el fin de que puedan declararse relaciones básicas de equivalencia o inclusión con las estructuras de datos del modelo de referencia origen. Estos arquetipos actúan como *intermediarios* en el proceso de definición de las correspondencias entre los modelos de referencia. La figura 7.10 muestra un ejemplo de arquetipo *intermediario* que define la estructura de datos *tabla*.

```

CLUSTER[at0000] matches { -- tabla
  parts cardinality matches {0..*; ordered} matches {
    CLUSTER[at0001] occurrences matches {0..*} { -- filas
      *
    }
  }
}

```

Figura 7.10: Arquetipo ISO EN 13606 que representa una estructura de datos tabular

Las correspondencias entre los modelos de referencia se definen a través de relaciones entre clases. La introducción de los arquetipos *intermediarios* en el proceso de transformación como extensiones del modelo de referencia implica que éstos también deben ser expresados como clases. Sin embargo, en el apartado anterior los arquetipos fueron definidos como instancias para que pudieran ser transformados a través de un proceso de clasificación de instancias. Las representaciones de arquetipos como clases o instancias OWL propuestas en la metodología Encorsetable son equivalentes en este contexto gracias a que tienen la misma interpretación en el modelo de arquetipos, es decir, las representaciones OWL como instancias y clases pueden ser convertidas de nuevo en arquetipos. La figura 7.11 contiene la definición del arquetipo *tabla* de ISO EN 13606 como clases OWL. Los términos del arquetipo corresponden con clases que se declaran subclases de la entidad del modelo de referencia que especializan. Estas clases se relacionan a través de la propiedad `has_property` con las clases que definen las restricciones de los atributos. Por último, se emplea el espacio de nombrado "13606\_tabla" para identificar las clases de la ontología del arquetipo.

```
Class: 13606_tabla:at0000
  SubClassOf:
    13606:CLUSTER,
    co:has_property some 13606_tabla:at0000_items

Class: 13606_tabla:at0000_items
  SubClassOf:
    13606:CLUSTER_items,
    co:is_ordered true,
    co:is_unique false,
    co:has_range some (has_type only 13606_tabla:at0001)

Class: 13606_tabla:at0001
  SubClassOf:
    13606:CLUSTER
```

Figura 7.11: Representación como clases del arquetipo *tabla* de ISO EN 13606

Desde el punto de vista de la representación en OWL, no hay distinción entre las clases que definen el modelo de referencia y los arquetipos, ya que en ambos casos todas las declaraciones corresponden con clases en OWL (tipos de datos, atributos, términos arquetipados y atributos redefinidos). Así pues, los arquetipos pueden emplearse como clases del modelo de referencia en la definición de las correspondencias entre estándares. La



figura 7.12 contiene las relaciones entre la estructura de datos `ITEM_TABLE` de openEHR y el arquetipo *tabla* de ISO EN 13606. Nótese que se han empleado relaciones de equivalencia ( $\equiv$ ) expresando que las correspondencias se establecen son bidireccionales.

```
openehr:ITEM_TABLE  $\equiv$  13606_tabla:at0000  
openehr:ITEM_TABLE_rows  $\equiv$  13606_tabla:at0000_items
```

Figura 7.12: Correspondencias entre la estructura de datos `ITEM_TABLE` de openEHR y el arquetipo *tabla* ISO EN 13606

El proceso de transformación se desarrolla en las mismas etapas descritas en el apartado anterior: (1) transformación del arquetipo origen en instancias OWL, (2) preparación de la base de conocimiento, (3) proceso de razonamiento, (4) filtrado de axiomas y (5) conversión del arquetipo transformado en ADL. La figura 7.13 incluye de transformación de un ejemplo de arquetipo `ITEM_TABLE` de openEHR, las inferencias obtenidas en la etapa de razonamiento (el espacio de nombres “arquetipo” identifica las instancias del arquetipo) y el arquetipo transformado a ISO EN 13606. Finalmente, cabe destacar el papel que tiene el arquetipo intermediario *tabla* en el proceso de inferencia que ha permitido que las instancias del arquetipo hayan sido clasificadas en el modelo de referencia destino.

El arquetipo transformado en la figura 7.13 ha migrado correctamente las restricciones asociadas a las filas en el arquetipo origen (cardinalidad, orden y unicidad), pero no representa una estructura de datos tabular en ISO EN 13606. El resultado de la transformación es un `CLUSTER` que puede contener hasta 5 componentes ordenados de cualquier tipo compatible con el atributo, es decir, tanto `CLUSTERS` como `ELEMENTS`. La pérdida de semántica en la transformación se debe fundamentalmente a que el componente del arquetipo *tabla* que define el contenido de las filas (*at0001*) no tiene correspondencia con la estructura de datos `ITEM_TABLE` (véase la definición de correspondencias en la figura 7.12). Por lo tanto, el proceso de transformación basado en clasificación de instancias no garantiza que se mantenga la semántica de las correspondencias cuando se introducen arquetipos *intermediarios* en el proceso.

```

----- Arquetipo origen -----
ITEM_TABLE[at0000] matches { -- Arquetipo openEHR
  rows cardinality matches {0..5; ordered} matches {
    *
  }
}

----- Inferencias -----

Individual: arquetipo:at0000
Types: 13606_tabla:at0000, 13606:CLUSTER

Individual: arquetipo:at0000_rows
Types: 13606_tabla:at0000_parts, 13606:CLUSTER_parts

----- Arquetipo transformado -----

CLUSTER[at0000] matches { -- arquetipo - ISO EN 13606
  parts cardinality matches {0..5; ordered} matches {
    *
  }
}

```

Figura 7.13: Transformación de un arquetipo `ITEM_TABLE` de openEHR a ISO EN 13606

El problema de la pérdida de semántica en la transformación de la figura 7.13 se resolvería mezclando las definiciones del arquetipo transformado y del arquetipo *tabla*. Sin embargo, el modelo de arquetipos no incluye ninguna relación entre arquetipos que permita fusionar las definiciones. Por un lado, la especialización de arquetipos es una relación de conformidad de extractos de HCE, pero no es una relación de herencia de definiciones. Por otro lado, la relación de slot es utilizada para componer arquetipos y no permite mezclar las restricciones. En cambio, el lenguaje OWL ofrece el axioma *SubClassOf* para declarar la relación de inclusión entre clases, y por tanto, la herencia de definiciones. Así pues, la mezcla de arquetipos se resuelve a través de la representación OWL basada en clases y utilizando axiomas de inclusión.

La combinación de un arquetipo transformado y un arquetipo intermedio se realiza antes de la etapa de filtrado de axiomas. El arquetipo expresado como instancias se convierte en un arquetipo basado en clases que especializa (axioma *SubClassOf*) las definiciones del arquetipo intermedio. Por ejemplo, el término raíz del arquetipo de la figura 7.13 se declara subclase del término raíz del arquetipo *tabla* y esta relación también se es-

tablece entre los atributos que ambos arquetipos redefinen (véase figura 7.14). Gracias a estas relaciones, la raíz del arquetipo transformado se asocia con el componente *at0001* del arquetipo *tabla* y da lugar a un arquetipo ISO EN 13606 que define una estructura de datos tabular manteniendo las restricciones del arquetipo original (véase figura 7.14).

```

_____ Axiomas para la combinación de arquetipos _____

Class: arquetipo:at0000
  SubClassOf: 13606_tabla:at0000

Classes: arquetipo:at0000_rows
  SubClassOf: 13606_tabla:at0000_parts

_____ Arquetipo transformado _____

CLUSTER[at0000] matches { -- arquetipo - ISO EN 13606
  parts cardinality matches {0..5; ordered} matches {
    CLUSTER[at0001] ocurrences matches {0..*} matches { * }
  }
}

```

Figura 7.14: Fusión del arquetipo transformado con el arquetipo *tabla*

En este apartado se ha desarrollado un ejemplo que motiva el uso de arquetipos como elementos intermediarios en el proceso de transformación. La propuesta consiste en extender el modelo de referencia destino utilizando arquetipos cuando no exista la posibilidad de definir correspondencias directas entre las estructuras de datos. La introducción de arquetipos *intermediarios* implica la modificación del proceso añadiendo una etapa de fusión del arquetipo transformado tras el proceso de inferencia y los arquetipos intermediarios que han formado parte de la transformación. El resultado es que el arquetipo origen puede enriquecerse en el proceso añadiendo la semántica necesaria a las estructuras de datos y al mismo tiempo se mantiene la simplicidad de la definición de correspondencias entre modelos de referencia basada en axiomas de equivalencia e inclusión.

### 7.3.3. Arquetipos abstractos

Los arquetipos intermediarios tienen un papel clave en la definición de correspondencias entre los modelos de referencia en la arquitectura Encorsetable. Sin embargo, a veces estos arquetipos introducen restricciones que limitan el conjunto de arquetipos que pueden ser transformados. La transformación desarrollada en el apartado anterior es un buen ejemplo de este problema. Las restricciones que establece el arquetipo que va a ser transformado sobre el atributo `rows` (véase figura 7.13) son compatibles con las restricciones del arquetipo *tabla* con el que se establece la correspondencia (véase figura 7.10): la cardinalidad `0..5` es consistente con `0..*` y las restricciones de orden y unicidad son iguales. Una definición diferente de la restricción de orden, como por ejemplo, no permitir filas repetidas, significaría que el arquetipo no podría ser transformado al no ser compatible con el arquetipo intermediario.

La metodología Encorsetable introduce la noción de *arquetipo abstracto* para definir aquellos arquetipos que actúan como intermediarios en el proceso de transformación. Dado que estos arquetipos no son creados para representar extractos de HCE, las restricciones que definen pueden estar *incompletas*. Esta característica es de gran utilidad en la metodología de transformación, ya que ofrece más flexibilidad en la definición de arquetipos intermediarios, y por tanto hace que sean aplicables en la transformación de más arquetipos. Un ejemplo sería dejar *vacía* la definición de la cardinalidad del atributo `parts` en el arquetipo *tabla*, lo que permitiría transformar cualquier arquetipo de tipo `ITEM_TABLE` de openEHR sin importar las restricciones que establezca en el atributo `rows`. La definición de arquetipos abstractos está soportada parcialmente por el analizador sintáctico del lenguaje ADL, que permite dejar vacía la restricción de cardinalidad y ocurrencias utilizando el símbolo `*`. La figura 7.15 muestra la definición del arquetipo *tabla* que no restringe la cardinalidad del atributo `parts`.

```
CLUSTER[at0000] matches { -- tabla ISO EN 13606
  parts cardinality matches {*} matches {
    CLUSTER[at0001] occurrences matches {0..*} { -- rows
      *
    }
  }
}
```

Figura 7.15: Arquetipo abstracto *tabla* ISO EN 13606

#### 7.3.4. Arquetipos abstractos como filtro de correspondencias

El concepto de *arquetipo abstracto* es una herramienta útil para la definición de correspondencias entre modelos de referencia. El arquetipo *tabla* de ISO EN 13606 introduce una nueva estructura de datos en el modelo de referencia que ha permitido establecer una correspondencia con la clase `ITEM_TABLE` de openEHR. En otras palabras, el arquetipo abstracto ha actuado como componente *intermediario* para la transformación de arquetipos cuando el modelo de referencia destino es menos expresivo que el origen. En este apartado se motiva el papel de los arquetipos abstractos cuando el nivel de expresividad del modelo de referencia origen es menor que el destino, por ejemplo, de ISO EN 13606 a openEHR.

En el apartado 7.3.1 se ha presentado el proceso de transformación a través de las estructuras de datos `CLUSTER` y se ha ilustrado mediante una transformación de openEHR a ISO EN 13606. La transformación en el sentido contrario sería similar y utilizaría las mismas correspondencias (véase figura 7.5). Sin embargo, el modelo de referencia destino, esto es, openEHR, contiene estructuras de datos multiparte con mayor riqueza semántica que un simple `CLUSTER`, tales como `ITEM_TABLE` o `ITEM_LIST`. Una transformación *plana* basada en las correspondencias de la figura 7.5 no aprovecharía la expresividad del modelo de información openEHR.

La figura 7.16 presenta un arquetipo de tipo `CLUSTER` en el estándar ISO EN 13606. La estructura de datos que define este arquetipo es una tabla de dos columnas que puede contener hasta diez filas. La transformación de este arquetipo al modelo de referencia openEHR daría como resultado un arquetipo prácticamente idéntico, que sería correcto en términos de estructura de datos, pero que no representaría de la forma más adecuada la estructura de datos en openEHR.

```
CLUSTER[at0000] matches { -- arquetipo ISO EN 13606
  parts cardinality matches {0..10; ordered} matches {
    CLUSTER[at0001] occurrences matches {0..10} matches {
      -- filas
      parts cardinality matches {2..2; ordered} matches {
        ELEMENT[at0002] matches { * } -- columna 1
        ELEMENT[at0003] matches { * } -- columna 2
      }
    }
  }
}
```

Figura 7.16: Ejemplo de arquetipo `CLUSTER` ISO EN 13606

El arquetipo *tabla* de ISO EN 13606 introducido en los apartados anteriores (véase figura 7.10) caracteriza una estructura de datos tabular organizada utilizando `CLUSTERS`. La alineación definida con la clase `ITEM_TABLE` de openEHR es bidireccional, es decir, se puede aplicar para hacer transformaciones de openEHR a ISO EN 13606 y viceversa. Así pues, el arquetipo de la figura 7.16 podría ser transformado a openEHR aplicando esas correspondencias, ya que la estructura de datos que define es *compatible* con una tabla.

La metodología Encorsetable propone el uso de *Archeck* como herramienta para comprobar la *compatibilidad* del arquetipo que va a ser transformado con un arquetipo abstracto. Archeck implementa un método de validación de la consistencia de las relaciones de especialización que puede emplearse para determinar si un arquetipo es compatible con otro, es decir, un arquetipo es compatible con un arquetipo abstracto si la relación de especialización con ese arquetipo es consistente.

Archeck actúa como selector de correspondencias en la arquitectura Encorsetable. Los arquetipos abstractos establecen correspondencias con el modelo de referencia destino (véase figura 7.12) que sólo deben ser aplicadas para aquellos arquetipos que representen una estructura de datos compatible para ser transformados con ellas. Por lo tanto, el arquetipo de la figura 7.16 sería transformado a una estructura de datos `ITEM_TABLE` de openEHR (véase figura 7.17), ya que el arquetipo es compatible con una *tabla* (arquetipo abstracto) y pueden ser aplicadas sus correspondencias.

```

ITEM_TABLE[at0000] matches { -- Arquetipo openEHR
  rows cardinality matches {0..10; ordered} matches {
    CLUSTER[at0001] occurrences matches {0..10} matches {
      items cardinality matches {2..2; ordered} matches {
        ELEMENT[at0002] ...
        ELEMENT[at0003] ...
      }
    }
  }
}

```

Figura 7.17: Resultado de la transformación del arquetipo de la figura 7.16 aplicando las correspondencias del arquetipo abstracto *tabla*

El proceso de transformación es igual al que ha sido descrito en los apartados anteriores. La etapa de inferencia obtiene que el término *at0000* es un `ITEM_TABLE` y que su atributo `parts` corresponde con el atributo `rows` de `ITEM_TABLE`. El resto de conceptos son clasificados según las co-

correspondencias generales entre el estándar ISO EN 13606 y openEHR, por ejemplo, *at0001* se clasifica como un `CLUSTER` openEHR. Las restricciones del término *at0001* se mantienen en el proceso de transformación, ya que su definición está conectada con el atributo `parts` que a su vez se ha ligado al atributo `rows` de `ITEM_TABLE`.

En resumen, los arquetipos abstractos en la arquitectura de transformación Encorsetable son los componentes clave para establecer el alineamiento de los modelos de referencia. Por una parte, actúan como arquetipos intermediarios cuando el modelo de referencia destino tiene menor expresividad que el origen, facilitando así la definición de correspondencias. Por otra parte, aumenta la precisión semántica del proceso de transformación cuando el modelo de referencia destino define estructuras de datos más ricas semánticamente que el origen. En este caso, los arquetipos abstractos actúan como filtro selector de correspondencias y el método `Archeck` se emplea para comprobar si el arquetipo que va a ser transformado cumple el filtro establecido por un arquetipo abstracto, y por tanto, pueden ser aplicadas las correspondencias que tiene asociadas.

### 7.3.5. Plantillas de transformación

En los apartados anteriores se ha mostrado la utilidad de los arquetipos *abstractos* como extensión del modelo de referencia destino o como filtro de correspondencias en origen de la transformación. La combinación de arquetipos abstractos tanto en el origen como en el destino de la transformación permite aplicar un proceso de definición de correspondencias basado en *plantillas*. En este apartado se introduce el proceso de alineamiento de modelos de referencia utilizando plantillas a través de la transformación de un arquetipo de tipo `ENTRY` ISO EN 13606 en un arquetipo `OBSERVATION` de openEHR. Este ejemplo también ilustra la utilidad de los enlaces terminológicos cuando son empleados en arquetipos abstractos que actúan como filtro.

Las plantillas son una herramienta habitual para la definición de transformaciones de Ingeniería de Modelos, especialmente en las transformaciones modelo a texto. La popularidad de las plantillas se debe fundamentalmente a que permiten representar el *esqueleto* del artefacto software que se pretende generar, fijando la ubicación de los elementos variables de la transformación. Un ejemplo es la generación de arquetipos en formato OWL y ADL en la herramienta Poseacle Converter, donde se han empleado plantillas MOFScript que son rellenas con la información proporcionada por los modelos que representan a los arquetipos semánticos.

La analogía de las plantillas puede ser trasladada al proceso de transformación de modelos clínicos en el que los arquetipos abstractos definen las plantillas y además capturan la información variable de la generación. La figura 7.18 muestra un ejemplo de arquetipo *plantilla* que representa

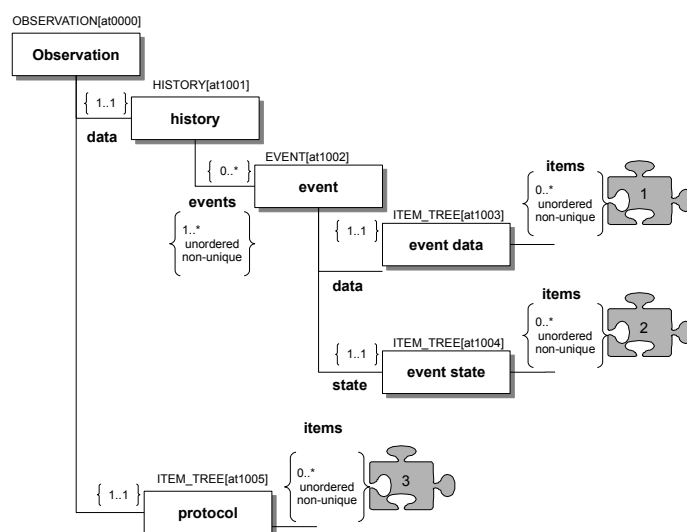


Figura 7.18: Arquetipo *plantilla* OBSERVATION de openEHR

la estructura básica de un arquetipo OBSERVATION en openEHR donde la parte variable de la transformación se muestra gráficamente como *piezas de puzzle*.

Las restricciones necesarias para completar una plantilla deben ser proporcionadas por el arquetipo que va a ser transformado. La plantilla sólo debe ser aplicada para aquellos arquetipos que definan una estructura de datos *compatible* con la plantilla, es decir, que declaren las restricciones suficientes para rellenar la plantilla. Para expresar el conjunto de arquetipos a los que puede ser aplicada una plantilla usamos un arquetipo abstracto que caracterice a los arquetipos de ISO EN 13606 que semánticamente representan una *observación* y cuya estructura de datos permita capturar las restricciones incompletas la plantilla.

El modelo de referencia ISO EN 13606 representa cualquier tipo de entrada clínica utilizando la estructura de datos genérica ENTRY. La transformación de una entrada ISO EN 13606 tiene una correspondencia directa en la clase GENERIC\_ENTRY de openEHR. No obstante, el modelo de información de openEHR también define otros tipos de entradas como observaciones (OBSERVATION), evaluaciones (EVALUATION), etc. Si, por ejemplo, una entrada almacena la información de algún hecho clínico *observable*, la correspondencia más adecuada debería realizarse con la estructura de datos OBSERVATION, y el arquetipo que captura los datos de la entrada podría expresar que representa el resultado de una observación clínica a través de un enlace terminológico a un concepto que identifique el tipo de observación que contiene. En general, el conjunto de arquetipos de tipo ENTRY en



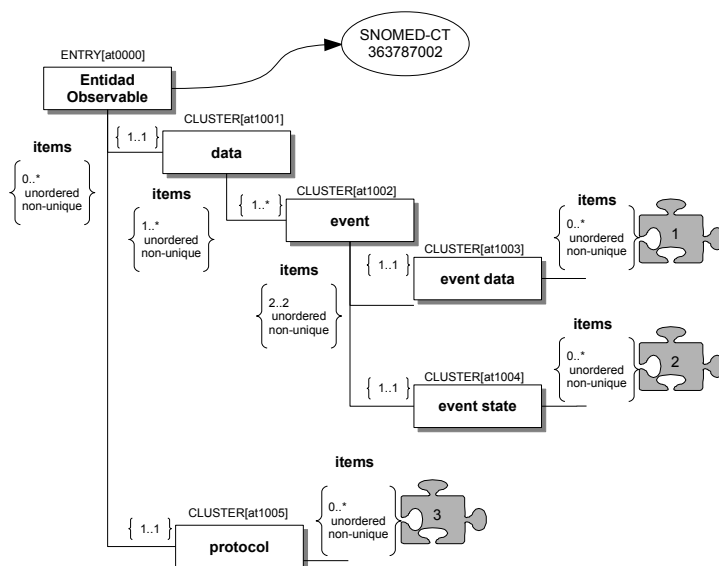


Figura 7.19: Arquetipo `ENTRY` que caracteriza a una entidad observable en ISO EN 13606

el estándar ISO EN 13606 que representan a una *observación* podrían caracterizarse mediante un arquetipo abstracto cuyo concepto raíz establezca un enlace con el concepto “Entidad observable” de SNOMED-CT (véase figura 7.19).

