**UNIVERSIDAD DE MURCIA**

**Departamento de Ingeniería de la Información y las Comunicaciones**


# TESIS DOCTORAL

# Un Entorno de Integración de Ontologías para el Desarrollo de Sistemas de Gestión de Conocimiento


**Jesualdo Tomás Fernández Breis**


*Marzo 2003*


Directores:

**Dr. Fernando Martín Rubio**

**Dr. Rodrigo Martínez Béjar**

# ÍNDICES

## Indices de contenidos

# ÍNDICE DE CONTENIDOS

Índice de contenidos …… 3

Índice de figuras …….. 11

Índice de tablas …… 15

Introducción …… 17

**CAPÍTULO 1. ESTADO DEL ARTE ……..** 23

1.1 Ontologías …….. 24

1.1.1 Definiciones …… 24

1.1.1.1 Definiciones relacionadas con la Filosofía …….. 25

1.1.1.2 Definiciones relacionadas con la Inteligencia Artificial …….. 26

1.1.2 Historia de la Ontología ……... 29

1.1.3 Tipos de Ontologías ……. 33

1.1.3.1 Clasificación por el Conocimiento Contenido ……. 34

1.1.3.2 Clasificación por Motivación ……... 35

1.2 Ontología Formal y Descriptiva ……. 37

1.3 Ingeniería Ontológica ……. 41

1.3.1 Elementos de Ontologías …….. 41

1.3.1.1 Relaciones ……. 42

Taxonomía ……. 42

Diseño y construcción de taxonomías ……... 46

Mereología ……. 48

1.3.1.2 Otra Clasificación de Alternativas ……... 54

1.3.2 Ejemplos de Ontologías ……. 55

1.3.3 Criterios para Diseñar Ontologías ……. 58

1.3.4 Metodologías para Construir Ontologías …….. 61

1.3.4.1 Metodologías para Construir Ontologías a partir de Cero …… 61

CYC ……... 62

Uschold y King …….. 62

Grüninger y Fox …… 63

Kactus …… 64

Methontology …… 64

Sensus …… 65

On-To-Knowledge ……. 66

Terminae …… 67

1.3.4.2 Metodologías para Reingeniería Ontológica ……. 68

1.3.5 Evaluación de Ontologías …….. 68

1.3.6 Papel de las Ontologías en el Desarrollo de Sistemas de Información ……. 70

1.3.7 Uso e Importancia de las Ontologías ……. 72

1.4 Desarrollo Colaborativo de Ontologías ……. 73

1.4.1 Sistemas de Integración de Ontologías ……. 76

1.4.1.1 PROMPT ……... 78

1.4.1.2 FCA-Merge ……... 80

1.4.1.3 Chimaera ……... 81

1.4.2 Construcción Cooperativa de Ontologías …….. 82

1.4.2.1 Ontolingua Server ……. 84

1.4.2.2 WebOnto …….. 84

1.4.2.3 Ontosaurus …… 85

1.5 Lenguajes Ontológicos …….. 86

1.5.1 SHOE …… 86

1.5.2 RDF(s) ……... 88

1.5.3 OML ……. 89

1.5.4 XOL …….. 90

1.5.5 OIL ……... 94

1.5.6 DAML+OIL ……. 96

1.5.7 Comentarios Generales ……. 98

1.6 Gestión de Conocimiento ……... 99

1.6.1 Definiendo Conocimiento …… 99

1.6.2 Tipos de Conocimiento …… 100

1.6.3 Principios de Conocimiento ……. 103

1.6.4 Gestión de Conocimiento como Disciplina ……. 103

1.6.5 Tecnologías del Conocimiento ……. 106

1.6.6 Gestión de Conocimiento desde una Perspectiva Tecnológica …… 110

1.6.6.1 Infraestructura TIC ……... 110

1.6.6.2 Esquemas de Representación de Conocimiento …… 111

1.6.6.3 Repositorios de Conocimiento ……. 112

1.6.6.4 *Integrated Performances Support Systems (IPSS)* …….. 112

1.6.6.5 Transformación de Conocimiento ……. 113

1.6.7 Sistemas de Gestión de Conocimiento ……. 114

1.7 Problema a Resolver en este Trabajo …… 115

1.8 Resumen ……. 119

**CAPÍTULO 2. FORMALIZACIÓN DEL MODELO ONTOLÓGICO ……..** 123

2.1 Supuestos Básicos …….. 124

2.2 Funciones Ontológicas ……... 126

2.3 Una Forma Complementaria de Definir Ontologías …….. 136

2.4 Ontologías como Sistemas de Información …… 140

2.5 Un Lenguaje Sencillo para Especificar Ontologías …… 144

2.5.1 Ejemplo ……. 146

2.5.2 El Lenguaje de Definición XML …… 149

2.6 Resumen ……. 150

**CAPÍTULO 3. UN ENTORNO COOPERATIVO PARA LA INTEGRACIÓN DE ONTOLOGÍAS ……** 153

3.1 Cooperación y sus Problemas …… 154

3.2 El Entorno para Integrar Conocimiento ……. 157

3.2.1 Principios de Integración …….. 163

3.2.2 Un Entorno para Integrar Ontologías …… 165

3.2.3 Usando Ontologías Derivadas de la Integración ……. 182

3.3 Una Visión Algorítmica de los Procesos de Integración …… 195

3.3.1 Algoritmo General de Integración de Ontologías ……. 195

3.3.2 Select_Ontologies ……. 196

3.3.3 Ontological_Instantiation ……. 197

3.3.4 Ontological_Transformation ……. 199

3.4 Comparación con otros Entornos para la Construcción Cooperativa de Conocimiento …….. 200

3.5 Resumen …… 204

**CAPÍTULO 4. UN EJEMPLO COMPLETO DE INTEGRACIÓN ……** 207

4.1 El Dominio ……. 208

4.2 Ontologías Fuente …….. 209

4.2.1 Integración …… 211

4.2.1.1 Selección de Ontologías ……. 212

4.2.1.2 Inicialización de la Integración ……...……. 214

4.2.1.3 Instanciación de Ontologías ……...…….. 215

4.2.1.4 Transformación de Ontologías ……. 218

4.3 Diferentes Puntos de Vista para el Ejemplo ……... 227

4.3.1 Usando $O_A(t)$ y $O_D(t)$ como Referencia …… 228

4.3.2 Usando $O_C(t)$ como Referencia ……. 229

4.4 Resumen …… 230

**CAPÍTULO 5. VALIDACION DEL ENTORNO DE INTEGRACION ……..** 231

5.1 Validación del Entorno de Integración …….. 232

5.1.1 Un Método para Medir Ganancias de Conocimiento …….. 232

5.1.2 Ejemplo ……. 235

5.1.3 Experimentos de Integración ……. 237

5.1.3.1 Experimento 1 ……... 237

5.1.3.2 Experimento 2 ……... 240

5.2 Validación del Proceso de Construcción de Ontologías …… 243

5.2.1 Resultados Ontológicos ……. 246

5.3 Aplicaciones ……... 249

5.3.1 Construcción de Ontologías en Biología Molecular ……. 250

5.3.1.1 Motivación ……. 250

5.3.1.2 Descripción …… 251

5.3.2 Gestión de Conocimiento en Unidades de Cuidados Intensivos ……... 262

5.3.2.1 Motivación ……. 262

5.3.2.2 Descripción del Problema ……. 263

5.3.2.3 Descripción del Entorno …… 265

Gestión Clínica de Eventos ……... 265

Otras Características …….. 268

5.3.2.4 Un Modelo Ontológico de una Unidad de Cuidados Intensivos ….. 269

5.3.2.5 Una Extensión Temporal al Modelo Ontológico …….. 272

5.3.2.6 Representación Ontológica de Ciclos …… 276

5.3.2.7 Consistencia basada en Ontologías ……... 277

5.4 Resumen ……. 282

**CAPITULO 6. UNA APLICACIÓN SOFTWARE PARA FACILITAR LA INTEGRACIÓN Y GESTION DE CONOCIMIENTO ……** 285

6.1 El Sistema Cliente-Servidor ……... 287

6.1.1 Arquitectura General ……. 287

El Servidor de Ontologías ……. 288

Cliente ……... 288

6.1.2 Especificación de Ontologías …… 290

6.2 Una Sesión con ONTOINT …… 291

6.2.1 Gestión de Usuarios …….. 292

6.2.2 Usuarios Expertos y Normales …….. 297

6.3 Interfaz Web ……... 303

6.3.1Registro como Usuarios ……. 305

6.3.2 Accediendo al Sistemas ……. 305

6.3.3 Opciones Web para el Administrador ……... 306

6.3.4 Opciones Web para Usuarios Normales …… 309

6.3.4.1 Personalización Terminológica ……. 311

6.3.4.2 Explorando la Ontologías …….. 313

6.3.5 Opciones Web para Usuarios Expertos ……. 317

6.4 Comparación con otras Herramientas de Ingeniería Ontológica …… 319