La definición de la plantilla de transformación finaliza con las correspondencias entre los dos arquetipos abstractos. La alineación de las declaraciones de estos arquetipos no necesariamente debe ser completa, es decir, no todos los términos del arquetipo abstracto origen deben tener una correspondencia con un término en el arquetipo abstracto destino. El arquetipo abstracto en el origen de la transformación actúa como un filtro de los arquetipos que pueden ser transformados con la plantilla. Una vez que se ha comprobado que un arquetipo es compatible con el *filtro*, lo único que interesa de éste son las restricciones necesarias para completar el arquetipo abstracto que define la plantilla en el destino de la transformación. Por lo tanto, sólo hay que establecer las correspondencias entre los términos raíz y entre los componentes variables de la plantilla, como por ejemplo, los atributos `items` de los términos `1003`. En las figuras 7.18 y 7.19 se han identificado las correspondencias con una numeración y la definición formal se muestra en la figura 7.20. Por último, nótese que en la definición de los arquetipos abstractos se han utilizado identificadores a partir de 1000 para los términos que definen la estructura de la plantilla y los términos que están alineados comparten el mismo identificador.

```
13606_filtro:ENTRY_at0000 ≡
  openehr_plantilla:OBSERVATION_at0000
13606_filtro:CLUSTER_at1003_items ≡
  openehr_plantilla:ITEM_TREE_at1003_items
13606_filtro:CLUSTER_at1004_items ≡
  openehr_plantilla:ITEM_TREE_at1004_items
13606_filtro:CLUSTER_at1005_items ≡
  openehr_plantilla:ITEM_TREE_at1005_items
```

Figura 7.20: Correspondencias de una plantilla de transformación

El objetivo principal de la metodología Encorsetable es promover la participación de los expertos en el dominio HCE en el proceso de alineación de estándares. Una plantilla es una herramienta conceptual que facilita notablemente la definición de correspondencias entre modelos de información. Las plantillas constan de tres componentes: (1) un arquetipo que define la *plantilla de transformación* en el modelo de información destino, (2) un arquetipo *filtro* en el modelo de información origen, que actúa como selector de la plantilla y que captura las definiciones variables, y (3) las correspondencias entre ambos arquetipos. El ejemplo presentado en este apartado muestra que el uso de plantillas simplifica la definición de correspondencias porque sólo es necesario prestar atención a los términos del arquetipo que realmente definen la información clínica, dejando a un lado el resto de componentes que representan la estructura de datos propia de cada estándar. Por último, los enlaces terminológicos en los arquetipos elevan el proceso de transformación a un nivel *semántico*. En el ejemplo, una entrada en ISO EN 13606 tiene una correspondencia genérica en el estándar openEHR, pero el uso de enlaces terminológicos permite que sea clasificada semánticamente de forma más precisa como un tipo de entrada concreto en openEHR.

### 7.3.6. Proceso de transformación

La metodología Encorsetable define un proceso de transformación de modelos clínicos basado en la capacidad de inferencia del lenguaje OWL-DL. Por una parte, los modelos de información son representados como ontologías en OWL y las correspondencias entre ellos son expresadas mediante axiomas OWL. Por otra parte, los arquetipos se definen como instancias de la ontología del modelo de información al que pertenecen y son clasificados en el modelo de información al que van a ser transformados utilizando razonamiento. Además, los arquetipos también participan en el proceso de transformación como filtros de correspondencias y como plantillas de transformación. En ambos casos son representados como clases en OWL.

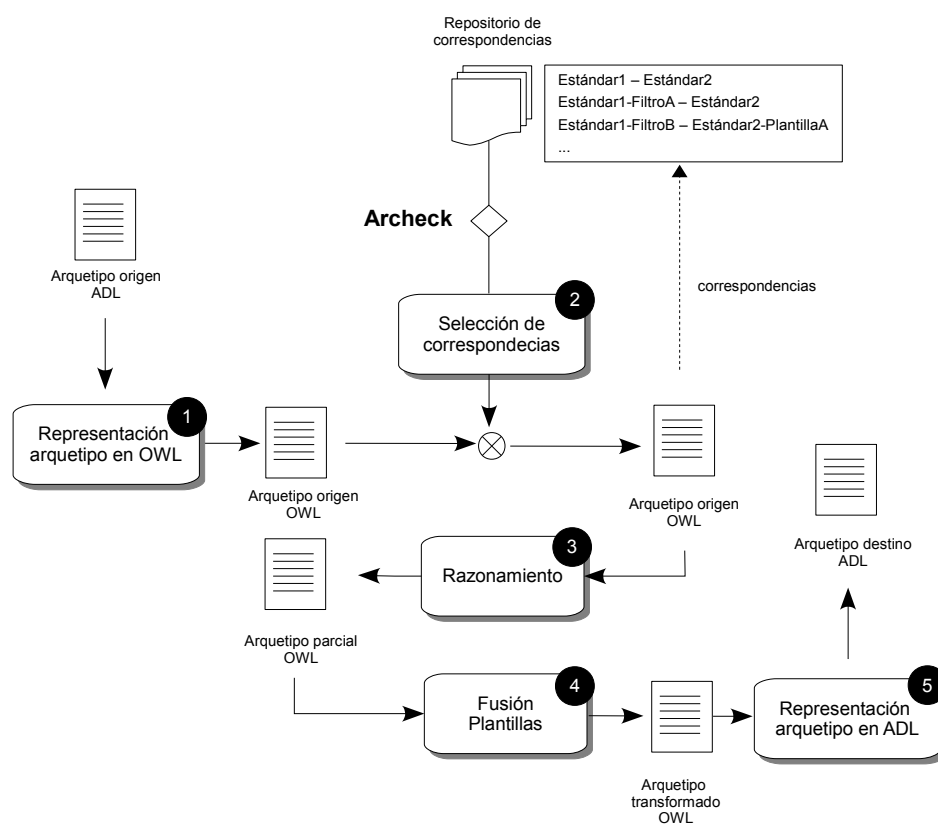


Figura 7.21: Proceso de transformación *Encorsetable*

El proceso de transformación se desarrolla en varias etapas en las que se requiere que el arquetipo esté expresado en el formato adecuado (véase figura 7.21). A continuación se describen estas etapas y los cambios de la representación necesarios en cada una de ellas:

1. Representación del arquetipo en OWL. La entrada del proceso de transformación es un arquetipo en formato ADL que es transformado a OWL. El arquetipo es representado como instancias de la ontología del modelo de referencia en el que está definido.
2. Selección de correspondencias. La base de conocimiento sobre la que se ejecuta el proceso está formada por las ontologías de los modelos de referencia de los estándares origen y destino, arquetipos *abstractos* de ambos estándares, el arquetipo representado en OWL y las ontologías con las correspondencias aplicables en la transformación. Entre cada pareja de estándares existe una ontología que declara las correspondencias que no dependen del arquetipo que va ser transformado, y que, por lo tanto, forman parte de la base de conocimiento. El

resto de correspondencias son seleccionadas según la estructura y la semántica del arquetipo. Estas correspondencias están asociadas con arquetipos abstractos que actúan como selectores y sólo son aplicadas si el arquetipo que va a ser transformado es *conforme* con el arquetipo que representa el filtro.

3. Razonamiento. La base de conocimiento contiene los axiomas que relacionan los modelos de referencia según el arquetipo que va a ser transformado. El proceso de inferencia clasifica las instancias que definen al arquetipo en las clases de las ontologías del modelo de información y arquetipos abstractos (plantillas) del estándar destino. La transformación que se obtiene a través de razonamiento es *parcial*, ya que la clasificación de instancias no es suficiente para fusionar el arquetipo con los arquetipos abstractos del modelo de referencia destino.
4. Fusión de plantillas. En esta etapa el arquetipo representado como instancias es transformado en clases OWL que establecen relaciones de inclusión con los términos de los arquetipos abstractos en los que hayan sido clasificados en la etapa de razonamiento. Estas relaciones lógicas consiguen combinar las definiciones de los arquetipos abstractos (plantillas) con el arquetipo *parcial* obtenido en la etapa anterior.
5. Representación del arquetipo en ADL. El resultado de la etapa anterior es un arquetipo representado mediante clases en OWL. Dado que las reglas de generación de arquetipos en OWL son bidireccionales, el proceso de transformación finaliza convirtiendo el arquetipo transformado en ADL y conforme al modelo de referencia destino.

### 7.3.7. Framework *Encorsetable*

El framework *Encorsetable* está formado por un marco ontológico construido en torno a la Ontología de Restricciones, un conjunto de módulos encargados de gestionar las representaciones en OWL de los modelos de información y arquetipos, y varios módulos que ejecutan la transformación de los modelos clínicos.

La base de conocimiento sobre la que se ejecuta el proceso de transformación está formada por varias ontologías que definen los modelos de referencia, los arquetipos abstractos y las correspondencias. La ontología asociada a un modelo de referencia caracteriza las restricciones de un modelo de información y permite instanciar modelos clínicos. Los arquetipos abstractos son ontologías que extienden los modelos de información con nuevos conceptos que facilitan la definición de correspondencias. La relación entre los modelos de referencia se establece utilizando axiomas

OWL que soportan el proceso de inferencia (transformación) de los modelos clínicos. Finalmente, todas estas ontologías se declaran utilizando las construcciones proporcionadas por la Ontología de Restricciones.

Una característica destacable del marco ontológico propuesto es la *representación dual* de los arquetipos en OWL. El proceso de transformación está soportado por las correspondencias entre estándares definidas utilizando axiomas estructurales en OWL, es decir, axiomas que relacionan clases. Estos axiomas permiten clasificar las instancias del modelo de información origen en las clases del modelo de información destino. Así pues, los arquetipos deben ser definidos como instancias de las ontologías de los modelos de información. Por otro lado, los arquetipos que sirven de apoyo para la definición de correspondencias, esto es, los arquetipos abstractos, son extensiones de los modelos de información que deben ser definidos como clases en OWL. Por último, el uso de arquetipos abstractos como plantillas requiere la conversión del arquetipo que está siendo transformado de instancias a clases OWL.

Los módulos encargados de la representación de los arquetipos y modelos de información en OWL son los siguientes:

- *RM2OWL*. Este módulo genera una ontología OWL a partir de la definición del modelo de referencia como esquema XML. Las reglas aplicadas en esta conversión se detallan en la sección 5.5.2.
- *ADL2Archetype*. La entrada del proceso de transformación es un arquetipo en formato ADL que es convertido por este módulo en una representación conforme al modelo de arquetipos (AOM).
- *Archetype2Instances*. La etapa de razonamiento necesita que el arquetipo sea representado como instancias de la ontología del modelo de referencia en el que está definido. Las reglas que definen esta transformación han sido descritas en la sección 5.5.3 y son implementadas en este módulo.
- *Instances2Archetype*. Las reglas de transformación de arquetipos en OWL son bidireccionales y este módulo está encargado de transformar un arquetipo definido como instancias OWL en la representación del modelo de arquetipos, es decir, realiza la tarea inversa al módulo *Archetype2Instances*.
- *Archetype2Classes*. Los arquetipos pueden ser representados como clases OWL que especializan las entidades de las ontologías de los modelos de referencia (véase sección 5.5.4). Ésta es la definición adecuada para los arquetipos *abstractos* y para la fusión de un arquetipo con una plantilla. Este módulo genera esta representación a partir de la definición basada en el modelo de arquetipos.

- *Classes2Archetype*. Módulo que implementa la transformación inversa de *Archetype2Classes* y que está soportado por la bidireccionalidad de las reglas de transformación descritas en la sección 5.5.4.
- *Instances2Classes*. La equivalencia semántica entre las representaciones basadas en clases e instancias se obtiene al compartir la misma interpretación en el modelo de arquetipos (AOM). Este módulo implementa la transformación de instancias a clases y resulta de la combinación de los módulos *Instances2Archetype* y *Archetype2Instances*.
- *Archetype2ADL*. La última etapa del proceso de transformación en la conversión de un arquetipo al formato ADL. Este módulo se encarga de esta tarea y utiliza la representación basada en el modelo de arquetipos.

La figura 7.22 muestra la organización de los módulos en el marco de trabajo Encorsetable. En el centro de la figura se incluyen los módulos encargados de la ejecución de la transformación.

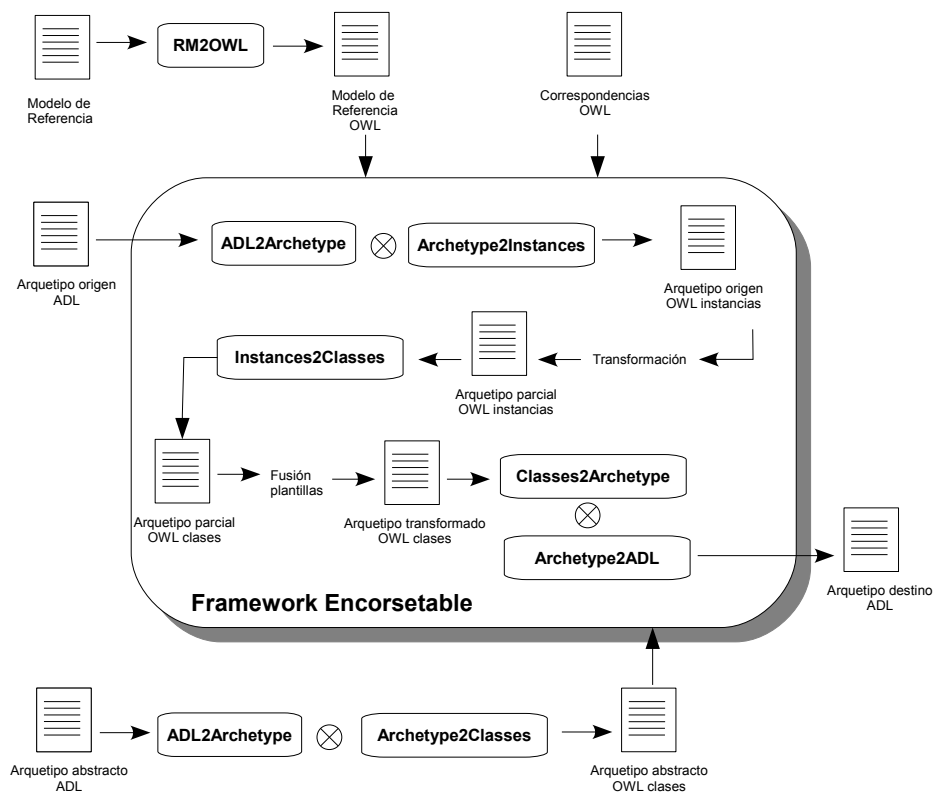


Figura 7.22: Framework *Encorsetable*

## Capítulo 8

# Ingeniería de Modelos aplicada a la Arquitectura Dual

### 8.1. Introducción

Hoy en día la mayoría de los sistemas sanitarios no utilizan estándares. Esta situación supone un serio obstáculo para lograr la interoperabilidad de la HCE, ya que la información clínica de los pacientes suele quedar inaccesible en los centros donde es obtenida. Por otro lado, las arquitecturas de HCE basadas en estándares son complejas, en especial, las que utilizan la arquitectura dual. La inversión en el desarrollo de sistemas compatibles con estas arquitecturas es alta debido a la dificultad conceptual del modelado dual y a la necesidad de conocer los modelos de información que utilizan en estos estándares.

La Ingeniería de Modelos es un área de la Ingeniería del Software que trata con la automatización de los procesos de desarrollo de software. La idea principal de esta iniciativa es promover un papel más activo de los modelos que especifican los sistemas de información, de forma que puedan guiar la generación de código de las aplicaciones. En este capítulo se analizan los beneficios de la aplicación del desarrollo de software dirigido por modelos a los estándares de HCE basados en la arquitectura dual. Las conclusiones de este análisis han sido implementadas en un generador de aplicaciones sanitarias denominado ArchForms.

En capítulos anteriores se ha estudiado la representación en OWL de la arquitectura dual como soporte para la implementación de tareas como la validación y transformación de modelos clínicos. En este capítulo también se aprovecha la representación ontológica de los arquetipos, pero esta vez el objetivo es generar aplicaciones sanitarias que guíen de forma correcta y precisa la captura y visualización de información clínica.

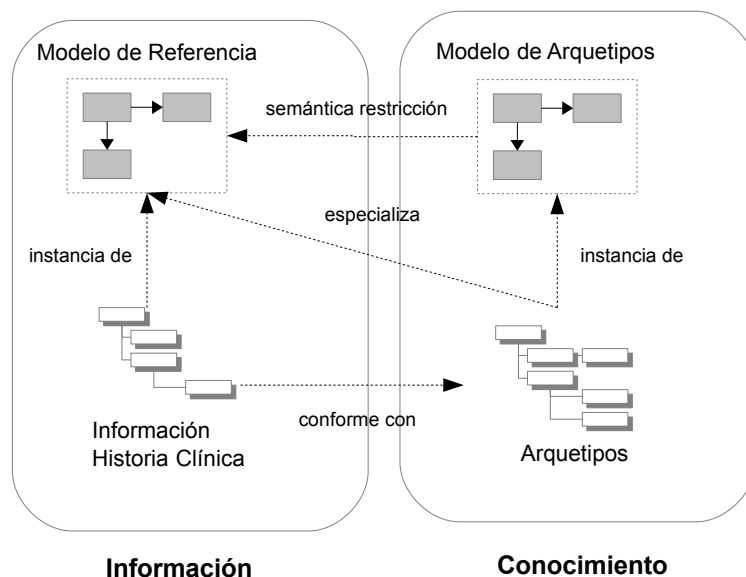


Figura 8.1: Niveles de modelado en la arquitectura dual

## 8.2. Niveles de modelado en la arquitectura dual

La arquitectura de modelo dual se propuso inicialmente como una solución a los problemas de evolución y mantenimiento de los sistemas de información, en especial, en dominios complejos como la medicina [9]. La propuesta consiste básicamente en extraer del dominio los requisitos de información estables y representarlos en un *modelo de referencia*, y definir los conceptos del dominio en base a combinaciones legales de las entidades del modelo de referencia denominadas *arquetipos*. La figura 8.1 representa los dos niveles de modelado promovidos por la arquitectura dual. Por una parte, los arquetipos representan el nivel de conocimiento y son expresados como instancias del *modelo de arquetipos*. Por otro lado, la información de la historia clínica está representada por el modelo de referencia y es conforme a las definiciones de los arquetipos. Entre el modelo de arquetipos y el modelo de referencia se establece una relación con *semántica de restricción*, que sin embargo no está definida formalmente. Lo mismo sucede con la relación de conformidad de la información clínica y los arquetipos. Finalmente, en la figura también se introduce una relación entre los arquetipos y el modelo de referencia que expresa que cualquier arquetipo es considerado una *especialización* de alguna entidad del modelo de referencia.



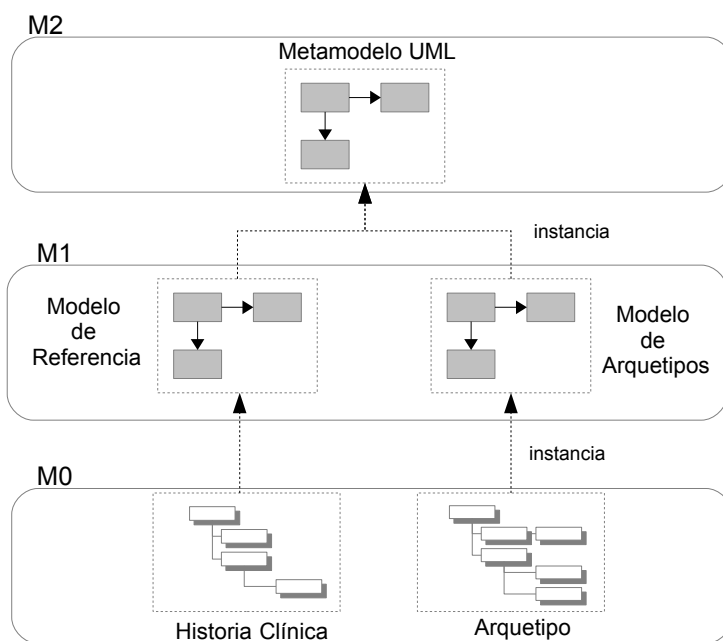


Figura 8.2: Relación de la arquitectura dual con la propuesta de modelado del OMG

Los estándares de HCE especifican el modelo de referencia y el modelo de arquetipos como diagramas de clases UML. Ante esta definición, cabe plantearse, en primer lugar, qué relación existe entre ambos modelos. La figura 8.2 representa la situación de los modelos de la arquitectura dual en la propuesta de modelado del OMG. El lenguaje UML se sitúa en el nivel M2 que representa el metamodelo o lenguaje de modelado. El modelo de referencia y el de arquetipos son modelos UML, y por tanto, instancias de los constructores especificados en este lenguaje. Así pues, los modelos se localizan formalmente en el nivel M1 de la arquitectura de modelado. Por último, los extractos de HCE y los arquetipos están en el nivel de datos.