6.4.1 Propiedades Ontológicas ……... 320

6.4.2 Facilidades de Cooperación ……... 321

6.4.2.1 Ontolingua ……. 321

6.4.2.2 WebOnto ……... 321

6.4.2.3 OntoSaurus …… 322

6.5 Resumen ……. 322

**CAPITULO 7. CONCLUSIONES Y LINEAS FUTURAS** 325

**CAPÍTULO 8. RESUMEN EN INGLES/SUMMARY IN ENGLISH** 331

**REFERENCIAS** 349

**APENDICE. EJEMPLOS DE FORMALIZACIÓN DE ONTOLOGÍAS** 363

# INDICE DE FIGURAS

Figura 1.1 Primer libro titulado Ontología…… 31

Figura 1.2 Filósofos Ontólogos Modernos…… 40

Figura 2.1 Ontología de referencia …….. 126

Figura 2.2 Un ejemplo de ontología parcial ……. 146

Figura 3.1 Inconsistencia Interna ……. 156

Figura 3.2 Inconsistencia Externa ……...……. 156

Figura 3.3. Ontología del experto A ……. 161

Figura 3.4. Ontología del experto B ……. 161

Figura 3.5. Ontología del experto C ……. 162

Figura 3.6. Ontología del experto D ……. 162

Figura 3.7. Subontología de la ontología B …….. 176

Figura 3.8. Subontología de la ontología C …….. 176

Figura 3.9. Ontología derivada de la integración ……. 182

Figura 3.10. Ontología instanciada e integrada …… 186

Figura 3.11.La ontología transformada y derivada de la integración ……... 194

Figura 4.1 Ontología A ……. 209

Figura 4.2 Ontología B ……. 210

Figura 4.3 Ontología C ……. 210

Figura 4.4 Ontología D ……. 211

Figura 4.5 Ontología derivada de la integración ……. 216

Figura 4.6 Ontología derivada de la integración e instanciada …… 217

Figura 4.7 PO(t) tras incluir $Book_D$ …….. 220

Figura 4.8 PO(t) tras incluir $Novel_D$ ……. 221

Figura 4.9 PO(t) tras incluir $Poetry_D$ …… 222

Figura 4.10 PO(t) tras incluir $Security_D$ ……... 223

Figura 4.11 PO(t) al incluir $Admin_D$ ……. 224

Figura 4.12 PO(t) tras incluir $novel_A$ …… 226

Figura 4.13 Ontología derivada de la integración, transformada y final …….. 227

Figura 4.14 Ontología global, final usando $O_A(t)$ y $O_D(t)$ como referencia ……. 228

Figura 4.15 Ontología final, global usando $O_C(t)$ como referencia ……. 229

Figura 5.1. Representación gráfica de los resultados ontológicos ……... 247

Figura 5.2 Ontología del primer experto …….. 252

Figura 5.3 Ontología del segundo experto ……... 252

Figura 5.4 Ontología derivada de la integración de la célula ……... 253

Figura 5.5 La ontología transformada y final de la célula …… 255

Figura 5.6. Ontología sobre proteínas del primer experto …… 257

Figura 5.7. La ontología de la proteína construida por el segundo experto ……. 258

Figura 5.8. La ontología final de la proteína …… 260

Figura 5.9. Modelo ontológico parcial de una UCI …….. 269

Figura 5.10. Ontología de la evolución de un paciente …… 270

Figura 5.11. Ontología para más de una UCI …….. 271

Figura 5.12. La ontología una vez incluidos los ciclos …… 280

Figura 6.1 Una visión del proceso de integración …… 289

Figura 6.2 Un editor gráfico para especificar ontologías ……. 290

Figura 6.3 Ventana principal de la aplicación …….. 292

Figura 6.4 Diálogo de registro …….. 293

Figura 6.5 Diálogo de acceso al sistema …….. 294

Figura 6.6 Opciones para gestionar el sistema ……. 294

Figura 6.7 Visión de los detalles de un usuario …… 295

Figura 6.8 Gestión de temas ……. 296

Figura 6.9 Mensaje del Día …….. 297

Figura 6.10. Mensaje del día …… 298

Figura 6.11 Opciones para expertos ……. 298

Figura 6.12 Experto seleccionando tema de trabajo ……. 299

Figura 6.13 Explorando una ontología …… 300

Figura 6.14 Expansión del nodo taxonómico PERSON …….. 301

Figura 6.15 Resultados del proceso de integración ……. 302

Figura 6.16 La página de inicio del integrador de ontologías …….. 304

Figura 6.17 Un usuario web dándose de alta en el sistema …….. 305

Figura 6.18 Página de acceso al sistema …….. 306

Figura 6.19 Opciones de administrador ……... 307

Figura 6.20 Gestión de usuarios …….. 308

Figura 6.21 Gestión de Temas ……. 309

Figura 6.22 Página para usuarios normales ……. 310

Figura 6.23 Resultados de la integración …… 311

Figura 6.24 Cambiar “staff” por “worker” …….. 312

Figura 6.25 Cambio de “worker” por “personal” ……. 313

Figura 6.26 Una ontología parcialmente expandida …… 314

Figura 6.27 Panel de atributos ……. 315

Figura 6.28 Elección del atributo a expandir …….. 316

Figura 6.29 La ontología expandida …… 317

Figura 6.30 Página de inicio para expertos ……. 318

Figura 6.31 La página de inicio en castellano …….……..... 319

# INDICE DE TABLAS

Tabla 2.1 Funciones para una ontología …….. 131

Tabla 2.2 Funciones para un concepto …… 131

Tabla 3.1. Funciones para dos conceptos …… 178

Tabla 3.2. Funciones para conjuntos de conceptos ……. 178

Tabla 3.3. Funciones para dos ontologías ……... 179

Tabla 3.4. Ontologías candidatas compatibles …… 181

Tabla 5.1 Descripción de las ontologías ……... 235

Tabla 5.2 Ganancia de conocimiento para cada dominio …… 239

Tabla 5.3 Ganancia de conocimiento global para cada dominio ……. 240

Tabla 5.4. Ontologías para el primer paso de integración ……... 241

Tabla 5.5. Ontologías para el segundo paso de integración ……. 241

Tabla 5.6 Ganancias medias de conocimiento para las dos iteraciones ……... 242

Tabla 5.7 Valores de impacto …….. 242

Tabla 5.8. Resultados ontológicos ……... 246

Tabla 5.9. Valores para cada categoría ontológica ……. 255

Tabla 5.10. Ganancia de conocimiento asociada a este experimento …….. 256

Tabla 5.11 Atributos de las dos ontologías ……. 259

Tabla 5.12. Elementos ontológicos para cada ontología ……. 261

Tabla 5.13 Ganancia de conocimiento para el dominio de la proteína …… 262

Tabla 5.14. La hipótesis diagnóstica …… 277

Tabla 5.15. La evolución del paciente …… 278

Tabla 5.16. La hipótesis ……... 281

Tabla 5.17. La evolución del paciente ……. 281

# INTRODUCCIÓN

El desarrollo de sistemas que faciliten la gestión de conocimiento es un elemento estratégico para las organizaciones en la actualidad. La necesidad de encontrar soluciones para obtener el conocimiento necesario para construir este tipo de sistemas ha sido la motivación primordial para esta tesis doctoral. La solución que propondremos en este trabajo se basa en la mejora de los procesos de integración del conocimiento explícito disponible. Con ello, el desarrollo de sistemas para la gestión de conocimiento sería más eficiente. En este trabajo se usará un punto de vista de ingeniería ontológica. Las mejoras que se proponen son debidas a los inconvenientes encontrados en las metodologías disponibles para desarrollo cooperativo de ontologías tales como Ontolingua (Farquhar et al, 1997), WebOnto (Domingue, 1998), Prompt (Fridman-Noy and Musen, 2000) o FCA-Merge (Stumme and Maedche, 2001).