El modelo de arquetipos es independiente de cualquier modelo de referencia. Esta característica impide que sus definiciones pueden estar asociadas de algún modo a los modelos de información. Esta idea se ilustra a través de un ejemplo de instancia de arquetipo conforme al modelo de arquetipos. La figura 8.3 muestra un arquetipo openEHR de tipo `CLUSTER`. Los términos del arquetipo son objetos de tipo `C_COMPLEX_OBJECT` y las restricciones de atributos se expresan en objetos `C_MULTIPLE_ATTRIBUTE`, en el caso de que el atributo sea multivaluado como en el ejemplo. Nótese que

las referencias al modelo de información de openEHR que se expresan en los atributos `rmTypeName` y `rmAttributeName` son definidas simplemente como cadenas.

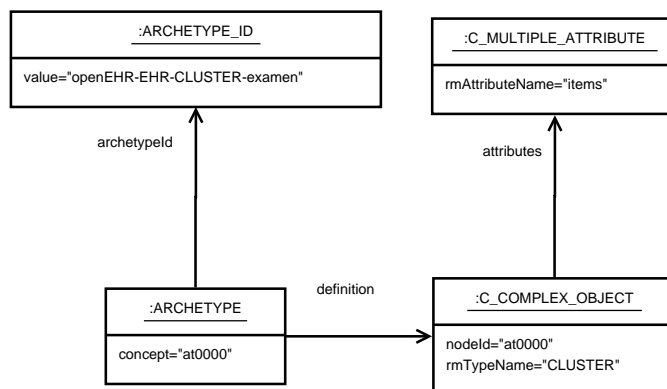


Figura 8.3: Relación entre arquetipos y el modelo de referencia

Por lo tanto, podemos afirmar que la especificación actual de la arquitectura dual en los estándares ISO EN 13606 y openEHR no cumple los requisitos necesarios para que pueda ser integrada en un arquitectura de modelado. En primer lugar, los arquetipos no establecen ninguna asociación con el modelo de referencia que permita expresar la relación de especialización. De hecho, en la arquitectura de metamodelado del OMG las relaciones entre dos capas de modelado sólo pueden expresarse como relaciones de *instancia*. En segundo lugar, el desarrollo dirigido por modelos se apoya en la definición de transformaciones para automatizar las tareas de generación de código. Esto exige que la información que guía la construcción del código debe ser representada como modelos. Sin embargo, la propuesta de los estándares HCE sitúa a los arquetipos a nivel de datos.

### 8.3. Ontologías y modelos

En el capítulo 5 se han propuesto varias representaciones de la arquitectura dual en OWL. El principio común en todas ellas ha sido la representación de las entidades del modelo de referencia como clases y propiedades. Sin embargo, la representación de los arquetipos varía en función de la tarea a la que están destinados. Así, por ejemplo, la definición de arquetipos como instancias de las clases del modelo de referencia propuesta en *Poseacle* es aplicada a la implementación de consultas semánticas [127] y, en cambio, la propuesta de *Archeck* basada en clases es el soporte de un método de validación. En este apartado se analizan los requisitos que ha de ofrecer la representación de arquetipos en OWL para ser integrada en una arquitectura de modelado.

El análisis de la especificación de la arquitectura dual en UML presentado en el apartado anterior concluye que es necesario que los arquetipos y el modelo de referencia estén situados en capas de modelado diferentes y contiguas. Las capas de modelado están vinculadas a través de la relación de instancia (relación de conformidad). Así pues, esta es la relación que debería soportar la semántica de especialización del modelo de arquetipos. Por otro lado, las transformaciones de modelos se expresan en el nivel M2 (metamodelos) y se ejecutan sobre modelos. Por consiguiente, si queremos que los arquetipos guíen el proceso de generación de código es necesario que sean modelos conformes a un metamodelo que exprese el modelo de referencia. Estos requisitos excluyen las representaciones de arquetipos basadas en clases debido a que sitúan arquetipos y modelo de referencia en la misma capa de modelado (nivel M1, modelos).

La especificación ODM (Ontology Definition Metamodel) del OMG propone un metamodelo para el lenguaje OWL (nivel M2). Situar el lenguaje OWL en el nivel de metamodelos permite que pueda relacionarse con otros metamodelos como UML a través de transformaciones. Así pues, las ontologías especificadas en OWL corresponden con modelos situados en el nivel M1 de la arquitectura de modelado. Por otro lado, el modelo de representación de conocimiento en el que se apoya el lenguaje OWL distingue dos tipos de conocimiento: el conocimiento terminológico (*TBox*) y el conocimiento asertivo o de instancias (*ABox*). Sin embargo, la propuesta ODM apenas establece esta separación de estos dos niveles de conocimiento de una ontología. La figura 8.4 muestra la organización de modelos consecuencia de una representación de arquetipos como instancias OWL en ODM. La ontología del modelo de referencia y las ontologías de los arquetipos son tratadas como modelos (nivel M1). Entre ellos se establece una asociación definida en el metamodelo de OWL. En definitiva, ODM no cumple el requisito de separar en dos niveles de modelado distintos el modelo de referencia y los arquetipos.

Por lo tanto, los estándares propuestos por el OMG no ofrecen soluciones que integren la arquitectura dual con la Ingeniería de Modelos. En la comunidad se ha trabajado en alternativas a los estándares del OMG que ofrezcan mayor flexibilidad para el modelado. El framework EMF incluye herramientas para promocionar distintos tipos de artefactos software a la categoría de metamodelos, tales como esquemas XML o interfaces en el lenguaje Java, aunque no ofrece soporte para ontologías OWL. Por otro lado, el entorno de edición de ontologías Protégé ha implementado un generador de código Java a partir de ontologías inspirado en la especificación ODM. La combinación de ambas herramientas permite transformar una ontología OWL en un metamodelo en el framework EMF. Esta solución ha sido adoptada en la herramienta Poseacle Converter para la implementación de una metodología de transformación de modelos clínicos. Las ontologías

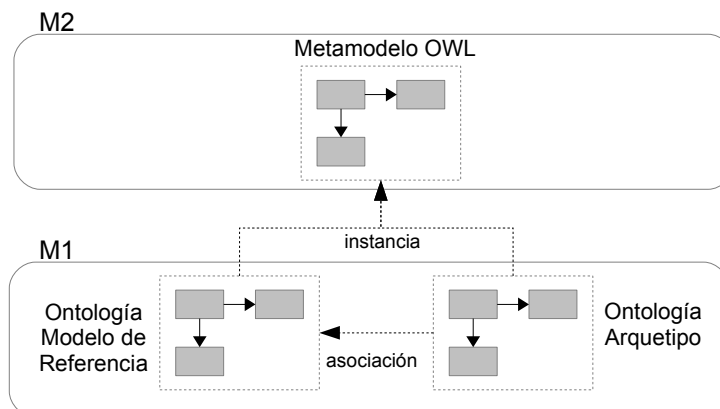


Figura 8.4: Arquitectura de modelado ontológico en ODM

*Poseacle* que representan a los modelos de referencia son convertidas en metamodelos EMF. Las instancias de la ontologías *Poseacle* son arquetipos conformes al modelo de referencia que se definen como modelos en EMF. Esta relación entre el modelo de referencia y arquetipos implementada en *Poseacle Converter* se muestra en la figura 8.5. Esta representación cumple los requisitos impuestos para integrar la arquitectura dual en un framework de modelado.

La tecnología OWL no sólo permite expresar de forma precisa la semántica del modelo de arquetipos y adaptarse a distintas tareas, sino que también sirve de puente para que la arquitectura dual de HCE aproveche los beneficios del desarrollo de software dirigido por modelos. La herramienta *Poseacle Converter* ha sacado partido de esta conexión para la implementación de un transformador de modelos clínicos y de extractos de HCE. En este capítulo también se explota la sinergia entre OWL y los modelos para la definición e implementación de un generador de aplicaciones sanitarias dirigido por arquetipos *semánticos*.

## 8.4. Modelos de arquetipos semánticos

El marco ontológico *Poseacle* propone integrar el modelo de referencia con las construcciones para instanciar arquetipos. Gracias a ello, la ontología para openEHR está formada por clases y propiedades que son utilizados para instanciar arquetipos, en vez de representar extractos de HCE para ese modelo de referencia.

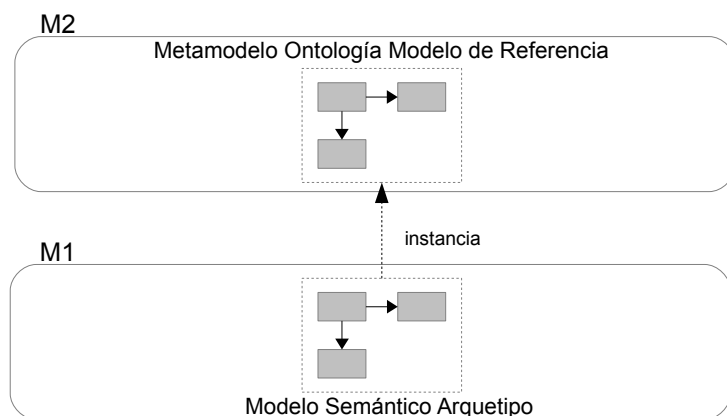


Figura 8.5: Arquitectura de modelado ontológico en *Poseacle Converter*

Un arquetipo *semántico* es el resultado de la interpretación semántica de los estándares de HCE de modelo dual, que es representado en el lenguaje OWL. La herramienta *Poseacle Converter* emplea arquetipos semánticos para la transformación de modelos clínicos entre estándares. Para la automatización del proceso de transformación y la definición de correspondencias entre los modelos de referencia, las ontologías han sido convertidas en metamodelos y los arquetipos en modelos.

Para la transformación de un arquetipo ADL en un modelo de arquetipo semántico es necesaria una arquitectura de transformación que ha sido implementada en el framework EMF. Los elementos centrales de la arquitectura son los metamodelos que expresan los arquetipos en su representación nativa (modelo de arquetipos, AOM) y como arquetipos semánticos (ontología *Poseacle*). En el apartado anterior se ha explicado el modo de obtener un metamodelo para la ontología *Poseacle* de un modelo de referencia combinando las herramientas Protégé y la importación de metamodelos de EMF. Asimismo, el framework EMF incluye un módulo que realiza la importación de esquemas XML que ha sido aplicado a la especificación de AOM como XML Schema para convertirlo en un metamodelo EMF.

La arquitectura de transformación de *Poseacle Converter* utiliza como entrada del proceso arquetipos en formato ADL. El hecho de disponer de un metamodelo para AOM no significa que los arquetipos ADL sean modelos de ese metamodelo. Esta cuestión ha sido abordada a nivel sintáctico. El lenguaje ADL no es el único formato en el que se pueden almacenar los arquetipos, ya que también puede utilizarse el esquema XML de AOM para guardar los arquetipos en formato XML. La conversión entre ambos

formatos es realizada por analizador de arquetipos proporcionado por la Fundación openEHR [29]. Gracias al cambio de sintaxis de ADL a XML, los arquetipos ya pueden ser tratados como modelos conformes al metamodelo de AOM en el framework EMF.

El metamodelo de la ontología *Poseacle* debe ser instanciado de acuerdo a las definiciones del arquetipo. Dado que los arquetipos pueden ser manejados como modelos, la propuesta de *Poseacle Converter* ha sido expresar la relación entre AOM y la ontología *Poseacle* como una transformación de modelos especificada con reglas en el lenguaje RubyTL. La ejecución de la transformación daría como resultado la obtención de un modelo del arquetipo semántico.

El proceso de obtención de arquetipos semánticos a partir de arquetipos en ADL puede resumirse en los siguientes pasos:

- Conversión ADL a XML. Un arquetipo en formato ADL es convertido al formato XML conforme al esquema XML de AOM. El resultado es un modelo que es instancia del metamodelo de AOM.
- Transformación del modelo AOM en un arquetipo *semántico*. La aplicación de las reglas de transformación especificadas en el lenguaje RubyTL permite instanciar el metamodelo de la ontología *Poseacle* de acuerdo a las definiciones del arquetipo.

## 8.5. Semántica de visualización de arquetipos

Los arquetipos describen qué información clínica debe ser capturada en los sistemas sanitarios y qué validaciones deben aplicarse para garantizar que la información sea correcta. En este sentido, un arquetipo se asemeja a un formulario clínico, y es esta característica la que permite utilizar arquetipos como modelo de inicial en el desarrollo de un sistema de información. Sin embargo, un arquetipo no es la representación más adecuada para expresar su significado en un sistema de información. Es por ello que resulta necesario disponer de un modo neutral de expresar esta semántica con independencia de cualquier tecnología empleada en la interfaz de usuario.

El lenguaje XForms es una recomendación del W3C para la especificación de interfaces gráficas [192]. La característica que hace especialmente interesante a esta tecnología es que la interfaces definidas con el lenguaje, denominadas formularios XForms, son independientes de cualquier dispositivo y tecnología de presentación, es decir, el lenguaje XForms proporciona una representación genérica de las interfaces de usuario. Además, el principio de diseño del lenguaje impone la separación de la lógica de presentación y la lógica de datos. Los componentes de la arquitectura de presentación gráfica propuesta en XForms son los siguientes:

- **Controles gráficos.** El lenguaje XForms ofrece un amplio abanico de controles para la entrada de datos, y en menor medida, para la visualización de la información.
- **Esquema de datos.** La información que es capturada y presentada en los controles XForms es almacenada de acuerdo a un esquema de datos expresado en XML.
- **Enlaces.** Estos componentes de la arquitectura están a cargo de establecer la conexión entre los controles gráficos y el esquema de datos. Asimismo, los enlaces también son utilizados para especificar las restricciones de validación de los datos.

Por tanto, los formularios XForms expresan de forma más adecuada que los arquetipos la semántica de visualización y entrada de la información clínica, manteniendo la independencia tecnológica. Así pues, el reto es trasladar esta semántica de la arquitectura dual a formularios XForms. Sin embargo, esta tarea no depende sólo del modelo de arquetipos, sino que también depende en gran medida del modelo de referencia. Los requisitos de información expresados en las estructuras de datos determinan los controles gráficos necesarios para capturar y visualizar la información.

El análisis de la visualización de las estructuras de datos y cómo las restricciones de los arquetipos influyen en su presentación es el aspecto más importante en la traducción de la semántica de la arquitectura dual a XForms. A continuación se muestra el ejemplo del tipo de datos `DV_INTEGER` de ISO EN 13606, que almacena valores enteros, para el que se analiza su presentación gráfica en función de varias restricciones. La representación esquemática de cada una de ellas se muestra en la figura 8.6.

- **Dato de tipo entero de entrada obligatoria.** El campo es representado con un control de tipo `input` que incluye la etiqueta visual y un mensaje de ayuda. Las restricciones de validación del control son expresadas en una declaración `bind`. En el ejemplo se especifica el tipo de datos (`type="xs:integer"`) y se indica que el campo es obligatorio (`required="true"`)

---

```
<input id="campo1" bind="campo1_bind">
  <label>Valor Entero:</label>
  <hint>Mensaje de Ayuda</hint>
</input>

<bind id="campo1_bind" type="xs:integer" required="true" />
```

---

- **Dato de tipo entero restringido por un rango de valores.** El control de tipo `range` se encarga de validar que el dato introducido se encuentre en el rango especificado (atributos `start` y `stop`). Asimismo permite

especificar la distribución de valores permitidos en ese rango. En el ejemplo, el atributo `step="10"` indica que se aceptan enteros del 0 al 100 en incrementos de 10 (0, 10, 20, etc.).

---

```
<range id="campo1" start="0" end="100" step="10" >
  <label>Rango Enteros:</label>
</range>
```

---

- Dato de tipo entero restringido a una enumeración de valores. El control de tipo `select1` permite restringir un campo mediante una lista de valores fijos.

---

```
<select1 id="campo1" type="close" appearance="minimal">
<label>Lista Enteros:</label>
  <item>
    <label>23</label>
    <value>23</value>
  </item>
  <item>
    <Label>42</label>
    <value>42</value>
  </item>
  ...
</select1>
```

---

El diagrama muestra tres ejemplos de interfaces de usuario para datos enteros:

- Valor entero:** Un campo de texto simple con el valor "0" ingresado.
- Rango enteros:** Un campo de texto con el valor "0" y un control de navegación (flechas arriba y abajo) a la derecha.
- Lista enteros:** Un campo de texto con un control de lista desplegable (una flecha hacia abajo) a la derecha.

Figura 8.6: Esquema de visualización del tipo de datos entero

La representación gráfica de las entidades de negocio del modelo es elegida atendiendo a las necesidades de captura de datos. Estos requisitos dependen tanto de la estructura de datos como de las restricciones aplicables según el modelo de arquetipos. La propuesta para ISO EN 13606 ha sido organizar la presentación en formularios que corresponden con las entradas (ENTRY) de los arquetipos. Las estructuras de datos organizativas (COMPOSITION, FOLDER y SECTION) son utilizadas para organizar los formularios de la aplicación en un *árbol de navegación*.



La semántica de la presentación visual de los arquetipos se especifica como transformaciones de modelos de arquetipos semánticos a formularios XForms. El lenguaje XForms puede ser traducido a un metamodelo EMF gracias a que se ha especificado como un esquema XML. Lo mismo ocurre con el modelo de arquetipos que también está definido en XML (véase apartado 8.4). En general, las correspondencias entre dos metamodelos se expresan utilizando un lenguaje de transformación modelo a modelo como RubyTL. Sin embargo, el tratamiento que hace EMF de los modelos que han sido importados en XML ofrece la opción de utilizar un lenguaje de transformación modelo a texto. El motivo es que el framework EMF acepta como modelos los documentos XML de aquellos metamodelos importados a partir de un esquema XML. Por tanto, se ha preferido la transformación modelo a texto en el lenguaje MOFScript que permite definir las correspondencias como *plantillas*, lo que facilita el desarrollo y el mantenimiento de la transformación.

La herramienta ArchForms implementa la conversión de arquetipos semánticos en formularios XForms. ArchForms es un generador de aplicaciones sanitarias construido en base a técnicas de Ingeniería de Modelos. La introducción de modelos *abstractos* de interfaz de usuario (formularios XForms) dentro en el proceso de generación de aplicaciones tiene fundamentalmente dos ventajas. En primer lugar, el conocimiento sobre el diseño de las interfaces de usuario se deja en manos de expertos en este área y se proporciona un método intuitivo para especificarlas (plantillas MOFScript). En segundo lugar, gracias a los formularios XForms se consigue la independencia tecnológica que puede ser aprovechada en un proceso de desarrollo dirigido por modelos.

## 8.6. ArchForms

ArchForms es un generador de aplicaciones de HCE para el estándar ISO EN 13606. La arquitectura de las aplicaciones ArchForms utiliza la tecnología JSF de Java [64] que promueve la organización del código conforme al patrón Modelo/Vista/Controlador. JSF no es una biblioteca de componentes, sino una especificación que dicta cómo han de construirse las interfaces gráficas para entornos web y cómo se relacionan éstas con los modelos de información. Existen numerosas bibliotecas de componentes gráficos que cumplen con la especificación [76], como por ejemplo RichFaces del framework Seam, PrimeFaces o TouchFaces que está especializada en interfaces gráficas para dispositivos móviles con interacción táctil.

La herramienta ArchForms ha sido diseñada utilizando una arquitectura generativa basada en técnicas de Ingeniería de Modelos. La característica más destacada de la arquitectura es la independencia de la tecnología de implementación y de los estándares de HCE. En esta sección se presen-

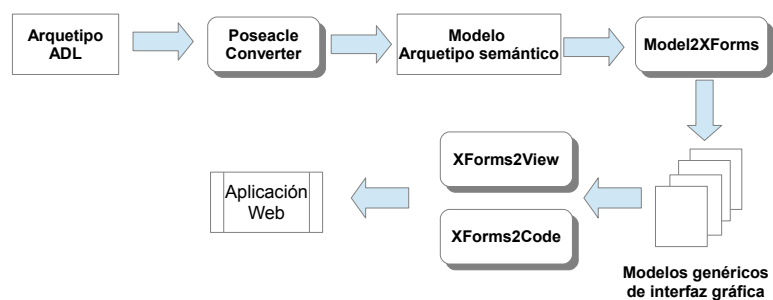


Figura 8.7: Proceso de generación de aplicaciones en ArchForms

ta la arquitectura de la herramienta y su extensibilidad. Seguidamente, se desarrolla un ejemplo de aplicación ArchForms y se muestra cómo gestiona extractos normalizados de HCE conforme al estándar ISO EN 13606. Finalmente, se analiza el papel de los componentes de la arquitectura ArchForms en el marco de metamodelado de MDA.