Para lograr este objetivo, se seguirá la siguiente metodología:

- Análisis del estado del arte en Ingeniería Ontológica. Esto implica estudiar las diferentes metodologías disponibles relacionadas con las diferentes actividades contempladas en el ciclo de vida ontológico, tales como diseño, creación o evaluación. El estudio de los lenguajes actuales para representar ontologías también está incluido en este análisis.

- Análisis detallado de las metodologías actuales para la construcción cooperativa de conocimiento. Dicho análisis se ha basado en factores tales como el modelo de conocimiento soportado por cada metodología o el grado de participación de expertos durante el proceso.

- Definición y formalización de un entorno para la integración de ontologías. Se ha dividido el proceso en dos partes: (1) definición y formalización de un modelo ontológico; (2) definición y formalización de una metodología para integrar ontologías especificadas según el modelo ontológico anterior. La participación de expertos del

dominio en los procesos de integración ha sido limitada a proporcionar una ontología privada, puesto que las ontologías son integradas automáticamente. Además, los resultados de los procesos de integración han sido adaptados a cada experto en particular.

- Diseño e implementación de una aplicación software para el desarollo cooperativo de ontologías. Se ha desarrollado una interfaz web para facilitar el uso de la aplicación usando navegadores web. En este sistema, los expertos no tienen permiso para acceder al conocimiento de otros expertos excepto a través de los resultados de procesos de integración de ontologías.

Las distintas secciones del entorno formal para integrar conocimiento han sido validadas. La metodología de validación se puede resumir a continuación: (1) validación de la metodología de construcción de ontologías; (2) evaluación de la calidad de las ontologías obtenidas a través de los procesos de integración; (3) evaluación de la utilidad de los procesos de integración.

Las cuestiones anteriores se han realizado con éxito, y los resultados obtenidos se presentan en esta tesis. La organización de este trabajo puede ser descrita como sigue. En el capítulo 1 se tratan diferentes aspectos relacionados con ingeniería ontológica y gestión de conocimiento. En primer lugar se realizan una serie de consideraciones generales sobre ontologías. Tras una discusión acerca del significado del término ontologías se hace un breve repaso a la evolución histórica del concepto de ontología. A continuación se describen aspectos importantes relacionados con ingeniería ontológica. Algunos ejemplos de los aspectos tratados en este capítulo son: tipos de ontologías, elementos de ontologías, uso y rol de las ontologías o ejemplos de ontologías. Asimismo, otros aspectos relacionados con el ciclo de vida ontológico han sido incluidos en este capítulo. Concretamente se han comentado aspectos tales como criterios para diseñar ontologías, metodologías para la construcción de ontologías (bien partiendo desde cero o a través de reingeniería), y evaluación de ontologías.

Además, se han analizado diferentes metodologías existentes para la construcción cooperativa de ontologías, bien desarrollando una ontología común, bien integrando ontologías

previamente existentes. Finalmente, este capítulo contiene una visión general sobre gestión de conocimiento, puesto que la finalidad de esta tesis es facilitar el desarrollo de sistemas para la gestión de conocimiento.

En el capítulo 2 se presenta y se formaliza el modelo ontológico diseñado, definido y propuesto en este trabajo. Este modelo se usará posteriormente para representar ontologías. El modelo ontológico se compone de los siguientes elementos:

- conceptos, que son las entidades más relevantes de un dominio.

- atributos, que poseen una doble naturaleza: específicos y heredados. Los atributos específicos son aquellos atributos definidos para un concepto en particular. Por otra parte, los atributos heredados son aquellos que un concepto hereda a través de sus relaciones con otros conceptos del dominio.

- relaciones, que estructuran el dominio en cuestión. En este modelo, se especifican dos tipos de relaciones, taxonomía y mereología, que son los dos tipos de relaciones interconceptuales más importantes. Ambos tipos de relaciones establecen relaciones jerárquicas entre los conceptos que intervienen en ellas.

- axiomas, que son verdades que se afirman en un dominio. Concretamente, este modelo cubre axiomas estructurales, que son los axiomas que se pueden extraer directamente de la estructura de la ontología, tal como "A es un concepto", "B es un atributo del concepto A", "el concepto A es un tipo de concepto C", o "el concepto C es una parte del concepto D". Por su parte, los axiomas no estructurales, que son el resto de axiomas (p.ej., F=m*a para sólidos), no están incluidos en el modelo.

Finalmente, en este capítulo se demuestra que el modelo ontológico usado es un sistema de información de acuerdo a la teoría de dominios de Scott (Winskel, 1994). Esta propiedad es bastante interesante para tareas de explotación del modelo ontológico.

El marco de integración ontológica se describe y formaliza en el capítulo 3. En primer lugar se señalan los problemas más comunes derivados de la naturaleza cooperativa de los procesos de integración. A continuación se formulan los principios de integración usados en este trabajo. Tras proceder a formalizar el marco de integración, se describe el proceso de integración desde un punto de vista algorítmico. Finalmente, se realiza una comparación con otros entornos de integración de conocimiento. En la práctica, los capítulos 2 y 3 representan el núcleo teórico-formal de este trabajo.

Un ejemplo práctico de integración de ontologías se muestra en el capítulo 4. Posteriormente, los resultados obtenidos a través de diferentes experimentos son analizados en el capítulo 5, cuyo objetivo es la validación del marco de integración ontológica propuesto en esta tesis. Concretamente se han validado los siguientes aspectos: (1) el proceso de construcción de ontologías; (2) el entorno de integración de ontologías; (3) la calidad de las ontologías construidas.

La aplicación software desarrollada en esta tesis se describe en el capítulo 6. Esta aplicación se divide en dos partes, una para usuarios web y otra para usuarios no web. Se describen diferentes aspectos de la herramienta tales como el tipo de usuarios o cómo especificar ontologías por medio de su uso. Asimismo, se muestra una sesión de trabajo con la herramienta para cada tipo de usuario. Finalmente, se presentan algunas conclusiones finales así como líneas futuras de trabajo para mejorar los resultados obtenidos a través de esta tesis.

# CAPÍTULO 1

# ESTADO DEL ARTE

# 1.1 ONTOLOGÍAS

## 1.1.1 Definiciones

La noción de ontología ha recibido múltiples definiciones a lo largo de la historia. En este apartado presentaremos diferentes definiciones propuestas por individuos y estamentos de diferentes áreas. Comenzaremos esta enumeración de definiciones por la exploración de diferentes diccionarios. El primer diccionario consultado fue el "Webster's Third New International Dictionary". En él, podemos encontrar dos definiciones para el término ontología. La primera de ellas tiene una naturaleza filosófica, mientras que la segunda tiene un significado más terminológico, y esta última es la definición más usada en Inteligencia Artificial y Representación del Conocimiento.

> *"Ciencia o estudio del ser: específicamente, una rama de la metafísica relacionada con la naturaleza y las relaciones del ser; un sistema particular según el cual se investigan los problemas de la naturaleza del ser; esto es, filosofía fundamental".*

> *"Teoría relativa a los tipos de entidades y específicamente los tipos de entidades abstractas que se admiten en el lenguaje de un sistema"*

A continuación se consultó la vigésimo segunda edición del Diccionario de la Real Academia de la Lengua Española. En este caso sólo encontramos una definición, similar a la primera definición del Webster.

> *"Parte de la metafísica relacionada con el ser en general y sus propiedades esenciales"*

Estas definiciones relacionan el concepto de ontología con la metafísica y la filosofía. Por lo tanto, sería interesante estudiar algunas definiciones de ontología realizadas por filósofos.

### *1.1.1.1 Definiciones relacionadas con la Filosofía*

La "Oxford Companion of Philosophy" define ontología de la siguiente forma:

> *"Ontología, entendida como una rama de la metafísica, es la ciencia del ser en general, abarcando aspectos como la naturaleza de la existencia y la estructura categórica de la realidad. El término ontología tiene algunos usos adicionales en filosofía. En un sentido derivativo se usa para referirse a un conjunto de cosas cuya existencia queda reconocida por una teoría o sistema de pensamiento. En este sentido se habla de la ontología de una teoría o de un sistema metafísico definido por tal ontología"*

En las siguientes líneas se presentan diferentes definiciones propuestas por importantes filósofos.

> *"Ontología es la ciencia de algo y de nada, del ser y del no ser, de la cosa y del modo de la cosa, de la sustancia y el accidente" (Leibniz) (Couturat, 1903)*

> *"La filosofía trascendental es el sistema de todas nuestras cogniciones puras a priori, que podemos llamar ontología. Así, ontología trata con cosas en general, desde abstractas hasta particulares. Abarca todos los conceptos puros de la comprensión y todos los principios de la razón. Las ciencias principales que pertenecen a la metafísica son: ontología, cosmología, y teología. Ontología es una pura doctrina de elemento de toda nuestra cognición al completo, o: contiene la suma de todos nuestros conceptos puros que podemos tener a priori sobre la cosas (Kant, 2001)*

> *"La gente trata con asuntos relacionados con la teoría de entidades desde la antigüedad bajo el título de 'Metafísica' y, especialmente, bajo el título de 'Ontología' como parte de la metafísica; y ellos no han fallado siempre a reconocer las características de la libertad de existencia". (Meinong, 1921)*

### *1.1.1.2 Definiciones relacionadas con Inteligencia Artificial*

Las definiciones de ontología en Inteligencia Artificial (IA) son similares a la interpretación del filósofo Quine (Quine, 1961), quien dijo que todo lo que puede ser cuantificado existe. La primera definición de ontología en Inteligencia Artificial apareció en (Neches et al, 1991):

> *"Una ontología define los términos básicos y relaciones que conforman el vocabulario de un área específica, así como las reglas para combinar dichos términos y las relaciones para definir extensiones de vocabularios"*

Una de las definiciones más extendidas es la dada por Tom Gruber (Gruber, 1993):

> *"Una ontología es una especificación explícita de una conceptualización. El término proviene de la filosofía, donde una ontología es un recuento sistemático de la existencia. En sistemas de Inteligencia Artificial, lo que existe es lo que puede ser representado. Cuando el conocimiento de un dominio se representa mediante un formalismo declarativo, el conjunto de objetos que puede ser representado se llama universo del discurso. Esos conjuntos de objetos, y las relaciones que se establecen entre ellos, son reflejados en un vocabulario con el cual representamos el conocimiento en un sistema basado en conocimiento. Así, en el contexto de IA, podemos describir la ontología de un programa como un conjunto de términos. En tal ontología, las definiciones asocian nombres de entidades del universo del discurso con textos comprensibles por los humanos que describen el significado de los nombres, y axiomas formales que limitan la interpretación y buen uso de dichos términos. Formalmente, una ontología es una teoría lógica"*

Para comprender esta definición se debería explicar qué es una conceptualización. Una conceptualización es una interpretación estructurada de una parte del mundo que usan los seres humanos para pensar y comunicar sobre ella. Para un informático, una conceptualización podría ser la clasificación de sistemas informáticos atendiendo a su naturaleza física en sistemas

hardware, sistemas software y sistemas firmware. Por explícita entendemos que los conceptos y las restricciones se definen de forma explícita.

La definición de Gruber ha recibido varias críticas. Por ejemplo, Nicola Guarino (Guarino, 1995), tras examinar siete posibles interpretaciones de ontologías, afirmó:

> *"Un punto de inicio en este esfuerzo clarificador será el cuidadoso análisis de la interpretación dada por Gruber. El problema principal de dicha interpretación es que se basa en la noción de conceptualización. Una conceptualización es un conjunto de relaciones extensionales que describen un estado particular, mientras que la noción que tenemos en mente es intensional, esto es, algo como una rejilla conceptual al que le imponemos varios posibles estados".*

Sin embargo, ésta no es la única crítica recibida por esta definición, puesto que es considerada como demasiado general. Algunas opiniones afirman que algunas ontologías satisfacen dicha definición pero que no son útiles para desarrollar aplicaciones.

Nicola Guarino propuso una definición alternativa de ontología:

> *"En el sentido filosófico, podemos referirnos a una ontología como un sistema particular de categorías que representa una cierta visión del mundo. Como tal, este sistema no depende de un lenguaje particular: la ontología de Aristóteles es siempre la misma, independientemente del lenguaje usado para describirla. Por otro lado, en su uso más típico en IA, una ontología es un artefacto ingenieril constituido por un vocabulario específico para describir una cierta realidad, más un conjunto de supuestos explícitos concernientes al significado pretendido de las palabras del vocabulario. Este conjunto de supuestos tiene generalmente la forma de teorías lógicas de primer orden, donde las palabras del vocabulario aparecen como predicados unarios o binarios, respectivamente llamados conceptos y relaciones. En el caso más simple, una ontología describe una jerarquía de conceptos relacionados por relaciones de subsunción; en los casos más*

*sofisticados, se añaden axiomas para expresar otras relaciones entre conceptos y restringir la posible interpretación."*

Según este autor, las ontologías usadas en aplicaciones reales son realmente adaptaciones ingenieriles de ontologías, no ontologías propiamente dichas. Así, por ejemplo, siguiendo esta perspectiva, "el peso de una persona es una masa" es una afirmación que podría aparecer en una ontología, mientras que "el peso de una persona es un número" puede aparece en una adaptación de la ontología debido a limitaciones de tiempo y recursos. Sin embargo, esta idea estricta de ontología no es usada ni por los más puristas en IA.

Así, en (Guarino, 1998), el autor establece que:

*"Una ontología puede especificar una conceptualización en una forma muy indirecta, puesto que i) solo puede aproximar un conjunto de modelos pretendidos; y ii) tal conjunto de modelos pretendidos sólo es una caracterización débil de una conceptualización."*

Otra definición de ontología es la presentada por Borst (Borst, 1997), que refine la definición de Gruber:

*"Una ontología es una especificación formal de una conceptualización compartida."*

En este contexto, "formal" se refiere a la necesidad de disponer de ontologías comprensibles por las máquinas. Esta definición enfatiza la necesidad de consenso en la conceptualización. Finalmente, "compartida" se refiere al tipo de conocimiento contenido en las ontologías, esto es, conocimiento consensuado y no privado.

Posteriormente, las definiciones de Gruber y Borst fueron explicadas en (Studer et al, 1998) de la siguiente forma:

> *"Conceptualización se refiere a un modelo abstracto de algún fenómeno en el mundo a través de la identificación de los conceptos relevantes de dicho fenómeno. Explícita significa que el tipo de conceptos y restricciones usados se definen explícitamente. Formal representa el hecho de que la ontología debería ser entendible por las máquinas. Compartida refleja la noción de que una ontología captura conocimiento consensual, esto es, que no es de un individuo, sino que es aceptado por un grupo"*

## 1.1.2 Historia de la ontología

La palabra Ontología proviene del griego, idioma en el que significa "estudio del ser". Constituye la interpretación fundamental de los constituyentes del mundo de la experiencia. Todos estos constituyentes- individuos, conceptos y objetos con sus atributos- tienen factores o aspectos en común que requieren ser definidos y explicados para ser comprendidos. Ontología es por tanto considerada como la ciencia fundamental que estudia los constituyentes basales y los principios de las ciencias especiales. Sin embargo, el concepto de ontología precede a la palabra "ontología". Se puede decir que los filósofos comenzaron a comprender las implicaciones de este concepto cuando comenzaron a utilizar esta palabra. A continuación, se presenta una evolución histórica de la noción de ontología, basada en varias historias de la ontología existentes (Ferrater, 1963; Smith, 1978, http://www.formalontology.it).