### 8.6.1. Arquitectura

El proceso de generación propuesto en ArchForms está organizado en tres fases (véase figura 8.7). La entrada del generador es un arquetipo en formato ADL. La primera fase consiste en transformar ese arquetipo en un *arquetipo semántico* expresado como un modelo. Esta tarea es ejecutada por la herramienta *Poseacle Converter*. A continuación, la semántica de visualización definida en los arquetipos semánticos es traducida a modelos genéricos de interfaz gráfica expresados en XForms. Esta tarea se implementa en un módulo de transformación en MOFScript (*Model2XForms*). La última fase de la generación consiste en construir una aplicación web que cumpla la especificación recogida en los formularios XForms. Esta fase está definida en dos transformadores MOFScript: *XForms2View* y *XForms2Code* (véase figura 8.8). El primero se encarga de procesar los modelos XForms para generar las vistas de la aplicación web. En cambio, el transformador *XForms2Code* está a cargo de la obtención del código de soporte de la arquitectura de la aplicación como controladores, validaciones, etc.

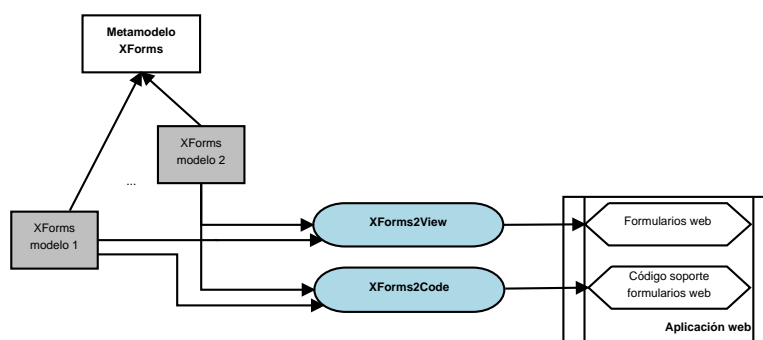


Figura 8.8: Módulos generadores de ArchForms

La característica más destacada del generador ArchForms es su extensibilidad. Los transformadores utilizados en la última fase de proceso de generación (*XForms2View* y *XForms2Code*) están parcialmente definidos, o dicho de otro modo, son transformadores *abstractos*. Estos transformadores implementan los requisitos de la arquitectura, dejando en manos de otros módulos los detalles concretos de la tecnología. Actualmente el generador incorpora tres módulos: *CodeGen\_Seam* que utiliza la biblioteca de componentes gráficos RichFaces y el framework Seam, *CodeGen\_Prime* que emplea la biblioteca PrimeFaces y *CodeGen\_Touch* que utiliza TouchFaces para generar interfaces optimizadas para los navegadores web de dispositivos móviles. La figura 8.9 muestra la relación entre los transformadores abstractos y los módulos específicos de la plataforma tecnológica.

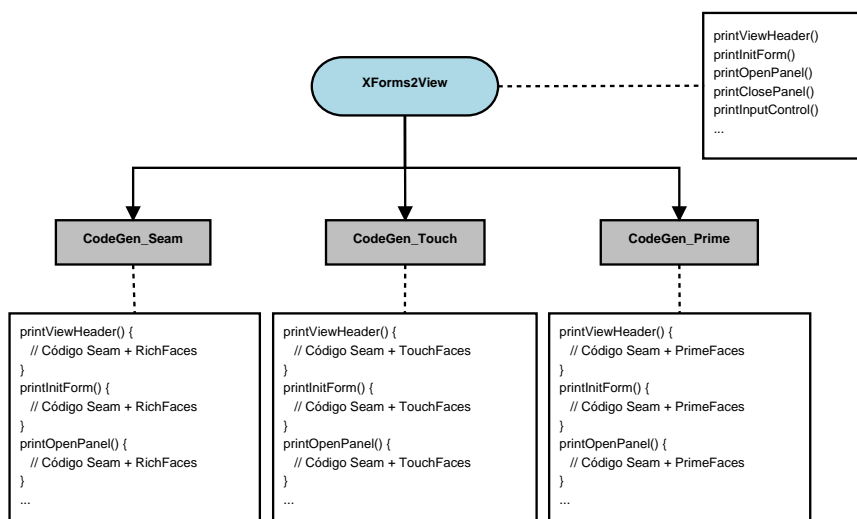


Figura 8.9: Lógica de generación del transformador XForms2View

Por último, la arquitectura ha sido desarrollada de acuerdo a los principios de desarrollo de software dirigido por modelos propuestos en [180]. En primer lugar, se diseñó e implementó un prototipo de aplicación JSF sobre el framework Seam, a partir del cual se obtuvieron los requisitos arquitectónicos de las aplicaciones ArchForms. En la siguiente etapa, se definieron e implementaron las correspondencias conceptuales entre el modelo semántico de arquetipos y la especificación XForms, y entre ésta y la biblioteca de componentes RichFaces. La extensibilidad de la herramienta se puso a prueba con la introducción de nuevos requisitos de interfaz para aplicaciones móviles (TouchFaces). En esta fase se identificaron las características propias de la arquitectura software, que fueron situadas en los transformadores abstractos, y las características específicas de las plataformas (módulos *CodeGen*). Finalmente, se validó la extensibilidad de la arquitectura introduciendo una nueva biblioteca de componentes gráficos (PrimeFaces).

### 8.6.2. Evolución

ArchForms ha sido diseñado como un generador extensible que puede ser adaptado a otras arquitecturas HCE basadas en el modelo dual y a cualquier plataforma tecnológica. Esta característica se consigue gracias a que la arquitectura del generador se organiza en varias fases que favorecen la evolución de la herramienta. En este apartado se hace un análisis del impacto de la adaptación a otros estándares y tecnologías de interfaz de usuario.

Actualmente el generador ArchForms ha sido implementado para el estándar ISO EN 13606. La adaptación a cualquier otro estándar basado en la arquitectura de modelo dual, como por ejemplo openEHR, requiere las siguientes tareas:

- Disponer de los *arquetipos semánticos* para openEHR. Los arquetipos semánticos son obtenidos con la herramienta Poseacle Converter que actualmente soporta arquetipos de los estándares ISO EN 13606 y openEHR
- Transformador *Model2XForms*. Las correspondencias entre arquetipos y formularios XForms dependen en gran medida de las estructuras y tipos de datos del modelo de referencia. Esto significa que el transformador *Model2XForms* no podría ser reutilizado completamente. No obstante, no sería costoso adaptarlo a la especificación openEHR, ya que el modelo de referencia de ISO EN 13606 puede considerarse un subconjunto de openEHR. Otra alternativa podría ser transformar el arquetipo openEHR a ISO EN 13606 utilizando Poseacle Converter y reutilizar el transformador *Model2XForms*.

- Transformador *XForms2Code*. Este transformador debería ser implementado en su mayor parte, ya que está a cargo de generar las clases de apoyo a las interfaces de usuario, que en este caso han de corresponder con las entidades de openEHR.
- Modelo de datos de openEHR. La versión actual de generador utiliza JAXB [184] gestionar los extractos de HCE del estándar ISO EN 13606. La misma estrategia se podría emplear para manejar extractos conformes con openEHR.

Nótese que el resto de módulos de la arquitectura generativa podrían ser reutilizados. Por ejemplo, el módulo encargado de la generación específica para la tecnología Seam (*CodeGen\_Seam*) sigue siendo válido ya que está definido en base a las definiciones genéricas de las interfaces XForms.

La versión actual de ArchForms genera aplicaciones para tres tecnologías basadas en JSF (RichFaces, TouchFaces y PrimeFaces). La introducción de nueva biblioteca de componentes gráficos JSF, como por ejemplo IceFaces [76], no afectaría a la arquitectura del generador. La única tarea necesaria sería implementar un módulo *CodeGen* para esa tecnología.

Los cambios menores en la arquitectura de las aplicaciones ArchForms implicarían cambios en los transformadores *XForms2View* y/o *XForms2Code*. Por ejemplo, la gestión del modelo de datos podría ser implementada con EMF en vez de JAXB. Este cambio sólo tendría impacto en el módulo *XForms2Code*.

Por último, la evolución tecnológica a otra plataforma como Microsoft .NET o Adobe Flash obligaría a sustituir los módulos de la tercera fase del proceso de transformación, esto es, los transformadores *XForms2View* y *XForms2Code*. Por lo tanto, la dos primeras fases del proceso ArchForms no se verían afectadas por los cambios en la tecnología. En cualquier caso, actualmente ArchForms implementa transformadores *abstractos* basados en patrones software que que podrían ser aprovechadas en otras tecnologías como .NET.

### 8.6.3. Ejemplo de aplicación

En este apartado se ilustra el proceso de obtención de una aplicación ArchForms a partir del arquetipo *Revisión obstetricia* en ISO EN 13606. El arquetipo está definido como una COMPOSITION formada por dos secciones que capturan los resultados de dos exámenes (general y abdominal). A su vez, el examen abdominal se divide en dos entradas clínicas (ENTRY). La figura 8.10 muestra un extracto de la definición del arquetipo que corresponde con la *ritmo cardíaco del feto* (at0005).

```

definition
  -- Revisión obstetricia
  COMPOSITION [at0001] matches {
    content cardinality matches {2..2; ordered; unique} matches {
      ...
      -- Examen abdominal
      SECTION [at0003] occurrences matches {1..1} matches {
        members cardinality matches {1..3; ordered; unique} matches {
          -- Frecuencia cardíaca del feto
          ENTRY [at0004] occurrences matches {1..1} matches {
            items cardinality matches {1..3; ordered; unique} matches {
              -- Ritmo cardíaco
              ELEMENT [at0005] occurrences matches {0..1} matches {
                value matches {
                  PQ [at0006] occurrences matches {1..1} matches {
                    value matches {|120..180|}
                    units matches {
                      CV [at0007] matches {
                        displayName matches {"Latidos por minuto"}
                        codeValue matches {"{H.B.}/min"}
                        codingSchemeName matches {"UCUM"}}
                      property matches {
                        CD [at0008] matches {
                          -- Latidos por minuto del feto
                          codeValue matches {"249043002"}
                          codingSchemeName matches {"SNOMED-CT"}}
                      }}}
                }}}
            ...
          }
        }
      }
    }
  }

```

Figura 8.10: Extracto de la definición del arquetipo *Revisión obstetricia* del estándar ISO EN 13606

Una vez obtenido el modelo semántico del arquetipo con *Posecle Converter*, el proceso de transformación genera un formulario XForms por cada entrada del arquetipo (ENTRY). Así pues, habrá un formulario XForm asociado a la entrada *Frecuencia cardíaca del feto* (at0004). La figura 8.11 muestra un extracto del formulario XForms obtenido que contiene los controles gráficos asociados a los términos at0005 y at0006. Este término captura la información sobre ritmo cardíaco (véase figura 8.10) y se define con una estructura de datos PQ (cantidad física) restringida en el término at0006: el atributo value debe tomar valores en el intervalo comprendido entre 120 y 180, el atributo units indica que la unidad de medida es “Latidos por minuto” según la codificación UCUM, y el atributo property fija la entidad clínica que está siendo medida, esto es, el concepto “Latidos por minuto del feto” en la terminología SNOMED-CT.

```
<group id="at0005" bind="at0005_bind">
  <hint id="at0005_hint">
    <output value="Frecuencia cardíaca del feto"/>
  </hint>
  <label><output value="Ritmo cardíaco"/></label>
  <group id="at0006" bind="at0006_bind" >
    <range id="at0006_value" bind="at0006_value_bind" start="120" end="
      180" step="1">
      <label><output value="Magnitud"/></label>
    </range>
    <input id="at0006_units" bind="at0006_units_bind">
      <label><output value="Unidades"/></label>
    </input>
  </group>
</group>
```

Figura 8.11: Extracto de los controles XForms que capturan la frecuencia cardíaca del feto

Los términos del arquetipo son declarados dentro de un grupo que puede tener asociada información de ayuda contextual (*hint*) obtenida de la sección *ontología* del arquetipo. En concreto, el término *at0005* está enlazado a la descripción textual “Frecuencia cardíaca del feto”. La información del atributo *value* del término *at0006* es capturada con un control de tipo *range* cuyas reglas de validación están definidas en el enlace *at0006\_value\_bind*. El atributo *units* se almacena con un código (*cv*) que tiene todos los atributos restringidos con un valor fijo. Por lo tanto, se presenta en modo *sólo lectura* en el que se visualiza la restricción del atributo *displayName*. El tercer atributo no tiene presentación visual, ya que su papel es definir el significado clínico de la información capturada.

Los controles de un formulario están enlazados con el esquema de datos a través del atributo *bind*. El modelo de datos de un formulario XForms declara la estructura de datos en la que se almacena la información y los enlaces con las restricciones de validación. La figura 8.12 muestra el enlace del atributo *value* del término *at0006* en el que se establecen las restricciones: (1) es obligatorio establecer un dato (*required="true"*), (2) el dato introducido debe ser un entero (*type="xs:integer"*) y (3) el intervalo de valores permitido (*constraint="interval=120..180"*). Finalmente, el atributo *nodeset* establece la ruta XPath del elemento del modelo que corresponde con la declaración del control gráfico.

```

<model>
  ...
  <bind id="at0006_value_bind" nodeset="data/at0004/items/at0005/value/
    at0006/value"
    required="true" type="xs:integer" constraint="interval=120..180"/>
  ...
</model>

```

Figura 8.12: Enlace que declara las restricciones de un control en XForms

El formulario completo asociado a la entrada *Frecuencia cardíaca* se muestra en la parte derecha de la figura 8.13. El árbol de navegación del arquetipo al que pertenece se sitúa a la izquierda de la página web, donde se puede apreciar que el arquetipo se organiza en dos secciones *Examen General* y *Examen Abdominal*, y la *Frecuencia cardíaca* forma parte del *Examen Abdominal*. Nótese cómo los requisitos de visualización de una aplicación para navegadores de escritorio son diferentes a los requisitos para dispositivos móviles (véase figura 8.14).

Figura 8.13: Formulario de la aplicación Revisión obstetricia





Figura 8.14: Aplicación Revisión obstetricia en iPhone

#### 8.6.4. Historia clínica normalizada

Las aplicaciones ArchForms gestionan la HCE conforme al modelo de referencia de la norma ISO 13606. Los formularios en los que está organizada la presentación del arquetipo permiten ir almacenando parcialmente la información introducida (véase figura 8.13). Una vez finalizada la introducción de los datos clínicos, el usuario puede solicitar la generación y almacenamiento de la información como un extracto de HCE. La vista encargada de la gestión de extractos se muestra en la figura 8.15.

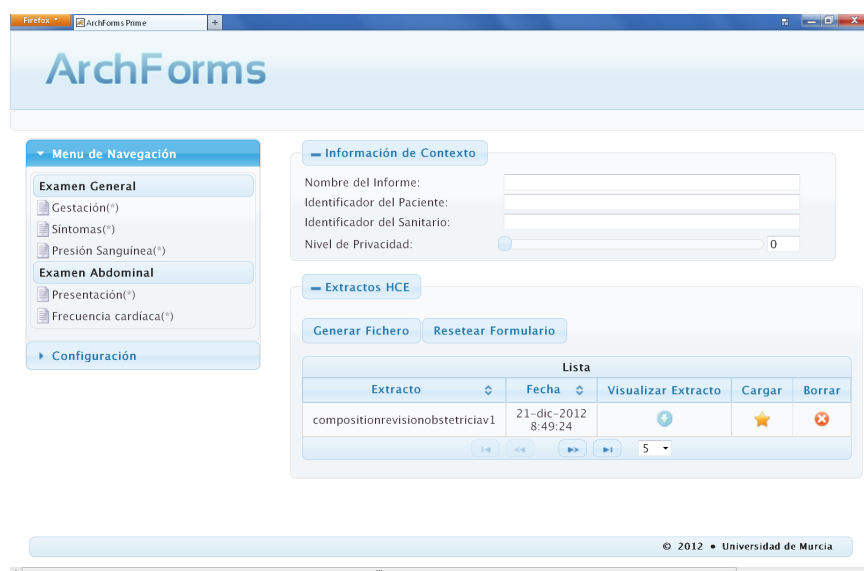


Figura 8.15: Gestor de extractos de la aplicación Revisión Obstetricia

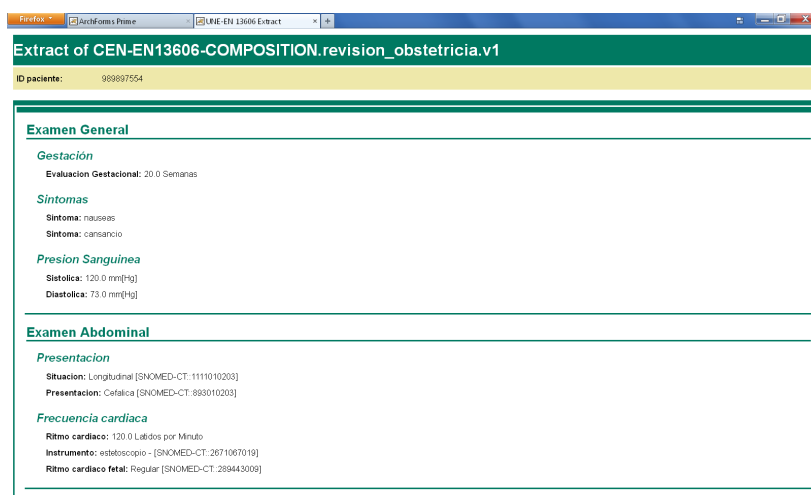


Figura 8.16: Representación gráfica de un extracto HCE de la aplicación Revisión Obstetricia

El gestor de extractos permite descargar un extracto, para poder enviarlo a otro sistema de información, o recuperarlo en la aplicación para ser editado. Finalmente los extractos pueden ser visualizados en una página, tal como muestra la figura 8.16.

### 8.6.5. ArchForms en la arquitectura MDA

ArchForms es la demostración de la aplicación de los principios de la Ingeniería de Modelos a la arquitectura dual de HCE. A lo largo del capítulo se ha estudiado el modo de integrar los niveles de modelo de la arquitectura dual el marco de metamodelado propuesto en MDA. En primer lugar, la representación de las ontologías *Poseacle* como metamodelos ha permitido la separación del modelo de referencia y arquetipos en dos capas diferentes de modelado. Asimismo, la relación de instancia establecida entre modelos y metamodelos soporta la semántica de especialización del modelo de arquetipos. Los arquetipos con esta representación son denominados modelos de *arquetipos semánticos*. La presentación visual de los arquetipos se especifica en formularios XForms obtenidos a partir de los arquetipos semánticos. Finalmente, se han implementado varios módulos de generan las aplicaciones web a partir de los modelos anteriores.

A continuación se analiza qué papel tiene cada uno de los componentes de la arquitectura ArchForms en MDA:

- Ontologías *Poseacle*. Son representadas como metamodelos y se sitúan, por tanto, en la capa M2 de la arquitectura de metamodelado.

- Arquetipos semánticos. Son considerados modelos de dominio que expresan los requisitos de información de las aplicaciones sanitarias sin especificar ningún detalle acerca de su implementación. Dentro de las categorías de modelos en MDA podrían ser considerados *modelos independientes de la computación* (CIM). En la arquitectura de modelado están en el nivel M1 (modelos) y son conformes con el metamodelo de una ontología *Poseacle*.
- Formularios XForms. Representan la especificación neutral y genérica de las interfaces de usuario de una aplicación. En este sentido podrían ser considerados modelos independientes de la plataforma (PIM). Los formularios XForms son gestionados como modelos conformes al metamodelo del lenguaje XForm.
- Transformación *Model2XForm*. Corresponde con la transformación de CIM a PIM en la arquitectura MDA. Esta transformación encapsula el conocimiento de presentación visual de la arquitectura de dual para un estándar concreto.
- Transformadores *XForms2View* y *XForms2Code*. En ArchForms el concepto de modelo específico de la plataforma (PSM) se omite en favor de la definición de transformaciones *abstractas*. Las características generales de la plataforma son definidas en estos transformadores, dejando los detalles específicos de la generación de código a otros módulos.
- Módulos *CodeGen*. Son las piezas que completan los transformadores específicos de la plataforma. En sí mismos no definen correspondencias, sino que establecen el código asociado a una tecnología software específica.

En resumen, la arquitectura generativa de ArchForms integra la arquitectura dual en la Ingeniería de Modelos. El principal objetivo de la herramienta es favorecer la introducción de sistemas basados en estándares en el ámbito sanitario, y, en consecuencia, lograr la interoperabilidad de la información clínica. Así mismo, la herramienta pretende promover los principios del modelado dual, es decir, que los expertos en el dominio clínico sean los responsables de la gestión de los conceptos clínicos (arquetipos), y que a partir de ellos se obtengan aplicaciones funcionales sin tener que ocuparse de los detalles técnicos de implementación. Finalmente, la aplicación de técnicas de Ingeniería de Modelos consigue reducir los costes de la evolución tecnológica de las aplicaciones sanitarias.