Tales de Mileto fue el primer filósofo. Tales buscaba la respuesta a la pregunta ¿de qué está compuesto todo? Tales concluyó que un elemento debe ser más básico que los otros, y se preguntó cuáles de los seis elementos eran capaces de tomar el mayor número posible de formas. Finalmente concluyó que el agua era el elemento más básico. Anaxímenes de Mileto criticó a Tales, y dijo que ninguno de los seis elementos era fundamental. Para él existía un elemento sin forma ni nombre detrás de todas las cosas, al que llamó "indeterminante" o "ilimitado". Creía que la realidad era como una moneda. Por un lado tenemos el mundo conocido y por el otro tenemos al "ilimitado".

Pitágoras fue el siguiente paso en lo que hoy conocemos como Ontología. Pitágoras no buscaba el componente último de los elementos materiales, sino que para él todas las cosas eran números. Si tomamos esta teoría de forma literal podríamos verla como absurda, pero lo que quería decir, entre otras cosas, es que una descripción correcta de la realidad debe ser expresada en términos de fórmulas matemáticas. Pitágoras creía que Anaxímenes tenía cierta razón. La gente recurría este "indeterminado", pero no a partir de otra realidad. Le cambió el nombre de "indeterminado" por "vis", que reside en el alma de los individuos, quienes tenían una cantidad limitada de "vis" a la que recurrir, puesto que son recipientes limitados. Este "vis" no puede ser percibido por los cinco sentidos, sin embargo, su contrario, llamado "sanguis", sí puede ser percibido por los cinco sentidos. Sanguis es la esencia que reside en el cuerpo y lo que constituye la realidad física. Pitágoras propuso que la formulación podía ser usada para manipular al "vis" y obtener los efectos deseados. Sin embargo, Pitágoras se dio cuenta de que para probarlo necesitaría la ayuda de filósofos de otras "razas", para ver cómo ejercían su magia. Durante diez años se dedicó a investigar este tema hasta caer en la más absoluta de las locuras. Al undécimo año escribió un libro con sus hallazgos. Lo que descubrió y escribió ha creado una nueva ciencia llamada Ontología. Descubrió cómo manipular al "vis" mediante el uso de fórmulas para crear efectos. Con esas fórmulas, lo que antes era considerado como magia ahora era considerado como ciencia. La esposa de Pitágoras, Myia, cuidó de él y nunca se conoció su paradero. Se rumoreaba que se trasladaba continuamente. Sus dos hijas, Damo y Arignote, se dedicaron a enseñar Ontología.

Posterioremente, Aristóteles y sus analizadores de la Edad Media y principios de la Edad Moderna (como Santo Tomás de Aquino) reconocieron que el "ens" se puede conocer de varias formas: (1) **ens como tal**; y (2) **ens en formas específicas**. De hecho, desde los inicios de la Metafísica había formas para distinguir entre ella y lo que después se llamaría "Ontología". Aristóteles asignó un nombre al estudio de la naturaleza y a los principios fundamentales del ens qua ens: "filosofía fundamental". Sin embargo existía bastante incertidumbre con respecto a los conceptos "metafísica" y "filosofía fundamental". En la mayor parte de las veces se suponía que eran lo mismo o eran declaradas de forma muy similar. Únicamente cuando los filósofos escolásticos comenzaron a sustituir exposiciones sistemáticas sobre Metafísica por comentarios sobre la metafísica aristotélica, y cuando se promovieron especulaciones más filosóficas que teológicas se comenzó a distinguir entre una ciencia del ente y una ciencia

sobre tipos de entes específicos. El término "ontología" (o similar) no lo propone ningún autor, y se seguía considerando que Metafísica era suficiente.

En el siglo XVII surgió un nuevo nombre para una nueva disciplina del carácter anterior, que es al mismo tiempo "un nuevo nombre para algunas formas antiguas de pensar". Fue propuesto por filósofos influidos, directa o indirectamente, por la tradición escolástica, pero complementados por la moderna tradición racional. Diversos historiadores mencionan a Johann Clauberg como el primer filósofo que acuñó el término "ontología", aunque esto es discutible, puesto que la primera ocurrencia fue en Rudolf Goclenius (Francoforti, 1613). Desde 1635 creció el interés por clasificar la filosofía en diferentes ramas. El concepto de ontología apareció para designar a la ciencia teórica cuando ésta no había sido dividida en ramas. Lamentablemente, esto no clarifica suficientemente el significado de "Ontología" tal y como fue introducido, a pesar de las consideraciones de que la ciencia teórica, como defendía Clauber, estaba a distancia infinita del conocimiento racional. Esto fue probablemente debido al hecho de que se consideraba que Ontología era puramente racional, pero puesto que Ontología trata con la realidad (aunque con la realidad como tal y no con tipos o partes específicas), la distinción entre Lógica y Ontología era menos clara que entre Ontología y Metafísica. Leibniz usó "Ontología" en su "Introductio ad Encyclopaediam arcanam" (Couturat, 1913) de forma similar a Clauberg.

PHILOSOPHIA
PRIMA,
SIVE
ONTOLOGIA,
METHODO SCIENTIFICA
PERTRACTATA,
QUA
OMNIS COGNITIONIS
HUMANÆ PRINCIPIA
CONTINENTUR.
AUTORE
CHRISTIANO WOLFIO,
EDITIO NOVA
PRIORI EMENDATIOR.
FRANCOFURTI ET LIPSIÆ, MDCCXXXI.
PROSTAT IN OFFICINA LIBRARIA RENGERIANA.

**Figura 1.1 Primer libro titulado Ontología**

Christian Wolff popularizó en círculos filosóficos la palabra 'ontología' (ontologia). Esta palabra aparece en el título de su obra "Philosophia prima sive ontologia methodo scientifica pertractata, qua omnes cognitionis humanae principia continentur", que fue publicada por primera vez en 1730 (ver Figura 1.1). Ontología usa un método demostrativo racional y deductivo, y pretende investigar los predicados más generales tales como entes como tal. Siguiendo a Wolff, Alexander Baumgarten (Baumgarten, 1740) definió ontología (también conocida como ontosofía, metafísica, metafísica universal, o filosofía fundamental) como la ciencia de los predicados más generales y abstractos de todo", de forma que a ella pertenecen los principios cognitivos fundamentales del pensamiento humano. Kant lanzó un ataque de los que hacen época contra la ontología racional de Wolff y Baumgarten; para él ontología era tanto una pseudociencia como una tentación. Estaba convencido de haber logrado eliminarla mediante su analítica trascendental. Toda su "Crítica a la Razón Pura" es, de alguna forma, el trabajo de un hombre obsesionado y profundamente angustiado por la ontología. Por otro lado, la expresión "prueba ontológica" (ontologischer Beweis) usada por Kant no es una forma alternativa de denominar a la prueba anselmiana, sino que se pretende enfatizar la naturaleza de dicha prueba. Sería interesante conocer qué tenía Kant en mente cuando atacó los ambiciosos proyectos de los ontologistas racionales. Un examen de los orígenes del concepto de ontología es indispensable para aclarar el pensamiento de Kant.