## Capítulo 9

# Conclusiones y Trabajo Futuro

### 9.1. Resumen y conclusiones

Algunas arquitecturas avanzadas de HCE y estándares están basados en la arquitectura de modelo dual, que representa el conocimiento clínico por medio de arquetipos. Un aspecto clave para el éxito de estas arquitecturas es la disponibilidad de arquetipos de calidad. Hasta ahora a pesar de la identificación de la necesidad de criterios de calidad y buenas prácticas para su diseño, la disponibilidad de métodos que midan la calidad de arquetipos es realmente limitada [103]. Por calidad de arquetipos nos referimos a arquetipos que sean correctos, consistentes, precisos y usables desde las perspectivas clínica, técnica, estructural y semántica, es decir, no nos referimos simplemente a tener un arquetipo que sea correcto de acuerdo con un estándar en concreto, que es el mínimo requisito para un arquetipo. Sin embargo, esta cuestión básica no es tratada de una forma adecuada por las herramientas de edición. El diseño de los arquetipos es una tarea compleja, por lo que es necesario disponer de métodos que guíen a los usuarios en su correcta construcción, como por ejemplo, métodos que garanticen que los arquetipos especializados son consistentes con sus ancestros.

En el capítulo 6 se presenta el método *Archeck* que hace uso de tecnologías OWL para el chequeo de la consistencia de arquetipos. El método propuesto está centrado fundamentalmente en la consistencia de las estructuras de datos y en la evaluación de los enlaces terminológicos en las especializaciones de arquetipos. El primero es un requisito técnico de los arquetipos mientras que el segundo es un indicador de calidad del alineamiento entre arquetipos y terminologías. La representación de arquetipos, el modelo de referencia y las terminologías en OWL es la base para definir e implementar métricas de calidad tales como la validación de enlaces terminológicos en especializaciones.

La propuesta presentada en *Archeck* parte de la idea de que las tecnologías OWL pueden ser útiles no sólo para comprobar la consistencia de los arquetipos, sino que además permiten que el desarrollo y mantenimiento de los métodos de validación sea más sencillo. La comunidad OWL está haciendo grandes esfuerzos en el desarrollo de herramientas de razonamiento efectivas y eficientes, las cuales proporcionan un gran valor para esta tarea.

El editor LinkEHR-Ed [123] define e implementa un método de validación de los arquetipos. Su solución se basa en un formalismo algebraico que define un sistema de tipos basado en árboles con nodos etiquetados y en la definición e implementación de una función de subsunción que representa la relación de conformidad en la relación de especialización. Esta propuesta requiere transformar y mezclar la arquitectura dual en este formalismo. El modelo de referencia es derivado como *arquetipos de negocio* de modo que la función de subsunción pueda ser aplicada para la edición de un arquetipo base o para la creación de una especialización. Por tanto, la primera diferencia con la propuesta LinkEHR-Ed es que no hemos desarrollado un formalismo propio, sino que utilizamos el que proporciona OWL. En concreto, hemos utilizado OWL-DL, que es la variante basada en lógica descriptiva. El proceso de validación propuesto depende en gran medida de la capacidad de inferencia de los razonadores, especialmente para encontrar relaciones de subsunción y clases inconsistentes. Por tanto, el objetivo ha sido analizar el mejor modo de mantener la semántica de los arquetipos en las tecnologías OWL.

El método propuesto por LinkEHR-Ed tiene buenas propiedades en términos de *genericidad*, lo que permite que pueda ser aplicado a cualquier modelo de referencia de una arquitectura de modelo dual. Sin embargo, el esfuerzo necesario para implementar este método es alto, ya que se ha tenido que definir el formalismo algebraico, proporcionar representaciones de arquetipos y del modelo de referencia, y diseñar e implementar la función de subsunción. En caso de que surjan nuevos requisitos, como comprobar la consistencia de los enlaces terminológicos, algunos componentes podrían necesitar cambios. En cuanto a *Archeck*, el método de validación puede ser aplicado a cualquier arquitectura HCE basada en el modelo dual. La representación OWL del modelo de referencia es generada automáticamente a partir de un modelo UML o esquema XML de acuerdo a las reglas especificadas en ODM [149]. Por tanto, el método puede ser de interés para iniciativas como CIMI, donde cabe esperar que usen varios modelos de información. Además, otros tipos de modelos clínicos son construidos restringiendo un modelo de referencia. CEM es un ejemplo al que podría aplicarse un enfoque similar para la validación de los modelos clínicos usando tecnologías OWL. Es más, extender la propuesta *Archeck* a CEM sería de interés dado el elevado número de modelos CEM disponibles.

El método *Archeck* ha sido diseñado con el objetivo de aprovechar la definición en OWL de la semántica de la arquitectura basada en el modelo dual para la validación de arquetipos. No obstante, la propuesta también proporciona una solución natural para tareas semánticas como el chequeo de la consistencia de enlaces terminológicos en las relaciones de especialización. Los enlaces terminológicos pueden ser expresados como axiomas en OWL siempre que la terminología también esté disponible en formato OWL, como es el caso de SNOMED-CT. El modelo de arquetipos no impone que los enlaces con las terminologías sean consistentes a nivel clínico. Por tanto, el análisis de esta condición puede ser considerado como una métrica que evalúa la calidad de los arquetipos.

La terminología SNOMED-CT está definida formalmente en una lógica descriptiva con expresividad  $\mathcal{EL}++$  que permite ser representada en OWL [177]. Actualmente esta terminología incluye más de 300.000 conceptos, lo que hace que la tarea de razonamiento con la base de conocimiento consuma un tiempo excesivo. Algunos razonadores como ELK [105] son capaces de clasificar la ontología de SNOMED-CT en segundos, pero no pueden ser utilizados en el método *Archeck* porque su expresividad no es suficiente para los requisitos de validación de arquetipos. Fact++ [190] es el razonador de propósito general más eficiente, aunque sólo es capaz de clasificar la terminología en minutos. Por tanto, actualmente no es práctico utilizar la ontología completa de SNOMED-CT en tareas de razonamiento. La solución para este problema consiste en utilizar el conjunto mínimo de conceptos SNOMED-CT para el chequeo de la corrección del arquetipo. El trabajo con subconjuntos de SNOMED-CT ha sido recomendado en informes internacionales [183] y actualmente es un tema activo de investigación [32, 115, 131]. Además, en el contexto de la validación de documentos clínicos, el uso de subconjuntos de SNOMED-CT también ha sido propuesto para la validación de documentos HL7 CDA [74].

El análisis de la consistencia de los enlaces terminológicos utilizando tecnologías OWL ha sido abordado en otros estándares de HCE. En HL7 se han propuesto unas guías sobre el uso de la terminología SNOMED-CT en los documentos CDA [113]. En [74] se ha definido una ontología OWL para CDA que representa los documentos como instancias de las clases de la ontología. Las guías son definidas como restricciones de integridad e implementadas como axiomas en OWL. La evaluación de las guías hace uso de un razonador OWL que aplica inferencias bajo la asunción de *mundo cerrado*. La diferencia con la propuesta *Archeck* es que el análisis de la consistencia de los enlaces terminológicos se realiza desde la perspectiva de la relación de subsunción. Es decir, se adopta el supuesto de que un término de un arquetipo especializado sólo puede establecer un enlace terminológico que sea *compatible* con el enlace establecido en el arquetipo padre (relación *is a* de SNOMED-CT).

En la literatura, la representación OWL de arquetipos ha sido abordada de dos modos, utilizando una aproximación basada en instancias o basada en clases. La primera es soportada por una ontología que define los conceptos de las arquitecturas de modelo dual. En esta propuesta, los arquetipos son representados como individuos (instancias) de los conceptos de esa ontología. Los razonadores OWL son capaces de comprobar la consistencia de las aserciones que definen a los individuos en relación al conocimiento estructural definido en la base de conocimiento. Por ejemplo, una clase del modelo de referencia puede declarar una propiedad con un axioma de cardinalidad que restringe el número de instancias. En este caso, una instancia de un arquetipo que viole la cardinalidad de esa asociación podría ser detectada como inconsistente. Nuestro grupo de investigación ha encontrado que esta representación es útil para implementar mecanismos de búsqueda y clasificación de arquetipos basados en tecnologías de la Web Semántica [127], así como para abordar el reto de la interoperabilidad de la HCE [128, 129]. Sin embargo, la conformidad de la relación de especialización no puede ser validada fácilmente, ni la consistencia clínica de las definiciones en función de los enlaces terminológicos puede ser implementada.

En algunos dominios de aplicación también es necesario representar los extractos HCE en OWL [109, 118]. Este requisito impone que los arquetipos sean especificados como clases para que los extractos de HCE sean definidos como individuos de tales clases. Esta propuesta también permite la validación de las definiciones de los arquetipos, pero el alcance de la validación depende en gran medida de los axiomas utilizados para la definición de las clases. En concreto, las propuestas citadas anteriormente hacen uso de axiomas de inclusión, que no son apropiados para comprobar la consistencia de las declaraciones de conceptos arquetipados tal como se explicó en la sección 5.4.2.3 y que, además, no definen completamente la semántica del modelo de arquetipos.

El método de validación de arquetipos propuesto en esta tesis se apoya en las capacidades de inferencia de los razonadores OWL, especialmente en la obtención de relaciones de inclusión y la detección de clases inconsistentes. Sin embargo, la expresividad del lenguaje OWL ha sido el principal inconveniente para representar la semántica del modelo de arquetipos. Por ejemplo, las propuestas de [109, 118] no soportan la definición de restricciones sobre tipos de datos primitivos, algo esencial en la definición de los arquetipos, debido a que cuando fueron definidas OWL no ofrecía estas construcciones. En 2009 se publicó OWL 2 que, entre otras mejoras, aumentó la expresividad del dialecto DL basado en lógica descriptiva e incorporó construcciones para restringir el dominio de los tipos primitivos. Sin embargo, la mayoría de los razonadores aún no ofrecen soporte para este tipo de restricciones, siendo el razonador Hermit [175] una excepción.



OWL no ofrece soporte nativo para la definición del orden de atributos multivaluados, lo que motiva la necesidad de aplicar diferentes patrones de modelado. En [46] se propone una solución para definir secuencias en OWL que es diferente a la implementada en el método de validación. El método *Archeck* propone el uso de dos clases disjuntas (`ORDERED` y `UNORDERED`) en combinación con el axioma de cuantificación universal (*only*) que establece la restricción asociada al atributo. En cambio, la propuesta de secuencias en OWL consiste en la definición de clases anónimas anidadas para representar la relación de orden cuya complejidad está en función del número de componentes del atributo, lo que perjudica el rendimiento de la tarea de razonamiento. La solución propuesta en *Archeck* es eficaz para la tarea de validación, aunque no sea capaz de representar la relación de orden que sería necesaria para el chequeo de la consistencia de extractos de HCE.

En la sección 6.3 se argumenta que la representación inicial de arquetipos del método *Archeck* es extendida para identificar de forma precisa los errores de modelado. Esta extensión no sería necesaria si los requisitos de validación se limitaran a determinar si un arquetipo es correcto. Así, por ejemplo, en la metodología *Encorsetable* el método de validación ha sido aplicado para determinar si existe una relación de conformidad entre arquetipos que no requiere determinar las causas de la no conformidad. En la comunidad OWL han sido desarrollados métodos para la justificación precisa de las inferencias, como por ejemplo en [79]. Sin embargo, dado que cabe esperar que el usuario final del servicio de validación sea un diseñador de arquetipos, se ha decidido extender la representación OWL de modo que se obtengan inferencias que ayuden a interpretar los errores de modelado al nivel de arquetipos y no de axiomas OWL.

La propuesta de validación ha sido implementada en la herramienta *Archeck* y también ha sido desplegada como un servicio web para facilitar la integración con aplicaciones de terceros. Por ejemplo, el servicio de validación está disponible en la plataforma ArchMS [58], que es el sistema para la gestión de arquetipos desarrollado por nuestro grupo de investigación.

El método de validación de arquetipos ha sido evaluado con los repositorios de arquetipos openEHR públicos más grandes disponibles actualmente (CKM y NHS). La utilidad de la propuesta ha sido demostrada gracias a la identificación de un número significativo de inconsistencias en arquetipos especializados. El descubrimiento de inconsistencias en los arquetipos ha sido un resultado esperado dadas, por una parte, la actual carencia de procesos y guías de buenas prácticas en el diseño de arquetipos, y por otra, la ausencia de soporte para el chequeo de las relaciones de especialización en la mayoría de editores de arquetipos. Sin embargo, no se esperaba un número de inconsistencias tan alto. Este resultado refuerza la necesidad de investigar los procesos de desarrollo de arquetipos. El análisis de los tipos de errores revela que el tipo de error más frecuente aparece en ambos

repositorios, mientras que algunos tipos de errores tienen una presencia desigual en cada repositorio. Este hecho probablemente pone de manifiesto la necesidad de que ambas comunidades de usuarios tomen medidas para prevenir los tipos de errores más frecuentes. Por otro lado, el método *Archeck* también es aplicable a arquetipos ISO EN 13606 sobre los que se han hecho pruebas con éxito. Sin embargo, dado que hasta la fecha hay pocos arquetipos disponibles públicamente, en los experimentos de evaluación del método han sido utilizados arquetipos de los repositorios de openEHR.

Los enlaces terminológicos son un indicador de calidad importante en arquetipos dado que aportan el significado a los términos de la definición. Sin embargo, la mayoría de los arquetipos públicos apenas contienen enlaces con terminologías, lo que presenta un problema que debería ser corregido en los arquetipos actuales. Los documentos CDA utilizados en [74] contienen muchos enlaces con SNOMED-CT y la validación de estos documentos ha mostrado que un elevado número de ellos son inconsistentes. Por ejemplo el 73 % de los enlaces realizados en documentos que representan observaciones. Este resultado apoya la idea de que los enlaces terminológicos son esenciales para la validación de modelos clínicos y que la tasa de errores detectados en la evaluación del método *Archeck* habría sido mayor si tuvieran más enlaces terminológicos.

La representación de arquetipos en OWL también ha sido analizada desde la perspectiva de la definición de métricas de calidad centradas en los enlaces terminológicos. El método *Archeck* incluye la validación de la consistencia de enlaces terminológicos, tal como se ha comentado anteriormente. La representación es adecuada para implementar métricas que evalúen si una característica del enlace cumple unos requisitos, como la relación de inclusión entre conceptos de las terminologías. Sin embargo, otras métricas aplicables a enlaces terminológicos miden aspectos cuantitativos, como por ejemplo, el porcentaje de cohesión semántica de un término en el contexto de su definición. Este requisito ha motivado la necesidad de cuantificar las características de los términos de un arquetipo. La representación OWL como instancias del método *Encorsetable* ha permitido aprovechar las capacidades de la lógica descriptiva en OWL para la definición de métricas. El método propuesto consiste en expresar como clases las características que van a ser evaluadas y contabilizar el número de términos (instancias) que cumplen con esas características. El beneficio de la propuesta es que las métricas se expresan de forma declarativa, la definición puede ser ejecutada y se aprovechan las capacidades de razonamiento de OWL. El editor de ontologías Protégé ofrece soporte para definir métricas en OWL, que puede ser organizadas en ontologías que organicen métricas de calidad de arquetipos.

La definición de métricas en OWL aprovecha las posibilidades de razonamiento sobre el conocimiento asertivo de una ontología. Un razonador puede descubrir relaciones implícitas entre los términos de un arquetipo obtenidas de las características de las propiedades de la ontología. Así, por ejemplo, la combinación de propiedades transitivas, inversas y cadenas de propiedades permite relacionar los enlaces terminológicos de un arquetipo. No obstante, la propuesta realizada debe entenderse como un beneficio adicional de la representación de arquetipos en OWL. Las métricas podrían ser implementadas en un lenguaje de programación como Java utilizando un analizador de arquetipos e interpretando la semántica de los enlaces terminológicos (por ejemplo, las relaciones *is a* de SNOMED-CT). La ventaja que ofrece el marco de definición de métricas propuesto es que establece requisitos sobre las terminologías para que pueden ser enlazadas que favorecen la evaluación de las métricas y que la definición de una métrica es ejecutable sin necesidad de un lenguaje de programación.

La propuesta CIMI ha sacado a la luz la necesidad de modelos clínicos *interoperables*. La metodología *Encorsetable* aborda esta necesidad definiendo una arquitectura de transformación de modelos clínicos capaz de integrar cualquier arquitectura de HCE basada en el modelo dual. La propuesta se basa en la idea de modelos de información *arquetipables*, es decir, modelos que representan restricciones y no estructuras de datos. Se ha propuesto una *Ontología de Restricciones* con las construcciones necesarias para expresar la semántica de los modelos de información y de los modelos clínicos. La propuesta ha sido implementada y evaluada con el modelo de arquetipos de ISO EN 13606 y openEHR.

En general, las propuestas de interoperabilidad semántica de la HCE se basan en establecer correspondencias entre modelos clínicos de distintos estándares que expresan el mismo concepto. La definición de los conceptos clínicos en OWL favorece el uso de herramientas de alineamiento ontológico y la aplicación de razonamiento para el descubrimiento de correspondencias [110, 13]. Sin embargo, los modelos clínicos son activos de conocimiento que pueden emplearse en el diseño de los sistemas de información [104]. Es por ello necesaria la conversión de modelos entre distintos formalismos tal como se propone en CIMI. La herramienta Poseacle Converter ha tratado esta necesidad en el marco de la arquitectura dual y ha sido implementada para la conversión de arquetipos entre los estándares ISO EN 13606 y openEHR.

El principal inconveniente comunicado por los usuarios de Poseacle Converter ha sido no ofrecer la posibilidad de ajustar las correspondencias establecidas entre los estándares. Esta necesidad está motivada por la existencia de más de una propuesta de correspondencia entre los modelos de referencia ISO EN 13606 y openEHR [125, 188]. Sin embargo, ofrecer esta funcionalidad a los usuarios de la herramienta presenta principalmente dos

problemas. En primer lugar, las correspondencias no se expresan utilizando un lenguaje de transformación de modelos estandarizado como QVT [147], ni tampoco mediante una notación de alineamiento de ontologías tal como Alignment API [39], sino que se utilizan reglas de transformación RubyTL [168], que es un lenguaje no estandarizado basado en el lenguaje de programación Ruby. En segundo lugar, las correspondencias no se establecen directamente entre las entidades de los modelos de referencia, sino a través de la *Ontología Común* del marco de trabajo propuesto en *Poseacle* [59], lo que obliga a conocer bien esta ontología para expresar correctamente las reglas de transformación.

Asimismo, la arquitectura de *Poseacle Converter* depende en gran medida de la evolución de la *Ontología Común*. La versión actual de esta ontología es el resultado del proceso de integración de dos estándares relacionados como son ISO EN 13606 y openEHR. La introducción de un modelo de información con características que no puedan expresarse en la *Ontología Común*, como los modelos de información de HL7 CDA o CEM, supondría la necesidad de adaptar y extender la ontología. Por último, la introducción de un nuevo modelo de información apenas podría reutilizar la implementación de las etapas comunes del proceso de transformación. El único módulo que es común a cualquier estándar es el encargado de la transformación de un arquetipo ADL en un modelo AOM. La generación de arquetipos semánticos en OWL o la generación de modelos en ADL es específica de cada modelo de referencia y no puede ser reutilizada.

La metodología *Encorsetable* propone el uso del lenguaje OWL para establecer las correspondencias entre los modelos de referencia. La propuesta además se basa en el uso de dos axiomas fundamentales en OWL, inclusión y equivalencia de clases. Esta característica de la metodología no exige que los usuarios necesiten conocer OWL para definir las correspondencias ni tampoco que éstas deban establecerse de forma totalmente manual. Hay disponibles varios entornos de alineamiento ontológico que facilitan esta tarea aplicando algoritmos de *matching* para el descubrimiento de correspondencias y almacenando los resultados en un formato compatible con OWL.

La *Ontología de Restricciones* de *Encorsetable* tiene un papel distinto a la *Ontología Común* empleada en *Poseacle Converter*. La *Ontología de Restricciones* define las construcciones básicas para la definición de las restricciones que configuran las estructuras de datos (propiedades, cardinalidad, rangos, etc.). Estas definiciones son suficientes para expresar la semántica de los modelos de información y se ha evaluado que también tienen la expresividad necesaria para cubrir la semántica del modelo de arquetipos. En cambio, la *Ontología Común* en *Poseacle* es el resultado de la agregación del conocimiento del dominio de la HCE de los estándares ISO EN 13606 y openEHR.

El marco de trabajo Encorsetable está formado por varios módulos encargados de automatizar las tareas de representación de los modelos en OWL. Actualmente, cualquier modelo de referencia especificado en XML Schema puede ser transformado automáticamente a una representación OWL basada en clases conforme a la Ontología de Restricciones. Asimismo, el módulo de conversión de arquetipos ADL a OWL es independiente del modelo de referencia y también es capaz de realizar la transformación OWL a ADL necesaria en la última etapa del proceso de transformación. No obstante, los arquetipos OWL generados son conformes al modelo de referencia sobre el que están definidos si previamente han sido validados por alguna herramienta como Archeck o LinKEHR-Ed.

La estrategia de transformación de modelos clínicos de Encorsetable está basada en las posibilidades de razonamiento ofrecidas por el lenguaje OWL. Los modelos de información son expresados mediante clases y propiedades en OWL, mientras que los modelos clínicos se declaran instancias de las entidades de los modelos de información. Las correspondencias expresadas entre los modelos de información utilizando axiomas OWL permiten clasificar los modelos clínicos en el modelo de referencia destino. Gracias a que todos los modelos son definidos conforme a las construcciones de la Ontología de Restricciones, las restricciones que establecen los modelos clínicos son migradas en el proceso de transformación. En [13] también se propone un proceso de transformación basado en razonamiento y el uso de axiomas OWL para establecer correspondencias. La diferencia con la propuesta *Encorsetable* es que el propósito de la transformación es la interoperabilidad de datos clínicos.