La tradición filosófico-lógica a la que pertenecen Brentano, Husserl, Meinong, Stumpf, Reinach e Ingarden fue superada por otra tradición inaugurada por Frege, Russell y Wittgenstein. Las dos tradiciones comparten un enfoque común y realista a la ontología, permaneciendo en oposición conjunta al idealismo de Hegel y Bradley, que aparecieron antes que ellos y a los excesos de idealismo subjetivista y lingüístico florecido desde entonces, especialmente, en escritos fenomenológicos franceses y en los trabajos de algunos filósofos del lenguaje corriente. Sin embargo difieren en la interpretación de las demandas de un realismo adecuado. Los partidarios de la ontología filosófica de Frege-Russell-Wittgenstein suponen que su realismo es compatible con reduccionismo, que se debe medir el progreso filosófico mediante el grado que se pueden explicar las teorías filosóficas; los partidarios de la ontología filosófica de Brentano-Husserl-Ingarden se caracterizan por la fe más absoluta en todo lo dado por la experiencia a cualquier nivel ontológico; sólo aprueban la reducción cuando ésta sea motivada por interconexiones reductivas que sean dadas por la experiencia. Pero ontologías es

más que un análisis superficial; algunas veces hay conocimiento difícil de ver desde el exterior. Esto es lo que los científicos en Inteligencia Artificial se dieron cuenta en la última década. Se encontraron frente a la inmensa tarea de imitar el comportamiento y razonamiento humano en sistemas, y se dio cuenta de que la formalización de conocimiento humano oculto y contextual era una precondición para cualquier sistema inteligente. Tomemos la analogía de una pareja que se da cuenta de la necesidad de estar de acuerdo en el significado de confianza antes de poder comenzar una discusión sobre los celos. Los científicos en IA se centraron en formalizar este conocimiento implícito para llegar a un consenso sobre lo que queremos decir cuando hablamos, de forma similar a como los ontólogos filósofos intentaron revelar los aspectos comunes de las cosas para entenderlas. De esta forma, se recobró la noción de ontología por la IA, que ahora se utiliza para permitir y facilitar la compartición y reutilización de conocimiento.

Desde comienzos de los noventa las ontologías se han convertido en un tema de investigación importante en diferentes comunidades IA, donde podemos incluir Ingeniería del Conocimiento, Procesamiento del Lenguaje Natural, o Representación del Conocimiento. Más recientemente, la noción de ontología se ha popularizado en campos como integración inteligente de información, sistemas cooperativos de información, recuperación de información, comercio electrónico, y gestión de conocimiento. La razón por la cual las ontologías se han convertido en tan populares es, en gran medida, debido a lo que prometen: una comprensión compartida y común de algún dominio que puede ser comunicado entre individuos y aplicaciones. Debido a que las ontologías pretender ser conocimiento consensuado del dominio, su desarrollo es frecuentemente un proceso cooperativo que implica a diferentes individuos, posiblemente en diferentes situaciones geográficas. Los individuos que aceptan una ontología en particular se dice que están comprometidos con dicha ontología.

### 1.1.3 Tipos de ontologías

En la literatura podemos encontrar diferentes tipos de ontologías. Se usan dos criterios principalmente para tales clasificaciones: (a) el tipo de conocimiento contenido; y (b) la motivación de la ontología.

### *1.1.3.1 Clasificación por el conocimiento contenido*

Este es el criterio donde existe mayor diversidad, la cual puede ser ilustrada por las dos siguientes clasificaciones de ontologías. La primera de ellas fue propuesta en (Van Heijst et al, 1997), donde se distinguen tres tipos de ontologías:

- **Ontologías terminológicas, lingüísticas**: Especifican los términos usados para representar conocimiento en el dominio. Un ejemplo de este tipo de ontologías es la red semántica UMLS (Unified Medical Language System) (Lindberg et al, 1993).

- **Ontologías de información**: Especifican la estructura de los registros de la base de datos. Los esquemas de bases de datos serían un ejemplo.

- **Ontologías para modelar conocimiento**: Especifican conceptualizaciones de conocimiento. Estas ontologías tienen una estructura interna mucho más rica que los anteriores tipos de ontologías, y éstas son las ontologías que interesan a los desarrolladores de sistemas basados en conocimiento.

Una clasificación alternativa fue propuesta en (Mizoguchi et al, 1995), donde también se proponen tres categorías:

- **Ontologías del dominio**: Contienen todos los conceptos asociados a un dominio particular.

- **Ontologías de tarea**: Establecen la forma en la cual se puede usar el conocimiento del dominio para realizar tareas específicas. De esta forma, una aplicación podría realizar búsquedas de información mientras otra podría gestionar la asignación de bloques libre de memoria.

- **Ontologías generales**: Contienen descripciones generales sobre objetos, eventos, relaciones temporales, relaciones causales, modelos de comportamiento y funcionalidades.

### *1.1.3.2 Clasificaciones por motivación*

Se van a presentar dos clasificaciones diferentes atendiendo a este criterio. Según la primera de ellas, se distinguen cuatro tipos de ontologías:

- **Ontologías para la representación de conocimiento**: Permiten explicar las conceptualizaciones que subyacen de los formalismos de representación de conocimiento (Davis et al, 1993).

- **Ontologías genéricas**: Definen conceptos considerados genéricos en diferentes áreas. Ejemplos de tales conceptos serían componente, subclase, proceso, estado, etc. Estas ontologías son reutilizables en diferentes dominios. Se llaman también ontologías abstractas o superteorías porque permiten definir conceptos abstractos, y dichas ontologías pueden ser usadas para definir conceptos de forma más específica en diferentes dominios. Como ejemplos podemos ver la taxonomía, la mereología, la topología y la teoría general de sistemas.

- **Ontologías del dominio**: Definen conceptualizaciones específicas del dominio. Las metodologías actuales de adquisición de conocimiento distinguen entre ontologías y conocimiento del dominio, porque el último describe situaciones factuales del dominio, mientras que las ontologías imponen descripciones sobre la estructura y contenido del conocimiento del dominio.

- **Ontologías de aplicación**: Están ligadas al desarrollo de una aplicación concreta. Tales ontologías cubren los aspectos relacionados con aplicaciones particulares. Típicamente, estas ontologías toman conceptos de ontologías del dominio y genéricas, así como

métodos específicos para realizar la tarea, por lo que no son muy adecuadas para ser reutilizadas.

Una clasificación alternativa fue propuesta por Poli (Poli, 2000). En dicha clasificación se identifican los siguientes tipos de ontologías:

- **Ontologías generales:** Tienen que ver con las categorías fundamentales y sus conexiones de dependencia. Con respecto a las categorías fundamentales, los investigadores se dan cada vez más cuenta de la dificultad de manejar este nivel supremo. Por ello, es de máxima importancia emplear una organización de categorías principales que sea lo más transparente posible. Existen categorías fundamentales que se aplican a todos los niveles ontológicos. Sin embargo, muchas de las categorías top-level pueden tener diferentes valores en niveles diferentes de la ontología, aunque deben tener algo en común.

- **Ontologías categóricas:** Estudian las diversas formas en las que una categoría se da cuenta de los diversos niveles ontológicos, determinando la posible presencia de una teoría general que subsume sus concretizaciones. Mientras que la ontología general está más relacionada con la arquitectura de la teoría, la ontología categórica es más sensible a los detalles de las categorías individuales. Sin embargo, es obvio que ambas son necesarias.

- **Ontologías del dominio:** Se refieren a la estructuración detallada de un contexto de análisis con respecto a los subdominios que lo componen.

- **Ontologías genéricas:** Parecen ligadas a corpus lingüísticos y léxicos conceptuales. De hecho, se pueden clasificar los términos en varios niveles. Esto significa que cada término debería ser accesible por defecto únicamente en su sentido genérico, mientas que sus significados especializados quedan para cuando se active una ontología del dominio específica. Por otro lado, la ontología del dominio contiene términos que no

tienen correspondencias analíticas en ontologías genéricas. El conocimiento del dominio "satura" el conocimiento genérico,

- **Ontología regional:** Analiza las categorías y sus conexiones de interdependencia para cada nivel ontológico (estrato o capa).

- **Ontología aplicada:** Estas ontologías son la aplicación concreta de entorno ontológico a un objeto específico (por ejemplo, un hospital).

## 1.2 ONTOLOGÍA FORMAL Y DESCRIPTIVA

Una tercera clasificación se basa en el grado de formalidad de la ontología. Según este criterio se distinguen tres tipos de ontologías en (Poli, 2002):

**Ontología descriptiva**, relacionada con la recolección de información sobre los ítems del dominio analizado. La unidad y variedad del mundo es la salida de las conexiones de dependencia y formas de independencia entre los ítems. Cosas materiales, plantas y animales, así como los productos de los talentos y actividades de animales y humanos, son ítems del mundo. En otras palabras, el mundo no solo contiene cosas, animadas o no, sino también actividades y procesos, así como los productos derivados de los mismos. Es difícil negar que existen pensamientos, sensaciones y decisiones, así como el completo espectro de actividades mentales, así como uno está obligado a admitir la existencia de reglas, lenguajes, sociedades y costumbres (Poli,2001a).