Los axiomas de inclusión y equivalencia de OWL en general no son suficientes para alinear las estructuras de datos de los modelos de información. La metodología *Encorsetable* aprovecha la semántica de especialización de los arquetipos como un mecanismo para extender los modelos de referencia. La idea ha sido tratar a los arquetipos como cualquier otra entidad del modelo de referencia de cara a la definición de correspondencias. De este modo, los arquetipos permiten aproximar modelos de información que tengan distinta granularidad, como es el caso de ISO EN 13606 y openEHR. Según sean utilizados en el origen o en el destino de la transformación juegan un papel distinto en el proceso. Los arquetipos empleados en el origen actúan de *filtro* de correspondencias. En cambio, en el destino actúan como *arquetipos abstractos* que son completados por el modelo clínico que es transformado. En conjunto el uso de *arquetipos abstractos* ha permitido definir un mecanismo de *plantillas* para expresar la correspondencia de los modelos de información.

A pesar de que la norma ISO EN 13606 fue aprobada en 2008, y unos años antes en la Unión Europea, y de la existencia de otros estándares relevantes a nivel internacional en Informática Médica, como los promovidos por HL7,

la mayoría de los sistemas sanitarios no utilizan estándares. Esta situación supone un serio obstáculo para lograr la interoperabilidad de la HCE. El motivo principal es que las arquitecturas de HCE basadas en estándares son complejas, en especial, las que utilizan la arquitectura dual. La inversión en el desarrollo de sistemas compatibles con estas arquitecturas es alta debido a la dificultad conceptual del modelado dual y a la necesidad de conocer los modelos de información que utilizan en estos estándares.

La herramienta *ArchForms* es un generador de aplicaciones sanitarias conformes a la norma ISO EN 13606. El principal objetivo de esta herramienta es la generación de sistemas basados en estándares en el ámbito sanitario, y, en consecuencia, lograr la interoperabilidad de la información clínica. El generador parte de la definición de uno o varios arquetipos y obtiene una aplicación sanitaria implementada con tecnologías web totalmente funcional. La generación de aplicaciones a partir de arquetipos ha sido un tema de investigación activo en los últimos años. Desde que en 2006 Schuler *et al.* [172] demostraron que los arquetipos contienen la información necesaria para poder obtener automáticamente una aplicación sanitaria, han sido muchas las herramientas que se han desarrollado siguiendo esa propuesta [24, 157, 3, 112]. Asimismo, también se han utilizado arquetipos para obtener interfaces gráficas en sistemas HCE heredados, en los que la estrategia seguida ha sido la conversión de arquetipos en los modelos de información propietarios de esos sistemas [28, 47].

Sin embargo, a través de la Fundación openEHR se ha realizado el proyecto *Opereffa* [153] en el que se han analizado los requisitos de interfaz de usuario que exige la arquitectura dual. La conclusión más importante de este trabajo es que el principal reto al que se enfrenta el desarrollo de aplicaciones sanitarias es la evolución tecnológica. El generador *ArchForms* parte de esta idea para implementar una arquitectura generativa que es independiente de la plataforma tecnológica. El proceso de generación se organiza en distintas fases que desacoplan los requisitos de información clínica, la presentación visual de los modelos y la tecnología que implementa la solución. El generador ha sido evaluado sobre el framework JSF, propuesto en *Opereffa*, en el que se han utilizado tres bibliotecas de componentes gráficos con diferentes capacidades visuales, y en especial, una de ellas orientada a las interfaces de dispositivos móviles.

La arquitectura de la herramienta *ArchForms* ha empleado técnicas de Ingeniería de Modelos en su desarrollo. En primer lugar, se ha analizado la relación entre la arquitectura de modelado MDA y las arquitecturas de HCE basadas en el modelo dual. Las conclusiones de este análisis han permitido construir una arquitectura generativa construida sobre la propuesta ontológica *Poseacle*. En segundo lugar, la semántica de visualización y captura de la HCE expresada en los arquetipos ha sido representada en modelos genéricos de interfaz de usuario soportados por el estándar XForms. La definición

abstracta de interfaces de usuario está en consonancia con las recomendaciones de van der Linden *et al.* [119] que proponen la representación de las interfaces gráficas de las aplicaciones sanitarias con modelos abstractos independientes de una plataforma, y que estos modelos sea utilizados para obtener interfaces gráficas para tecnologías específicas. En general, la aplicación de técnicas de Ingeniería de Modelos en el proceso de desarrollo consigue reducir los costes de la evolución tecnológica de las aplicaciones sanitarias.

Finalmente, la herramienta ArchForms pretende promover los principios del modelado dual, es decir, que los expertos en el dominio clínico sean los responsables de la gestión de los conceptos clínicos (arquetipos), y que a partir de ellos se obtengan aplicaciones funcionales sin tener que ocuparse de los detalles técnicos de implementación.

## 9.2. Contribuciones

La principal aportación de esta tesis ha sido el estudio de la tecnología OWL como soporte de la representación de arquetipos en la arquitectura dual. Las soluciones obtenidas han sido aplicadas al aseguramiento de la calidad en arquetipos y a la interoperabilidad de la HCE:

- Diseño de un método de análisis de la consistencia de arquetipos. En primer lugar, se ha propuesto una representación de la arquitectura dual en OWL que expresa formalmente la semántica estructural del modelo de arquetipos. De este modo, un arquetipo es correcto respecto al modelo de referencia si un razonador no detecta que su definición sea inconsistente. La relación de especialización entre arquetipos se ha formalizado a través del axioma de subsunción en OWL. El método de validación se basa en la identificación de esta inferencia como indicador de que la especialización es consistente. Finalmente, el método de validación ha sido implementado en la herramienta *Archeck*.
- Marco de definición y evaluación de métricas de calidad de arquetipos. Las métricas de calidad se han clasificado en dos grupos, las que expresan restricciones y las que cuantifican características de la definición del arquetipo. Para el primer grupo, la métrica se expresa como una restricción más del modelo de arquetipos y se ha ilustrado su aplicabilidad a través de la validación de la consistencia de enlaces terminológicos en relaciones de especialización. En el segundo grupo, la propuesta consiste en la definición de métricas como clases OWL y en la clasificación de los términos del arquetipo de acuerdo a esas métricas. El valor de la métrica se expresa mediante una fórmula aritmética que usa el número de términos del arquetipo que pertenecen a cada clase.

- Metodología para la interoperabilidad de modelos clínicos en arquitecturas de modelo dual. El marco de trabajo *Encorsetable* promueve el uso de OWL en todas las etapas del proceso de transformación. El principio metodológico consiste en aproximar la transformación de modelos clínicos como un proceso de clasificación. La propuesta garantiza que las restricciones que definen a un modelo clínico son migradas correctamente gracias a que son expresadas de acuerdo a una ontología común, y la correspondencia entre las estructuras de datos se expresa formalmente como axiomas en el lenguaje OWL. Asimismo, la metodología promueve el concepto de arquetipo como herramienta que facilita la definición de correspondencias entre modelos de información.
- Propuesta de integración de la arquitectura dual en Ingeniería de Modelos. El lenguaje OWL ha permitido establecer un puente entre las especificaciones de los estándares de HCE de modelo dual y el espacio tecnológico de la Ingeniería de Modelos. La representación de arquetipos como modelos permite que sean utilizados para guiar la generación automática de aplicaciones utilizando un proceso de desarrollo dirigido por modelos. La semántica que expresa los requisitos de validación, visualización y captura de la HCE ha podido ser representada de forma genérica y su correspondencia con los arquetipos se ha definido mediante transformaciones de modelos.
- Desarrollo de un generador de aplicaciones sanitarias para el estándar ISO EN 13606. La herramienta *ArchForms* utiliza técnicas y herramientas de Ingeniería de Modelos para la generación de aplicaciones sanitarias basadas en arquetipos. La arquitectura de la herramienta es extensible e independiente de cualquier plataforma tecnológica. Las aplicaciones generadas representan la información clínica como extractos de HCE conforme al estándar ISO EN 13606.

### 9.3. Trabajo futuro

El trabajo de esta tesis doctoral ha sido aplicado al modelo de arquetipos del estándar ISO EN 13606 y openEHR. En concreto, se han propuesto varias representaciones de arquetipos en OWL que han sido aplicadas a la validación de arquetipos, definiciones de métricas de calidad, soporte para un marco de interoperabilidad de modelos clínicos y para la generación automática de aplicaciones sanitarias. Por tanto, el trabajo en general puede ser extendido a otras arquitecturas de HCE de modelo dual, como por ejemplo, el modelo de *templates* de HL7 CDA y CEM.



El método *Archeck* trabaja sobre las inferencias obtenidas de la representación de los términos de un arquetipo como clases en OWL. El análisis de las inferencias también podría aplicarse a otras tareas como la identificación de estructuras de datos equivalentes. Esto sería de gran utilidad para reutilizar estructuras ya definidas en el mismo arquetipo a través de referencias internas o factorizando estructuras comunes que pueden ser reutilizadas a través de *slots*. Por tanto, la representación de *Archeck* podría incorporarse a la aplicación de métodos de *refactorización* de arquetipos, en la línea de las técnicas que se aplican en Ingeniería del Software.

El análisis de los repositorios de arquetipos ha revelado que en distintas comunidades de usuarios se están adoptando criterios de modelado distintos. Sería de interés el análisis de los patrones de diseño presentes en los repositorios de arquetipos y recopilar un conjunto de buenas prácticas de modelado.

El rendimiento de las tareas de razonamiento se ve afectado sensiblemente por el uso de grandes terminologías como SNOMED-CT. La definición de subconjuntos de terminologías es una solución eficaz cuando la base de conocimiento de la tarea está formada por unos pocos arquetipos. Una continuación del trabajo sería explorar conjuntamente varias representaciones de arquetipos en OWL adaptadas al tipo de validación. Así, por ejemplo, una representación de arquetipos en OWL EL permitiría aprovechar los razonadores optimizados para la lógica descriptiva  $\mathcal{EL}++$  en la que está expresada la terminología SNOMED-CT.

La propuesta de métricas de calidad está soportada por las definiciones de una ontología, en concreto, la ontología de restricciones de la metodología *Encorsetable*. A pesar de que las métricas pueden ser expresadas de forma declarativa, el conocimiento de la ontología es una limitación para su uso. Un modo más intuitivo de definir métricas para diseñadores de arquetipos sería expresarlas en lenguaje ADL. Un arquetipo podría representar un *patrón de calidad* que tendría una representación equivalente en OWL. En este sentido, se podría estudiar la extensión del método *Archeck* para clasificar los términos del arquetipo en función de las métricas.

En el campo de la Ingeniería del Software existen estándares para la evaluación de la calidad de productos software, como la norma ISO 25000, *Software Product Quality Requirements and Evaluation (SQuaRE)*[94]. En nuestro grupo de investigación se ha realizado una adaptación de esta norma para la evaluación de la calidad de ontologías. El resultado es la metodología *Ontology Quality Requirements and Evaluation (OQuaRE)*[48] que incluye, entre otros componentes, un modelo de calidad de ontologías definido por características, subcaracterísticas, métricas y las asociaciones características-subcaracterísticas y subcaracterísticas-métricas. Las métricas incluidas en OQuaRE se basan en el estudio de diferentes tipos de características de calidad de las ontologías. Puesto que las propuestas de esta tesis se basan

en hacer uso de OWL para la definición de métricas, sería interesante comprobar cómo se podría adaptar OQuARE a la evaluación de arquetipos y cómo se podrían integrar las métricas de calidad de arquetipos existentes en dicha metodología.

La propuesta metodológica *Encorsetable* facilita que los expertos en HCE puedan establecer las correspondencias entre los modelos de información. Una extensión del framework sería proporcionar un entorno de edición integrado para la gestión de correspondencias, la selección de algoritmos de alineamiento y la asistencia en el establecimiento manual de correspondencias. Además, este entorno de gestión de correspondencias debería facilitar el uso de arquetipos *abstractos*, facilitando la detección de inconsistencias en las definiciones y proporcionando ayuda en la definición de *plantillas de transformación*.

La herramienta *ArchFoms* ha sido diseñada para el estándar ISO EN 13606. Una línea de trabajo sería extenderla a la especificación openEHR y al modelo de plantillas propuesto por esta especificación. Por otro lado, la generación automática de las aplicaciones apenas puede ser ajustada sin modificar los transformadores de modelos. Introducir un metamodelo para controlar aspectos de interfaz como el tipo de control asociado a una entidad del modelo de referencia y otras preferencias de visualización sería de utilidad en la herramienta.

## 9.4. Publicaciones

### Publicaciones JCR

- MENÁRGUEZ-TORTOSA, M. y FERNÁNDEZ-BREIS, J. T. OWL-based Reasoning Methods for Validating Archetypes. *Journal of Biomedical Informatics*, vol. 46, 304-317, 2013.
- MENÁRGUEZ-TORTOSA, M., MARTÍNEZ-COSTA, C. y FERNÁNDEZ-BREIS, J. T. A generative tool for building health applications driven by ISO 13606 archetypes. *Journal of Medical Systems*, vol. 36(5), 3063-3075, 2012.
- MALDONADO, J. A., MARTÍNEZ-COSTA, C., MONER, D., MENÁRGUEZ-TORTOSA, M., BOSCA, D., MIÑARRO GIMÉNEZ, J. A., FERNÁNDEZ-BREIS, J. T. y ROBLES, M. Using the ResearchEHR platform to facilitate the practical application of the EHR standards. *Journal of Biomedical Informatics*, vol. 45(4), 746-762, 2012.
- MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M. y FERNÁNDEZ-BREIS, J. T. Clinical data interoperability based on archetype transformation. *Journal of Biomedical Informatics*, vol. 44(5), 869-880, 2011.

- MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M. y FERNÁNDEZ-BREIS, J. T. An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes. *Journal of Biomedical Informatics*, vol. 43(5), 736-746, 2010.
- MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M., FERNÁNDEZ-BREIS, J. T. y MALDONADO, J. A. A model-driven approach for representing clinical archetypes for Semantic Web environments. *Journal of Biomedical Informatics*, vol. 42(1), 150-164, 2009.

### Capítulos de libro

- MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M., MALDONADO, J. A. y FERNÁNDEZ-BREIS, J. T. *Semantic Web technologies for managing EHR-related clinical knowledge*. Semantic Web, páginas 210-219, Ed. IN-TECH, ISBN 978-953-7619-33-6, 2009.
- MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M. y FERNÁNDEZ-BREIS, J. T. *Interoperability of EHR Systems based on Semantic Representation and Transformation Models*. (en proceso de publicación).

### Publicaciones en congresos

- MENÁRGUEZ-TORTOSA, M. y FERNÁNDEZ-BREIS, J. T. Validation of the openEHR archetype library by using OWL reasoning. En *XXIII International Conference of the European Federation for Medical Informatics, MIE 2011*, Studies in Health Technology and Informatics, volumen 789, páginas 169-175, 2011.
- MENÁRGUEZ-TORTOSA, M., MARTÍNEZ-COSTA, C. y FERNÁNDEZ-BREIS, J. T. Validating Archetypes using OWL Reasoning. En *II OpenHealth Spain Symposium*, 2010.
- MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M., LÓPEZ-LÓPEZ, P. D. RUBIO-GONZÁLEZ, M. M. y FERNÁNDEZ-BREIS, J. T. ArchForms: Automatic generation of applications based on ISO 13606 archetypes. En *II OpenHealth Spain Symposium*, 2010.
- MARTÍNEZ-COSTA, C., MIÑARRO-GIMÉNEZ, J. A., MENÁRGUEZ-TORTOSA, M., VALENCIA-GARCÍA, R. y FERNÁNDEZ-BREIS, J. T. Flexible Semantic Querying of Clinical Archetypes. En *KES 2010*, Lecture Notes in Computer Science, volumen 6276, 597-606, ISSN 0302-9743, 2010.
- ROBLES, M., FERNÁNDEZ-BREIS, J. T., MALDONADO, J. A., MONER, D., MARTÍNEZ-COSTA, C. y MENÁRGUEZ-TORTOSA, M. ResearchEHR: Use of semantic web technologies and archetypes for the description of

EHRs. En *EFMI STC 2010, Studies in Health Technology and Informatics*, volumen 155, páginas 129-135, ISSN 0926-9630, 2010.

- MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M. y FERNÁNDEZ-BREIS, J. T. ARCHFORMS: Generación automática de aplicaciones ISO 13606. En *Inforsalud 2010*, ISBN 978-84-692-9230-3, 2010.
- MIRANDA-MENA, T., MARTÍNEZ-COSTA, C., MONER, D., MENÁRGUEZ-TORTOSA, M., MALDONADO, J. A., ROBLES-VIEJO, M. y FERNÁNDEZ-BREIS, J. T. MIURAS 2: Motor de integración universal para aplicaciones sanitarias avanzadas. En *Inforsalud 2010*, ISBN 978-84-692-9230-3, 2010.
- MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M. y FERNÁNDEZ-BREIS, J. T. Towards UNE-EN 13606 and OpenEHR Archetype-based semantic inter-operability. En *XXII International Conference of the European Federation for Medical Informatics, MIE 2009, Studies in Health Technology and Informatics*, volumen 150, 260-264 páginas 260-264, ISBN 978-1-60750-044-5, 2009.
- MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M., VALENCIA-GARCÍA, R., MALDONADO, J. A. y FERNÁNDEZ-BREIS, J. T. Transformación automática de arquetipos UNE-EN 13606 y openEHR para facilitar la interoperabilidad semántica. En *Inforsalud 2009*, ISBN 978-84-691-9561-1, 2009.
- MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M. y FERNÁNDEZ-BREIS, J. T. Ontology-based archetype interoperability and management. En *First Spanish Open-Health*, 2009.
- FERNÁNDEZ-BREIS, J. T., MENÁRGUEZ-TORTOSA, M., MARTÍNEZ-COSTA, C., FERNÁNDEZ-BREIS, E., HERRERO, J. SÁNCHEZ-CUADRADO, J., VALENCIA-GARCÍA, R. y ROBLES, M. A Semantic Web-based System for Managing Clinical Archetypes. En *30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, páginas 1482-1485, ISBN 9781424418145, 2008.
- FERNÁNDEZ-BREIS, J. T., MENÁRGUEZ-TORTOSA, M., MONER, D., VALENCIA-GARCÍA, R., MALDONADO, J. A., VIVANCOS-VICENTE, P. J. y MARTÍNEZ-BÉJAR, R. An ontological infrastructure for the semantic integration of clinical archetypes. En *Advances in Knowledge Acquisition and Management*, páginas 156-167, Springer Berlin Heidelberg, 2006.

# Bibliografía

- [1] AHN, S. Development and application of development principles for clinical information model. *Journal of the Korea Academia-Industrial Cooperation Society 2010*, vol. 11(8), páginas 2899–2905, 2010.
- [2] AHN, S., HUFF, S. M., KIM, Y. y KALRA, D. Quality metrics for detailed clinical models. *International Journal of Medical Informatics*, vol. Article in Press DOI: 10.1016/j.ijmedinf.2012.09.006, 2012.
- [3] ATALAG, K., YANG, H., TEMPERO, E. y WARREN, J. Model driven development of clinical information systems using openEHR. *Studies in Health Technology and Informatics*, vol. 169, páginas 849–853, 2011.
- [4] BAADER, F. Terminological cycles in a description logic with existential restrictions. En *International Joint Conference on Artificial Intelligence*, vol. 18, páginas 325–330. Lawrence Erlbaum Associates Ltd., 2003.
- [5] BAADER, F., BRANDT, S. y LUTZ, C. Pushing the EL envelope. En *International Joint Conference on Artificial Intelligence*, vol. 19. Lawrence Erlbaum Associates Ltd., 2005.
- [6] BAADER, F., CALVANESE, D., MCGUINNESS, D., NARDI, D. y PATEL-SCHNEIDER, P. *The description logic handbook: Theory, implementation and applications*. Cambridge University Press, 2003.
- [7] BAADER, F., HORROCKS, I. y SATTLER, U. Description logics as ontology languages for the semantic web. *Mechanizing Mathematical Reasoning*, páginas 228–248, 2005.
- [8] BEALE, T. The GEHR software architecture for a reliable EHR. En *Toward an Electronic Health Record Europe*, vol. 99, páginas 328–339. 1999.
- [9] BEALE, T. Archetypes: Constraint-based domain models for future-proof information systems. *OOPSLA 2002 workshop on behavioural semantics*, páginas 1–18, 2002.