**Ontología formal**, que destila, filtra, codifica y organiza los resultados de una ontología descriptiva. Según esta interpretación, la ontología formal es formal en el sentido usado por Husserl es sus "*Logical Investigations"*. Ser formal en este sentido implica tratar con categorías como cosa, proceso, materia, forma, todo, parte, etc. Estas categorías caracterizan aspectos y tipos de realidad que todavía no han sido utilizados bajo ningún formalismo.

La codificación formal en sentido estricto se da al nivel de **ontología formalizada**. El objetivo es encontrar la codificación formal apropiada para los constructores adquiridos de forma descriptiva y purificarlos formalmente como se indica. El nivel de construcciones formalizadas también está relacionado con la evaluación de la adecuación (expresiva, computacional, cognitiva) de los distintos formalismos, y con el problema de las traducciones recíprocas. La fuerte similaridad entre los términos "formal" y "formalizada" es un contratiempo. Una forma de evitar la confusión es utilizar "categórica" en vez de formal. La mayor parte de las teorías contemporáneas sólo reconocen dos niveles de análisis y suelen unir las categorías formales con el análisis formalizado. Como consecuencia, se suele negar la relevancia específica de los análisis categóricos.

Los tres niveles ontológicos son diferentes pero no están separados, puesto que están relacionados en muchos aspectos. El conocimiento descriptivo puede referirse a categorías formales, y las salidas formalizadas a los otros dos niveles. Por otro lado, es más delicado establecer las diferencias y conexiones entre varias facetas ontológicas como se muestra en (Poli, 2002a).

La aplicación de métodos lógico-formales a una ontología la transforma en ontología formal. Los primeros ontólogos formales creían que la tarea de construcción podía ser llevada a cabo de forma sistemática y está completamente basada en la resolución de problemas lógicos, esto es, en la gramática lógica de lenguajes particulares. En contraste, la antigua tradición ontológica se ha quedado en un almacén de intuiciones ontológicas, constituyendo argumentos informales e incluso retóricos sobre esas intuiciones como base. Como se establece en (Gangemi et al, 1999), las relaciones formales implican entidades de todas las esferas materiales, de forma que son comprensibles per se como nociones universales. Por el contrario, las relaciones materiales son específicas de una o más esferas materiales. Esto presupone una división a priori del dominio en esferas materiales: primero se debe realizar una distinción entre relaciones formales y materiales en base a su comportamiento con respecto a tales subdominios. De esta forma, las relaciones formales establecen las conexiones y las diferencias entre subdominios primitivos, mientras que las relaciones materiales caracterizan las propiedades de un subdominio específico. Si se asume un dominio plano, sin estructura a priori, entonces no sería válida la distinción entre relaciones formales y materiales.

En particular, Edmund Husserl fue el primer filósofo en distinguir entre lógica formal y ontología formal. Por un lado, la lógica formal trata las interconexiones de verdades con relaciones inferencia, consistencia y validez. Por otro lado, la ontología formal trata las interconexiones de las cosas con objetos y propiedades, partes y todos, relaciones y colectivos. Puesto que la lógica formal trata las relaciones de inferencia formales en el sentido de que se aplican inferencias únicamente en virtud de su forma, la ontología formal trata con estructuras y relaciones formales en el sentido que son ejemplificadas por objetos de todas las esferas materiales o dominios de realidad.

La ontología forma de Husserl se basa en la mereología, en la teoría de dependencias y en la topología. En (Smith and Mulligan, 1983) se define la ontología formal de forma similar. La ontología formal es la investigación de de estructuras o relaciones formales al nivel de cosas, mientras que está indirectamente relacionado con el nivel de las verdades. Según estos autores, la ontología formal busca desarrollar un lenguaje formal cuya sintaxis contendrá directamente todas aquellas principios estructurales formales de construcción que se identifiquen en el mundo. Sin embargo, debemos distinguir entre ontología formal y lógica formal puesto que poseen diferente complejidad. La complejidad de la lógica formal es el resultado del trabajo de los lógicos. Por otro lado, la complejidad de la ontología formal no se comprende tan bien, y este hecho tiene su raíz en la confusión, una característica de la filosofía analítica desde su concepción. En (Cocchiarella, 1996) se presenta como una teoría de formas lógicas y una teoría metafísica sobre la estructura ontológica del mundo. Según este autor, lo que la hace una teoría de formas lógicas es que las diferentes categorías ontológicas o modos de ser se representan por medio de diferentes categorías ontológicas. Sin embargo, está más relacionado con una ontología forma que con una gramática ontológica. En particular, una ontología formal determina cómo se pueden transformar dichas expresiones de forma deductiva. Como teoría de formas lógicas, la ontología formal no sólo implica gramáticas ontológicas sino también leyes ontológicas que determinan la validez de la formulación de dichas gramáticas.

En (Albertrazzi, 1996), el autor afirma que la ontología formal se ha desarrollado de dos maneras principales. El primer enfoque consiste en estudiar la ontología formal como parte de la ontología, y analizarla usando las herramientas y se aproxima a la lógica forma: desde este punto de vista la ontología formal examina las características lógicas de predicación, así como

aquellas de las diferentes teorías de universales. El uso del paradigma específico de la teoría de conjuntos aplicada a predicación condiciona su interpretación. El segundo enfoque vuelve a los orígenes Husserlianos y analiza las categorías fundamentales de objeto, estado, parte, todo, etc, así como las relaciones entre partes y todos y sus leyes de dependencia una vez que los conceptos materiales han sido sustituidos por sus conceptos formales correlativos al "algo" puro. Este tipo de análisis no trata con el problema de relaciones entre ontología formal y ontología material.

Finalmente, podría ser interesante ver las relaciones entre diferentes filósofos ontólogos modernos. Tales relaciones aparecen en la Figura 1.2. En esta figura, las líneas indican dos tipos de dependencia: las líneas gruesas significan dependencias importantes y las finas indican dependencias menos relevantes. También se considera una distinción entre dependencia individual y dependencia general. Los nombres están organizados cronológicamente de arriba abajo. Esta organización fue propuesta por Poli, Cocchiarella, Johansson y Smith, y se puede encontrar en http://www.formalontology.it.

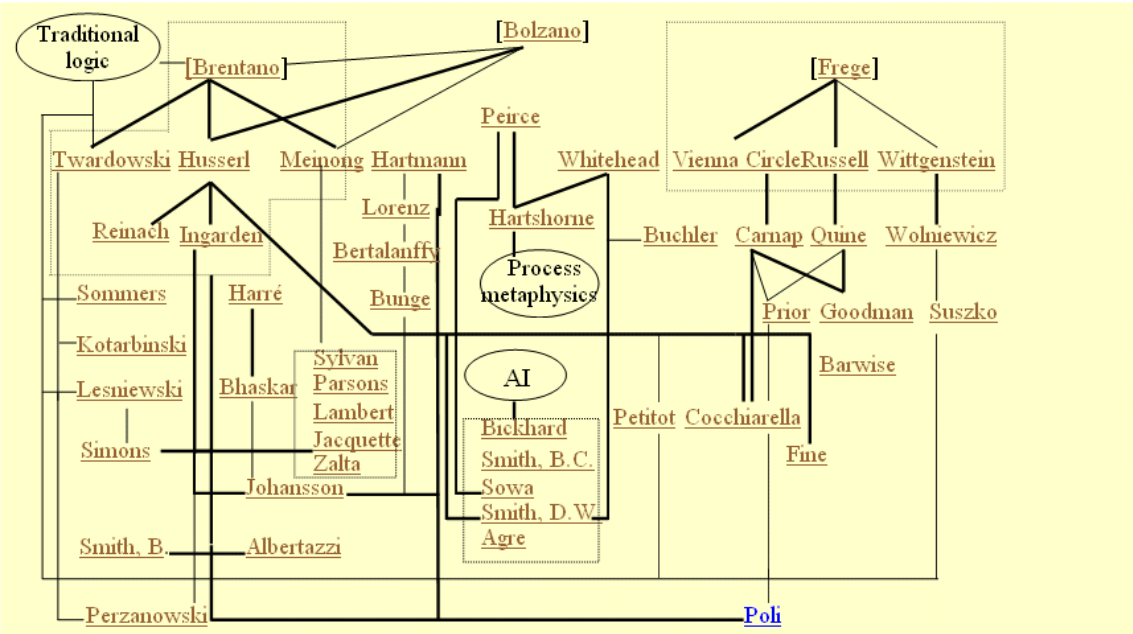


**Figura 1.2 Filósofos Ontólogos Modernos**

# 1.3 INGENIERÍA ONTOLÓGICA

## 1.3.1 Elementos de ontologías

Las ontologías proporcionan un vocabulario común de un área y definen, a diferentes niveles de formalismo, el significado de los términos y relaciones entre ellos. El conocimiento en ontologías se formaliza principalmente usando cinco tipos de componentes: clases, relaciones, funciones, axiomas e instancias (Gruber, 93).