- [10] BEALE, T. The openEHR archetype model: Archetype Object Model. 2007. Disponible en <https://raw.githubusercontent.com/openEHR/specifications/Release-1.0.1/publishing/architecture/am/aom.pdf> (último acceso, abril, 2013).
- [11] BEALE, T. y HEARD, S. Archetype definition language (ADL). 2007. Disponible en <https://raw.githubusercontent.com/openEHR/specifications/Release-1.0.1/publishing/architecture/am/adl.pdf> (último acceso, abril, 2013).
- [12] BECHHOFFER, S., VAN HARMELLEN, F., HENDLER, J., HORROCKS, I., MCGUINNESS, D. L., PATEL-SCHNEIDER, P. F., STEIN, L. A. ET AL. OWL Web Ontology Language reference - W3C Recommendation. 2004. Disponible en <http://www.w3.org/TR/owl-ref/> (último acceso, abril, 2013).
- [13] BERGES, I., BERMÚDEZ, J. y ILLARRAMENDI, A. Toward semantic interoperability of electronic health records. *Information Technology in Biomedicine, IEEE Transactions on*, vol. 16(3), páginas 424–431, 2012.
- [14] BERNERS-LEE, T., HENDLER, J., LASSILA, O. ET AL. The Semantic Web. *Scientific american*, vol. 284(5), páginas 28–37, 2001.
- [15] BICER, V., KILIC, O., DOGAC, A. y LALECI, G. Archetype-based semantic interoperability of web service messages in the health care domain. *International Journal on Semantic Web and Information Systems (IJSWIS)*, vol. 1(4), páginas 1–23, 2005.
- [16] BICER, V., LALECI, G. B., DOGAC, A. y KABAK, Y. Artemis message exchange framework: semantic interoperability of exchanged messages in the healthcare domain. *SIGMOD Rec.*, vol. 34(3), páginas 71–76, 2005. ISSN 0163-5808.
- [17] BISBAL, J. y BERRY, D. Archetype alignment: A two-level driven semantic matching approach to interoperability in the clinical domain. En *Proceedings of the International Conference on Health Informatics*, páginas 216–221. INSTICC Press, 2009.
- [18] BLOBEL, B. y OEMIG, F. Establishing semantic interoperability between HL7 v2. x and V3: a Communication Standards Ontology (CSO). *Journal of Health Informatics*, vol. 3(4), páginas 153–157, 2011.
- [19] BODENREIDER, O. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, vol. 32(suppl 1), páginas D267–D270, 2004.

- [20] BOLEY, H., HALLMARK, G., KIFER, M., PASCHKE, A., POLLERES, A. y REYNOLDS, D. RIF Core Dialect. W3C Recommendation. 2010. Disponible en <http://www.w3.org/TR/rif-core/> (último acceso, abril, 2013).
- [21] BOONE, K. W. The HL7 clinical document architecture. En *The CDA TM book*, páginas 17–21. Springer, 2011.
- [22] BOONE, K. W. Templates. En *The CDA TM book*, páginas 265–274. Springer, 2011.
- [23] BOZSAK, E., EHRIG, M., HANDSCHUH, S., HOTH, A., MAEDCHE, A., MOTIK, B., OBERLE, D., SCHMITZ, C., STAAB, S., STOJANOVIC, L. ET AL. Kaon - towards a large scale semantic web. En *E-Commerce and Web Technologies*, páginas 304–313. Springer, 2002.
- [24] BRASS, A., MONER, D., HILDEBRAND, C. y ROBLES, M. Standardized and flexible health data management with an archetype driven EHR system (EHRflex). *Studies in Health Technology and Informatics*, vol. 155, páginas 212–218, 2010.
- [25] CASTANO, S., FERRARA, A. y MONTANELLI, S. Matching ontologies in open networked systems: Techniques and applications. En *Journal on Data Semantics V*, páginas 25–63. Springer, 2006.
- [26] CHAN, H. HL7 RIM ontology in OWL. 2010. Disponible en <http://esw.w3.org/topic/HCLS/ACPPTaskForce?action=AttachFile&do=get&target=RIMV3OWL.zip> (último acceso, abril, 2013).
- [27] CHANDRASEKARAN, B., JOSEPHSON, J. R. y BENJAMINS, V. R. What are ontologies, and why do we need them? *Intelligent Systems and Their Applications, IEEE*, vol. 14(1), páginas 20–26, 1999.
- [28] CHEN, R., KLEIN, G., SUNDVALL, E., KARLSSON, D. y ÅHLFELDT, H. Archetype-based conversion of EHR content models: pilot experience with a regional EHR system. *BMC medical informatics and decision making*, vol. 9(1), página 33, 2009.
- [29] CHEN, R., KLEIN, G. ET AL. The openEHR Java reference implementation project. *Studies in health technology and informatics*, vol. 129(1), páginas 58–62, 2007.
- [30] CHUTE, C. Clinical classification and terminology. *Journal of the American Medical Informatics Association*, vol. 7(3), páginas 298–303, 2000.
- [31] CLINICAL INFORMATION MODELING INITIATIVE. CIMI wiki. 2013. Disponible en [http://informatics.mayo.edu/CIMI/index.php/Main\\_Page](http://informatics.mayo.edu/CIMI/index.php/Main_Page) (último acceso, abril, 2013).

- [32] COENEN, A. y KIM, T. Y. Development of terminology subsets using ICNP®. *International Journal of Medical Informatics*, vol. 79(7), páginas 530 – 538, 2010. ISSN 1386-5056.
- [33] CONDE, A. M. y BERRY, D. *Towards best practice in the Archetype Development Process*. Proyecto Fin de Carrera, Trinity College Dublin, 2010.
- [34] CORNET, R. y SPACKMAN, K. A., editores. *Proceedings of the Third International Conference on Knowledge Representation in Medicine, Phoenix, Arizona, USA, May 31st - June 2nd, 2008*, vol. 410 de *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.
- [35] COYLE, J., HERAS, Y., ONIKI, T. y HUFF, S. Clinical Element Model. 2008. Disponible en [http://informatics.mayo.edu/sharp/images/e/e2/CEM\\_Reference20081114.pdf](http://informatics.mayo.edu/sharp/images/e/e2/CEM_Reference20081114.pdf) (último acceso, abril, 2013).
- [36] COYLE, J., MORI, A. y HUFF, S. Standards for detailed clinical models as the basis for medical data exchange and decision support. *International journal of medical informatics*, vol. 69(2), páginas 157–174, 2003.
- [37] CRUZ, I. F., ANTONELLI, F. P. y STROE, C. Agreementmaker: efficient matching for large real-world schemas and ontologies. *Proceedings of the VLDB Endowment*, vol. 2(2), páginas 1586–1589, 2009.
- [38] DAN BRICKLEY, R. G. RDF vocabulary description language 1.0: RDF schema, W3C Recommendation. 2004. Disponible en <http://www.w3.org/TR/rdf-schema/> (último acceso, abril, 2013).
- [39] DAVID, J., EUZENAT, J., SCHARFFE, F. y TROJAHN DOS SANTOS, C. The alignment api 4.0. *Semantic web*, vol. 2(1), páginas 3–10, 2011.
- [40] DEWSBURY, R. *Google web toolkit applications*. Addison-Wesley Professional, 2007.
- [41] DOBREV, A., JONES, T., STROETMANN, V., STROETMANN, K., VATTER, Y. y PENG, K. Interoperable ehealth is worth it-securing benefits from electronic health records and eprescribing. *Bonn/Brussels: European Commission on Information Safety and Media*, 2010.
- [42] DOGAC, A., LALECI, G., KIRBAS, S., KABAK, Y., SINIR, S., YILDIZ, A. y GURCAN, Y. Artemis: Deploying semantically enriched web services in the healthcare domain. *Information Systems*, vol. 31(4), páginas 321–339, 2006.



- [43] DOGAC, A., NAMLI, T., OKCAN, A., LALECI, G., KABAK, Y. y EICHELBERG, M. Key issues of technical interoperability solutions in ehealth and the ride project. *Software R&D Center, Dept. of Computer Eng., Middle East Technical University, Ankara*, vol. 6531, 2007.
- [44] DOLIN, R. H., ALSCHULER, L., BOYER, S., BEEBE, C., BEHLEN, F. M., BIRON, P. V. y SHABO, A. HL7 Clinical Document Architecture, Release 2. *Journal of the American Medical Informatics Association*, 2005.
- [45] DON NEWSHAM, C. M., DIPAK KALRA. HL7 Implementation Guide for 13606. 2006. Disponible en <http://www.hl7.org.uk/repository/uploads/402/1/HL7%20Implementation%20Guide%20for%2013606%20scopev0r6.doc> (último acceso, abril, 2013).
- [46] DRUMMOND, N., RECTOR, A., STEVENS, R., MOULTON, G., HORRIDGE, M., WANG, H. y SEIDENBERG, J. Putting OWL in order: Patterns for sequences in OWL. *Proc. 2nd Workshop on OWL: Experiences and Directions*, 2006.
- [47] DUFTSCHMID, G., CHALOUPKA, J. y RINNER, C. Towards plug-and-play integration of archetypes into legacy electronic health record systems: the archimed experience. *BMC medical informatics and decision making*, vol. 13(1), páginas 1–12, 2013.
- [48] DUQUE-RAMOS, A., FERNÁNDEZ-BREIS, J. T., INIESTA, M., DUMONTIER, M., EGAÑA ARANGUREN, M., SCHULZ, S., AUSSENAC-GILLES, N. y STEVENS, R. Evaluation of the OQuARE framework for ontology quality. *Expert Systems with Applications*, vol. 40(7), páginas 2696–2703, 2013.
- [49] EHRIG, M. *Ontology alignment: bridging the semantic gap*, vol. 4. Springer Science+ Business Media, 2007.
- [50] EICHELBERG, M., ADEN, T., RIESMEIER, J., DOGAC, A. y LALECI, G. B. A survey and analysis of electronic healthcare record standards. *ACM Comput. Surv.*, vol. 37(4), páginas 277–315, 2005. ISSN 0360-0300.
- [51] EN 13606 ASSOCIATION. 2013. Disponible en <http://www.en13606.eu> (último acceso, abril, 2013).
- [52] EUZENAT, J. An API for ontology alignment. En *The Semantic Web–ISWC 2004*, páginas 698–712. Springer, 2004.
- [53] EUZENAT, J., MEILICKE, C., STUCKENSCHMIDT, H., SHVAIKO, P. y TROJAHN, C. Ontology alignment evaluation initiative: six years of experience. En *Journal on data semantics XV*, páginas 158–192. Springer, 2011.
- [54] EUZENAT, J. y SHVAIKO, P. *Ontology matching*. Springer-Verlag, 2007.

- [55] EVERMANN, J. A UML and OWL description of Bungeés upper-level ontology model. *Software & Systems Modeling*, vol. 8(2), páginas 235–249, 2009.
- [56] FEIGENBAUM, L., HERMAN, I., HONGSERMEIER, T., NEUMANN, E. y STEPHENS, S. The semantic web in action. *Scientific American Magazine*, vol. 297(6), páginas 90–97, 2007.
- [57] FERNÁNDEZ-BREIS, J. T. y MARTINEZ-BEJAR, R. A cooperative framework for integrating ontologies. *International Journal of Human-Computer Studies*, vol. 56(6), páginas 665–720, 2002.
- [58] FERNÁNDEZ-BREIS, J. T., MENÁRGUEZ-TORTOSA, M., MARTÍNEZ-COSTA, C., FERNÁNDEZ-BREIS, E., HERRERO-SEMPERE, J., MONER, D., SÁNCHEZ, J., VALENCIA-GARCÍA, R. y ROBLES, M. A Semantic Web-based System for Managing Clinical Archetypes. *Conference Proceedings of the International Conference of IEEE Engineering in Medicine and Biology Society*, vol. 2008, páginas 1482–1485, 2008.
- [59] FERNÁNDEZ-BREIS, J. T., MENÁRGUEZ-TORTOSA, M., MONER, D., VALENCIA-GARCÍA, R., MALDONADO, J. A., VIVANCOS-VICENTE, P. J., MIRANDA-MENA, T. G. y MARTÍNEZ-BÉJAR, R. An ontological infrastructure for the semantic integration of clinical archetypes. En *Advances in Knowledge Acquisition and Management*, páginas 156–167. Springer, 2006.
- [60] FERNANDEZ-LOPEZ, M., GOMEZ-PEREZ, A. y JURISTO, N. Methontology: from ontological art towards ontological engineering. En *Proceedings of the AAAI97 Spring Symposium*, páginas 33–40. Stanford, USA, 1997.
- [61] FERNÁNDEZ-LÓPEZ, M., GÓMEZ-PÉREZ, A. ET AL. Overview and analysis of methodologies for building ontologies. *The Knowledge Engineering Review*, vol. 17(2), páginas 129–156, 2002.
- [62] GALL, W., REHNELT, C., BOLDT, G. y PREMAUER, W. Archimed: a medical information and retrieval system. *Meth Inform Med*, vol. 38, páginas 16–24, 1999.
- [63] GARDE, S., KNAUP, P., HOVENGA, E. y HEARD, S. Towards semantic interoperability for electronic health records-domain knowledge governance for openehr archetypes. *Methods of information in medicine*, vol. 46(3), páginas 332–343, 2007.
- [64] GEARY, D. y HORSTMANN, C. *Core JavaServer Faces, third edition*. Prentice Hall Press, Upper Saddle River, NJ, USA, second edición, 2010. ISBN 9780131738867.

- [65] GOMEZ-PEREZ, A., CORCHO-GARCIA, O. y FERNANDEZ-LOPEZ, M. Ontological engineering. *Computing Reviews*, vol. 45(8), páginas 478–479, 2004.
- [66] GOOSSEN, W., GOOSSEN-BAREMANS, A. y VAN DER ZEL, M. Detailed clinical models: a review. *Healthcare informatics research*, vol. 16(4), páginas 201–214, 2010.
- [67] GRIMSON, W., BERRY, D., GRIMSON, J., STEPHENS, G., FELTON, E., GIVEN, P. y O'MOORE, R. Federated healthcare record server-the synapses paradigm. *International Journal of Medical Informatics*, vol. 52(1), páginas 3–27, 1998.
- [68] GRUBER, T. ET AL. A translation approach to portable ontology specifications. *Knowledge acquisition*, vol. 5(2), páginas 199–220, 1993.
- [69] GUARINO, N. *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*, vol. 46. Ios PressInc, 1998.
- [70] HAARSLEV, V., HIDDE, K., MÖLLER, R. y WESSEL, M. The racerpro knowledge representation and reasoning system. *Semantic Web*, vol. 3(3), páginas 267–277, 2012.
- [71] HEALTH LEVEL SEVEN. HL7 Template and Archetype Architecture Version 3.0. 2003. Disponible en [http://www.hl7.org/documentcenter/public/wg/template/HL7\\_Atlanta\\_10\\_20\\_04.doc](http://www.hl7.org/documentcenter/public/wg/template/HL7_Atlanta_10_20_04.doc) (último acceso, abril, 2013).
- [72] HEALTH LEVEL SEVEN. HL7 and openEHR datatype mappings. 2007. Disponible en [http://wiki.hl7.org/index.php?title=OpenEHR\\_datatypes\\_mapping](http://wiki.hl7.org/index.php?title=OpenEHR_datatypes_mapping) (último acceso, abril, 2013).
- [73] HEALTH LEVEL SEVEN. HL7 Reference Information Model. 2013. Disponible en <http://www.hl7.org/implement/standards/rim.cfm> (último acceso, abril, 2013).
- [74] HEYMANS, S., MCKENNIREY, M. y PHILLIPS, J. Semantic validation of the use of SNOMED CT in HL7 clinical documents. *Journal of Biomedical Semantics*, vol. 2(1), páginas 1–16, 2011.
- [75] HL7. HL7 v3 Guide. 2002. Disponible en <http://www.hl7.org/documentcenter/public/wg/mnm/hdf/v3guide.doc> (último acceso, abril, 2013).
- [76] HLAVATS, I. *JSF 1.2 Components*. Packt Pub Limited, 2009.

- [77] HOEKSTRA, R. *Ontology Representation - Design Patterns and Ontologies that Make Sense*, vol. 197 de *Frontiers in Artificial Intelligence and Applications*. IOS Press, 2009. ISBN 978-1-60750-013-1.
- [78] HORRIDGE, M. y BECHHOFFER, S. The OWL API: a Java API for working with OWL 2 ontologies. *Proc. of OWL Experiences and Directions*, 2009.
- [79] HORRIDGE, M., PARSIA, B. y SATTTLER, U. Laconic and Precise Justifications in OWL. En *International Semantic Web Conference*, páginas 323–338. 2008.
- [80] HORRIDGE, M. y PATEL-SCHNEIDER, P. F. OWL 2 web ontology language manchester syntax - W3C Working Group Note 11. 2012. Disponible en <http://www.w3.org/TR/owl2-manchester-syntax/> (último acceso, abril, 2013).
- [81] HORROCKS, I., KUTZ, O. y SATTTLER, U. The even more irresistible SROIQ. En *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR 2006)*, páginas 57–67. 2006.
- [82] HORROCKS, I. y PATEL-SCHNEIDER, P. F. Reducing OWL entailment to description logic satisfiability. En *The Semantic Web-ISWC 2003*, páginas 17–29. Springer, 2003.
- [83] HORROCKS, I., PATEL-SCHNEIDER, P. F. y VAN HARMELEN, F. From SHIQ and RDF to OWL: The making of a web ontology language. *Web semantics: science, services and agents on the World Wide Web*, vol. 1(1), páginas 7–26, 2003.
- [84] HORROCKS, I., SATTTLER, U. y TOBIES, S. Practical reasoning for expressive description logics. En *Logic for Programming and Automated Reasoning*, páginas 161–180. Springer, 1999.
- [85] HU, W. y QU, Y. Falcon-AO: A practical ontology matching system. *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 6(3), páginas 237–239, 2008.
- [86] HUFF, S. M. Ontologies, vocabularies, and data models. *Clinical Decision Support: The Road Ahead*, páginas 307–324, 2007.
- [87] IBIME GROUP. LinkEHR Normalization Platform. 2013. Disponible en <http://www.linkehr.com/> (último acceso, abril, 2013).
- [88] IFOMIS. Basic formal ontology. 2013. Disponible en <http://www.ifomis.org/bfo> (último acceso, abril, 2013).
- [89] IHTSDO. SNOMED-CT. 2013. Disponible en <http://www.ihtsdo.org/snomed-ct/> (último acceso, abril, 2013).

- [90] INGRAM, D. The good european health record. *Health in the new communication age*, MF Lares, MF Ladeira and JP Christensen (Eds), IOS, páginas 66–74, 1995.
- [91] IQBAL, A. An OWL-DL ontology for the HL7 reference information model. *Toward Useful Services for Elderly and People with Disabilities*, páginas 168–175, 2011.
- [92] ISO. ISO/TR 18307 Health informatics - Interoperability and compatibility in messaging and communication standards - Key characteristics. 2001. Disponible en [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=33396](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=33396) (último acceso, abril, 2013).
- [93] ISO. ISO 8601 Data elements and interchange formats - Information interchange - Representation of dates and times. 2004. Disponible en [http://www.iso.org/iso/catalogue\\_detail?csnumber=40874](http://www.iso.org/iso/catalogue_detail?csnumber=40874) (último acceso, abril, 2013).
- [94] ISO. ISO/IEC 25000: Software Engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE. 2005. Disponible en [http://www.iso.org/iso/catalogue\\_detail?csnumber=35683](http://www.iso.org/iso/catalogue_detail?csnumber=35683) (último acceso, abril, 2013).
- [95] ISO. ISO/TR 20514 Health informatics - Electronic health record - Definition, scope and context. 2005. Disponible en [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=39525](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=39525) (último acceso, abril, 2013).
- [96] ISO. Health informatics - electronic health record communication standard (ISO/EN 13606). Disponible en <http://www.iso.org/iso/>.
- [97] ISO. ISO EN 13606:2 Electronic health record communication - Part 2 : Archetype interchange specification. 2008. Disponible en [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=50119](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=50119) (último acceso, abril, 2013).
- [98] ISO. ISO 18308 Health informatics - Requirements for an electronic health record architecture. 2011. Disponible en [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=52823](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=52823) (último acceso, abril, 2013).

- [99] ISO. ISO 21090 - Health informatics - Harmonized data types for information interchange. 2011. Disponible en [http://www.iso.org/iso/home/store/catalogue\\_tc/catalogue\\_detail.htm?csnumber=35646](http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=35646) (último acceso, abril, 2013).
- [100] ISO. ISO/IEC 25010 Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuARE) - System and software quality models. 2011. Disponible en [http://www.iso.org/iso/home/store/catalogue\\_ics/catalogue\\_detail\\_ics.htm?csnumber=35733](http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=35733) (último acceso, abril, 2013).
- [101] JIMENO-YEPES, A., JIMÁÑEZ-RUIZ, E., BERLANGA-LLAVORI, R. y REBHOLZ-SCHUHMANN, D. Reuse of terminological resources for efficient ontological engineering in life sciences. *BMC Bioinformatics*, vol. 10(Suppl 10), páginas 1–13, 2009.
- [102] KALRA, D., BEALE, T. y HEARD, S. The openehr foundation. *Studies in health technology and informatics*, vol. 115, páginas 153–173, 2005.
- [103] KALRA, D., TAPURIA, A., AUSTIN, T. y DE MOOR, G. Quality requirements for EHR archetypes. *Stud Health Technol Inform*, páginas 48–52, 2012.
- [104] KALRA, D., TAPURIA, A., FRERIKS, G., MENNERAT, F. y DEVLIES, J. Management and maintenance policies for EHR interoperability resources. *Q-REC Project IST*, vol. 27370(3.3), 2008.
- [105] KAZAKOV, Y., KRÖTZSCH, M. y SIMANČÍK, F. ELK: a reasoner for OWL EL ontologies. Informe técnico, Tech. rep., University of Oxford, 2012.
- [106] KERRIGAN, M., MOCAN, A., TANLER, M. y FENSEL, D. The web service modeling toolkit-an integrated development environment for semantic web services. En *The Semantic Web: Research and Applications*, páginas 789–798. Springer, 2007.
- [107] KHAN, W. A., HUSSAIN, M., KHATTAK, A. M., AMIN, M. B. y LEE, S. Saas based interoperability service for semantic mappings among health-care standards. En *8th International Conference on Innovations in Information Technology*. 2012.
- [108] KHAN, W. A., KHATTAK, A. M., LEE, S., HUSSAIN, M., AMIN, B. y LATIF, K. Achieving interoperability among healthcare standards: building semantic mappings at models level. En *Proceedings of the 6th International Conference on Ubiquitous Information Management and Communication*, página 101. ACM, 2012.

- [109] KILIC, O., BICER, V. y DOGAC, A. Mapping Archetypes to OWL. Informe técnico, Software Research and Development Center, Middle East Technical University, Turkey, 2005.
- [110] KILIC, O. y DOGAC, A. Achieving clinical statement interoperability using R-MIM and archetype-based semantic transformations. *IEEE transactions on information technology in biomedicine : a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 13(4), páginas 467–77, 2009. ISSN 1558-0032.
- [111] KLEPPE, A. G., WARMER, J. y BAST, W. *MDA Explained: The Model Driven Architecture: Practice and Promise*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003. ISBN 032119442X.
- [112] KOHLER, M., RINNER, C., HÜBNER-BLODER, G., SABOOR, S., AMMENWERTH, E. y DUFTSCHMID, G. The Archetype-Enabled EHR System ZK-ARCHE - Integrating the ISO/EN 13606 Standard and IHE XDS Profile. *Studies In Health Technology And Informatics*, vol. 169, páginas 799–803, 2011.
- [113] KROG, R., MARKWELL, D., DOLIN, R., DAVERA, G., CHEETHAM, E., HAMM, R., SPACKMAN, K., RECTOR, A., HUFF, S. y RYAN, S. Using SNOMED CT in HL7 Version 3, Implementation Guide, Release 1.5. 2012. Disponible en [http://wiki.hl7.org/index.php?title=Using\\_SNOMED\\_CT\\_in\\_HL7\\_Version\\_3;\\_Implementation\\_Guide,\\_Release\\_1.5](http://wiki.hl7.org/index.php?title=Using_SNOMED_CT_in_HL7_Version_3;_Implementation_Guide,_Release_1.5) (último acceso, abril, 2013).
- [114] KURTEV, I., BÉZIVIN, J. y AKSIT, M. Technological spaces: An initial appraisal. En *CoopIS, DOA'2002 Federated Conferences, Industrial track*. 2002.
- [115] LEE, D., CORNET, R. y LAU, F. Implications of SNOMED CT versioning. *International Journal of Medical Informatics*, vol. 80(6), páginas 442–453, 2011. ISSN 1386-5056.
- [116] LEHMANN, F. *Semantic networks in artificial intelligence*. Elsevier Science Inc., 1992.
- [117] LEZCANO, L., SÁNCHEZ-ALONSO, S. y SICILIA, M.-A. Associating clinical archetypes through UMLS metathesaurus term clusters. *Journal of Medical Systems*, vol. 36(3), páginas 1249–58, 2010. ISSN 0148-5598.
- [118] LEZCANO, L., SICILIA, M.-A. y RODRÍGUEZ-SOLANO, C. Integrating reasoning and clinical archetypes using OWL ontologies and SWRL rules. *Journal of biomedical informatics*, vol. 44(2), páginas 343–53, 2011. ISSN 1532-0480.

- [119] VAN DER LINDEN, H., AUTIN, T. y TALMON, J. Generic screen representations for future-proof systems, is it possible?: There is more to a GUI than meets the eye. *Computer Methods and Programs in Biomedicine*, vol. 95(3), páginas 213 – 226, 2009.
- [120] VAN DER LINDEN, H., TALMON, J., TANGE, H., GRIMSON, J. y HASMAN, A. Proper revisited. *International Journal of Medical Informatics*, vol. 74(2), páginas 235–244, 2005.
- [121] MAEDCHE, A., MOTIK, B., SILVA, N. y VOLZ, R. Mafra - a mapping framework for distributed ontologies. En *Knowledge engineering and knowledge management: ontologies and the semantic web*, páginas 235–250. Springer, 2002.
- [122] MALDONADO, J. *Historia Clínica Electrónica Federada Basada en la Norma Europea CEN/TC251 EN13606*. Tesis Doctoral, Universidad Politécnica de Valencia, 2005.
- [123] MALDONADO, J., MONER, D., BOSCA, D., FERNÁNDEZ-BREIS, J., ANGULO, C. y ROBLES, M. LinkEHR-Ed: A multi-reference model archetype editor based on formal semantics. *International Journal of Medical Informatics*, vol. 78(8), páginas 559–570, 2009.
- [124] MANOLA, F., MILLER, E. y MCBRIDE, B. RDF primer - W3C recommendation. 2004. Disponible en <http://www.w3.org/TR/rdf-primer/> (último acceso, abril, 2013).
- [125] MARTÍNEZ-COSTA, C. *Modelos de representación y transformación para la interoperabilidad semántica entre estándares de Historia Clínica Electrónica basados en arquitectura de modelo dual*. Tesis Doctoral, Facultad de Informática - Universidad de Murcia, 2011.
- [126] MARTÍNEZ-COSTA, C. Repositorio de Poseacle Converter. 2013. Disponible en <http://miuras.inf.um.es/~researchehr/websvn/listing.php?repname=poseacle> (último acceso, abril, 2013).
- [127] MARTÍNEZ-COSTA, C., MIÑARRO GIMÉNEZ, A., MENÁRGUEZ-TORTOSA, M., VALENCIA-GARCIA, R. y FERNÁNDEZ-BREIS, J. T. Flexible semantic querying of clinical archetypes. *Knowledge based and intelligent information and engineering systems*, páginas 597–606, 2010.
- [128] MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M. y FERNÁNDEZ-BREIS, J. T. An approach for the semantic interoperability of ISO EN 13606 and OpenEHR archetypes. *Journal of Biomedical Informatics*, vol. 43(5), páginas 736–746, 2010.



- [129] MARTÍNEZ COSTA, C., MENÁRGUEZ-TORTOSA, M. y FERNÁNDEZ-BREIS, J. T. Clinical data interoperability based on archetype transformation. *Journal of Biomedical Informatics*, vol. 44(5), páginas 869–80, 2011.
- [130] MARTÍNEZ-COSTA, C., MENÁRGUEZ-TORTOSA, M., FERNÁNDEZ-BREIS, J. T. y MALDONADO, J. A. A model-driven approach for representing clinical archetypes for Semantic Web environments. *Journal of Biomedical Informatics*, vol. 42(1), páginas 150–164, 2009.
- [131] MATNEY, S. A., WARREN, J. J., EVANS, J. L., KIM, T. Y., COENEN, A. y AULD, V. A. Development of the nursing problem list subset of SNOMED CT. *Journal of Biomedical Informatics*, vol. 45(4), páginas 683–688, 2012.
- [132] McDONALD, C., HUFF, S., SUICO, J., HILL, G., LEAVELLE, D., ALLER, R., FORREY, A., MERCER, K., DEMOOR, G., HOOK, J. ET AL. LOINC, a universal standard for identifying laboratory observations: a 5-year update. *Clinical chemistry*, vol. 49(4), páginas 624–633, 2003.
- [133] MCGUINNESS, D. L., VAN HARMELLEN, F. ET AL. OWL web ontology language overview - W3C recommendation. 2004. Disponible en <http://www.w3.org/TR/owl-features/> (último acceso, abril, 2013).
- [134] MEIZOSO GARCÍA, M., IGLESIAS ALLONES, J. L., MARTÍNEZ HERNÁNDEZ, D. y TABOADA IGLESIAS, M. J. Semantic similarity-based alignment between clinical archetypes and SNOMED CT: An application to observations. *International journal of medical informatics*, vol. 81(8), páginas 566–78, 2012.
- [135] MICROSOFT. Microsoft Health Common User Interface. 2013. Disponible en <http://www.mscai.net/> (último acceso, abril, 2013).
- [136] MILLER, G. A. Wordnet: a lexical database for english. *Communications of the ACM*, vol. 38(11), páginas 39–41, 1995.
- [137] MOCAN, A., CIMPIAN, E. y KERRIGAN, M. Formal model for ontology mapping creation. En *The Semantic Web-ISWC 2006*, páginas 459–472. Springer, 2006.
- [138] MOTIK, B., GRAU, B. C., HORROCKS, I., WU, Z., FOKOUE, A. y LUTZ, C. OWL 2 Web Ontology Language: Profiles (Second Edition) - W3C recommendation. 2012. Disponible en <http://www.w3.org/TR/owl2-profiles/> (último acceso, abril, 2013).
- [139] MOTIK, B. y STUDER, R. Kaon2—a scalable reasoning tool for the semantic web. En *Proceedings of the 2nd European Semantic Web Conference (ESWC05), Heraklion, Greece*, vol. 17. 2005.

- [140] NOY, N. F., MCGUINNESS, D. L. ET AL. *Ontology development 101: A guide to creating your first ontology*. 2001. Disponible en [http://protege.stanford.edu/publications/ontology\\_development/ontology101-noy-mcguinness.html](http://protege.stanford.edu/publications/ontology_development/ontology101-noy-mcguinness.html) (último acceso, abril, 2013).
- [141] NOY, N. F. y MUSEN, M. A. The prompt suite: interactive tools for ontology merging and mapping. *International Journal of Human-Computer Studies*, vol. 59(6), páginas 983–1024, 2003.
- [142] NOY, N. F., SINTEK, M., DECKER, S., CRUBÉZY, M., FERGERSON, R. W. y MUSEN, M. A. Creating semantic web contents with protege-2000. *Intelligent Systems, IEEE*, vol. 16(2), páginas 60–71, 2001.
- [143] OCEAN INFORMATICS. *Template designer*. 2013. Disponible en <http://wiki.oceaninformatics.com/confluence/display/TTL/Template+Designer> (último acceso, abril, 2013).
- [144] OLDEVIK, J. MOFScript eclipse plug-in: metamodel-based code generation. En *Eclipse Technology Workshop (EtX) at ECOOP*, vol. 2006. 2006.
- [145] OMG. *Clinical observation access service (coas)*. 2001. Disponible en <http://www.omg.org/spec/COAS/1.0/> (último acceso, abril, 2013).
- [146] OMG. *MDA Guide Version 1*. 2003. Disponible en <http://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf> (último acceso, abril, 2013).
- [147] OMG. *Meta Object Facility (MOF) 2.0 Query/View/Transformation, V1.0*. 2008. Disponible en <http://www.omg.org/spec/QVT/1.0/PDF/> (último acceso, abril, 2013).
- [148] OMG. *Mof model to text transformation language*. 2008. Disponible en <http://www.omg.org/spec/MOFM2T/1.0/PDF> (último acceso, abril, 2013).
- [149] OMG. *Ontology Definition Metamodel*. 2009. Disponible en <http://www.omg.org/spec/ODM/1.0/> (último acceso, abril, 2013).
- [150] OMG. *Meta Object Facility (MOF) 2.4.1*. 2011. Disponible en <http://www.omg.org/spec/MOF/2.4.1> (último acceso, abril, 2013).
- [151] OMG. *MOF 2 XMI Mapping*. 2011. Disponible en <http://www.omg.org/spec/XMI/> (último acceso, 2013).
- [152] OPENEHR FOUNDATION. *OpenEHR specifications*. 2013. Disponible en <http://www.openehr.org> (último acceso, abril, 2013).

- [153] OPEREFFA. Opereffa: openehr reference framework and application. 2009. Disponible en <http://www.openehr.org/wiki/display/projects/Opereffa+Project> (último acceso, abril, 2013).
- [154] ORGANIZATION, W. H. International classification of diseases. 2013. Disponible en <http://www.who.int/classifications/icd/en/> (último acceso, abril, 2013).
- [155] ORGUN, B. y VU, J. HL7 ontology and mobile agents for interoperability in heterogeneous medical information systems. *Computers in Biology and Medicine*, vol. 36(7), páginas 817–836, 2006.
- [156] PARSIA, C. . Pellet Integrity Constraints: validating RDF with OWL. 2011. Disponible en <http://clarkparsia.com/pellet/icv/> (último acceso, abril, 2013).
- [157] PAZOS, P. Ehrgen: Generador de sistemas normalizados de historia clínica electrónica basados en openehr. En *3er Congreso Argentino de Informática y Salud*. 2012.
- [158] PINTO, H. S., STAAB, S. y TEMPICH, C. Diligent: Towards a fine-grained methodology for distributed, loosely-controlled and evolving engineering of ontologies. En *ECAI*, vol. 16, página 393. Citeseer, 2004.
- [159] PRESUTTI, V. y GANGEMI, A. Content ontology design patterns as practical building blocks for web ontologies. En *Conceptual Modeling-ER 2008*, páginas 128–141. Springer, 2008.
- [160] PRUD HOMMEAUX, E., SEABORNE, A. ET AL. SPARQL query language for RDF - W3C recommendation. 2008. Disponible en <http://www.w3.org/TR/rdf-sparql-query/> (último acceso, abril, 2013).
- [161] RECTOR, A. Barriers, approaches and research priorities for integrating biomedical ontologies. Disponible en *SemanticHEALTHD6.1, EC-supportedFP6*, [www.semantichealth.org](http://www.semantichealth.org).
- [162] RECTOR, A., QAMAR, R. y MARLEY, T. Binding ontologies and coding systems to electronic health records and messages. *Applied Ontology*, vol. 4(1), páginas 51–69, 2009.
- [163] RECTOR, A., ROGERS, J., ZANSTRA, P. y VAN DER HARING, E. Opengalen: open source medical terminology and tools. En *AMIA Annual Symposium Proceedings*, vol. 2003, página 982. American Medical Informatics Association, 2003.
- [164] ROMÁN, I., ROA, L., REINA-TOSINA, J. y MADINABEITIA, G. Demographic management in a federated healthcare environment. *International journal of medical informatics*, vol. 75(9), páginas 671–682, 2006.

- [165] ROSENBLOOM, S. T., MILLER, R. A., JOHNSON, K. B., ELKIN, P. L. y BROWN, S. H. Interface terminologies facilitating direct entry of clinical data into electronic health record systems. *Journal of the American Medical Informatics Association*, vol. 13(3), páginas 277–288, 2006.
- [166] RUMBAUGH, J., JACOBSON, I. y BOOCH, G. *Unified Modeling Language Reference Manual, The*. Pearson Higher Education, 2004.
- [167] SAHAY, R., FOX, R., ZIMMERMANN, A., POLLERES, A. y HAUSWRITH, M. A methodological approach for ontologising and aligning health level seven (HL7) applications. *Availability, Reliability and Security for Business, Enterprise and Health Information Systems*, páginas 102–117, 2011.
- [168] SÁNCHEZ-CUADRADO, J., GARCÍA-MOLINA, J. y MENÁRGUEZ-TORTOSA, M. RubyTL:: A practical, extensible transformation language. En *Model Driven Architecture—Foundations and Applications*, páginas 158–172. Springer, 2006.
- [169] SARI, A. K., RAHAYU, W. y BHATT, M. Archetype sub-ontology: Improving constraint-based clinical knowledge model in electronic health records. *Knowledge-Based Systems*, vol. 26(August), páginas 75–85, 2012. ISSN 09507051.
- [170] SCHADOW, G. y McDONALD, C. J. The unified code for units of measure. *The Regenstrief Institute For Health Care*, 2009.
- [171] SCHARFFE, F. y DE BRUIJN, J. A language to specify mappings between ontologies. En *Proc. of the Internet Based Systems IEEE Conference (SITIS05)*. Citeseer, 2005.
- [172] SCHULER, T., GARDE, S., HEARD, S. y BEALE, T. Towards automatically generating graphical user interfaces from openEHR archetypes. *Stud Health Technol Inform*, vol. 124, páginas 221–226, 2006. ISSN 0926-9630.
- [173] SCHULZ, S., SCHOBER, D., DANIEL, C. y JAULENT, M.-C. Bridging the semantics gap between terminologies, ontologies, and information models. *Stud Health Technol Inform*, vol. 160(Pt 2), páginas 1000–4, 2010.
- [174] SHADBOLT, N., HALL, W. y BERNERS-LEE, T. The semantic web revisited. *Intelligent Systems, IEEE*, vol. 21(3), páginas 96–101, 2006.
- [175] SHEARER, R., MOTIK, B. y HORROCKS, I. HermiT: A highly-efficient OWL reasoner. En *Proceedings of the 5th International Workshop on OWL: Experiences and Directions (OWLED 2008)*, páginas 26–27. 2008.

- [176] SIRIN, E., PARSIA, B., GRAU, B., KALYANPUR, A. y KATZ, Y. Pellet: A practical OWL-DL reasoner. *Web Semantics: science, services and agents on the World Wide Web*, vol. 5(2), páginas 51–53, 2007.
- [177] SPACKMAN, K. An examination of OWL and the requirements of a large health care terminology. En *Proceedings of the OWL: Experiences and Directions Third International Workshop (OWLED 2007)*, CEURWS. 2007.
- [178] SPACKMAN, K. A. y CAMPBELL, K. E. SNOMED RT: a reference terminology for health care. En *Proceedings of the AMIA annual fall symposium*, página 640. American Medical Informatics Association, 1997.
- [179] STAAB, S., STUDER, R., SCHNURR, H.-P. y SURE, Y. Knowledge processes and ontologies. *Intelligent Systems, IEEE*, vol. 16(1), páginas 26–34, 2001.
- [180] STAHL, T., VOELTER, M. y CZARNECKI, K. *Model-Driven Software Development: Technology, Engineering, Management*. John Wiley & Sons, 2006. ISBN 0470025700.
- [181] STARKE, S. *IHE - Integrating the Healthcare Enterprise: Realization of the Retrieve Information for Display Profile Shown Upon the Clinical Multimedia Archive DOMAIN. Design, Implementation, Perspectives*. VDM Verlag, 2008.
- [182] STEINBERG, D., BUDINSKY, F., MERKS, E. y PATERNOSTRO, M. *EMF: eclipse modeling framework*. Addison-Wesley Professional, 2008.
- [183] STROETMANN, V. N., KALRA, D., LEWALLE, P., RECTOR, A., RODRIGUES, J. M., STROETMANN, K. A., SURJAN, G., USTUN, B., VIRTANEN, M. y ZANSTRA, P. E. Semantic Interoperability for Better Health and Safer Healthcare. Deployment and Research Roadmap for Europe. Disponible en ISBN-13 : 978-92-79-11139-6.
- [184] SUN MICROSYSTEMS. Java Architecture for XML Binding (JAXB) 2.0. 2004. Disponible en <http://java.sun.com/xml/jaxb/> (último acceso, abril, 2013).
- [185] TAO, C., JIANG, G., ONIKI, T. A., FREIMUTH, R. R., ZHU, Q., SHARMA, D., PATHAK, J., HUFF, S. M. y CHUTE, C. G. 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*, vol. Article in Press DOI: doi: 10.1136/amiajnl-2012-001326, 2012.

- [186] TAO, J., SIRIN, E., BAO, J. y MCGUINNESS, D. L. Integrity constraints in OWL. En *AAAI2010: Proceedings of the AAI Conference on Artificial Intelligence*. 2010.
- [187] THE MOZILLA DEVELOPER FOUNDATION. XML user interface language (XUL). 2013. Disponible en <https://developer.mozilla.org/en-US/docs/XUL> (último acceso, abril, 2013).
- [188] THE OPENEHR FOUNDATION. openEHR to ISO 13606-1, ISO 21090 mapping. 2012. Disponible en <http://www.openehr.org/wiki/display/stds/openEHR+to+ISO+13606-1,+ISO+21090+mapping> (último acceso, abril, 2013).
- [189] THE OPENEHR FOUNDATION. OpenEHR Clinical Knowledge Manager. 2013. Disponible en <http://www.openehr.org/knowledge/> (último acceso, abril, 2013).
- [190] TSARKOV, D. y HORROCKS, I. Fact++ description logic reasoner: System description. *Automated Reasoning*, páginas 292–297, 2006.
- [191] TU, S. Health Level Seven (HL7) Data Types and Top-Level Reference Information Model (RIM) Classes. 2003. Disponible en <http://protege.stanford.edu/ontologies/HL7RIM/index.html> (último acceso, abril, 2013).
- [192] W3C. XForms 1.1 - W3C Recommendation. 2009. Disponible en <http://www.w3.org/TR/2009/REC-xforms-20091020/> (último acceso, abril, 2013).
- [193] YU, S., BERRY, D. y BISBAL, J. Clinical coverage of an archetype repository over SNOMED-CT. *Journal of Biomedical Informatics*, vol. Article in Press doi: 10.1016/j.jbi.2011.12.001, 2012.
- [194] ZIEGELER, C. *Cocoon: building XML applications*. Pearson Education, 2002.