- Las *clases* en la ontología se suelen organizar en taxonomías. Algunas veces, la noción de ontología se diluye en el sentido que las taxonomías se consideran ontologías completas [Studer et al.; 98]. Se suele usar tanto el término clases como conceptos. Un concepto puede ser algo sobre lo que se dice algo y, por lo tanto, también podría ser la descripción de una tarea, función, acción, estrategia, proceso de razonamiento, etc.

- Las *relaciones* representan un tipo de interacción entre los conceptos del dominio. Se definen formalmente como cualquier subconjunto de un producto de n conjuntos, esto es: R: C1 x C2 x ... x Cn. Como ejemplos de relaciones binarios incluimos: "subclase de" y "conectado a".

- Las *funciones* son un tipo especial de relaciones en las que el n-ésimo elemento de la relación es único para los n-1 precedentes. Formalmente, definimos las funciones F como: F: C1 x C2 x ... x Cn-1 → Cn. Como ejemplos podemos mencionar las funciones "madre de" y "precio de un coche usado".

- Los *axiomas* son expresiones que son siempre ciertas. Pueden ser incluidas en una ontología con muchos propósitos, tales como definir el significado de los componentes ontológicos, definir restricciones complejas sobre los valores de los atributos, argumentos de relaciones, etc verificando la corrección de la información especificada

en la ontología o deduciendo nueva información. Tales ontologías son llamadas ontologías pesadas, en contraste con las ontologías ligeras que no incluyen axiomas.

- Las *instancias* se usan para representar elementos específicos.

### *1.3.1.1 Relaciones*

En (Gómez-Pérez et al, 2000) se enumeran las relaciones más comunes en dominios reales, a saber: equivalencia, taxonómica, partonómica, dependencia, topológica, causal, funcional, cronológica, similaridad, condicional y propósito. Sin embargo, no todas las relaciones tienen la misma relevancia ni imponen el mismo tipo de propiedades jerárquicas a la ontología. Entre este conjunto de relaciones podemos subrayar tres de ellas: taxonomía, mereología, y topología.

**Taxonomía**

La palabra taxonomía tiene su origen en dos términos griegos, a saber, taxis (orden) y nomos (tratado) y esta palabra proviene de la Filosofía. Taxonomía es la ciencia que estudia la división en grupos ordenados o categorías. Desde un punto de vista ontológico, una taxonomía es una organización ontológica basada en una relación de orden parcial llamada IS-A, a través de la cual se agrupan las entidades y son subsumidas por clases de más ato nivel. En general, las taxonomías han sido importantes para modelar esquemas de bases de datos, sistemas basadas en conocimiento y vocabularios semánticos (Guarino and Welty, 2001).

A continuación se presentan las propiedades satisfechas por las relaciones taxonómicas. Con este propósito, se usará la notación empleada en (Guarino and Welty, 2001). De esta forma, se dice que un individuo x perteneciente a una clase θ, IS-A individuo de la clase φ si for all x, φ (x) ➔ θ (x). Esto es, existe cierta dependencia entre las clases φ y θ, por lo tanto, cada predicado satisfecho por los individuos de la clase φ también es satisfecho por los individuos de la clase θ. Estas son las propiedades generales taxonómicas:

*Asimetría*: Esta propiedad significa que la inclusión de una clase de individuos, X, en una clase Y implica la no inclusión de Y en X. Formalmente, esta propiedad garantiza que: *(X IS-A Y) if and only if not(Y IS-A X)*.

*Transitividad:* Sea X incluido en una clase Y, que a su vez está incluido en una clase Z, ambas inclusiones a través de relaciones IS-A, entonces, existe una relación IS-A entre las clases X y Z. Formalmente, *(X IS-A Y) and (Y IS-A Z)* ➔ *(X IS-A Z)*.

*Irreflexividad*: Admitir la reflexividad en relaciones taxonómicas solo tendría sentido para modelar tautologías. Una tautología es la expresión de un mismo hecho de distintas maneras. La relación taxonómica se considera no reflexiva. Formalmente, *not(X IS-A X)*.

Existen otras propiedades taxonómicas que están relacionadas con los atributos de los conceptos a través de la taxonomía:

*Redefinición*: Esta propiedad consiste en cambiar el nombre de una propiedad común a dos conceptos, padre e hijo, y se asigna un nombre diferente al atributo en el hijo.

*Herencia múltiple*: Esta propiedad está asociada con atributos conceptuales. Un concepto puede tener diferentes padres taxonómicos, así que este concepto heredaría propiedades de todos sus padres.

Además de las propiedades taxonómicas básicas, existen otras condiciones basadas en cuestiones filosóficas relacionadas con taxonomías. Algunas de estas condiciones se señalan en (Guarino and Welty, 2001):

- Identidad
- Unidad

- Esencia
- Dependencia
- Rigidez

Las dos primeras condiciones se enlazan al concepto filosófico de "ser". Según Guarino, las intuiciones tras ambos conceptos requieren, con la finalidad de comprenderlos, hacer una distinción entre ellos. Así, la condición de identidad se relaciona al problema de distinguir una instancia de su clase específica de instancias de la misma clase, por medio de lo que llamamos "propiedad característica", la cual es única para cada instancia.

Por otro lado, la unidad está relacionada con la distinción de partes en una instancia del resto del mundo. Con este propósito se usa una "relación unificante" que es un nexo entre todas las partes. Sin embargo el límite entre ambas nociones no es evidente, especialmente cuando se invoca una dimensión temporal. De hecho, el principal problema aparece con la noción de "identidad a través de cambios". Eso nos lleva a admitir que los individuos pueden permanecer invariables aunque varíen algunas de sus propiedades. Estas propiedades invariables se llaman "propiedades esenciales", y constituyen la esencia de la clase.

Análogamente, la rigidez se define a través de "propiedades rígidas". Una propiedad rígida es una propiedad esencial para todas las instancias en las que aparece dicha propiedad. Finalmente, la dependencia ontológica se refiere a la existencia de relaciones particulares entre atributos y conceptos. Concretamente, en (Guarino and Welty, 2000a) se distinguen dos categorías, intrínseca y extrínseca, según la dependencia de objetos. En particular, las propiedades extrínsecas no son inherentes a objetos sino que están ligadas a objetos. Finalmente, hay un número de conceptos que están asociados a relaciones IS-A que se han tomado directamente del modelado conceptual de bases de datos o del paradigma orientado a objetos, tales como claves primarias o identificadores únicos. Sería interesante formalizar las condiciones presentadas anteriormente. Para ello se usará la formalización presentada en (Guarino, 2000a).

Identidad

Supongamos dos individuos taxonómicos, x e y, y una relación θ tal que se cumplan θ(x) y θ(y). Se dice que hay una condición de identidad en uno de ellos si hay una relación φ tal que:

1. θ(x) Λ θ(y)

2. φ(x,y)

3. x=y

Rigidez

Según (Guarino, et al, 1994; Guarino and Welty, 2000a), la rigidez de una propiedad se define formalmente con respecto a la cardinalidad del conjunto de instancias para el que tal propiedad es esencial, de acuerdo a las siguientes definiciones:

*Definición 1. Propiedad rígida*: Una propiedad rígida es una propiedad esencial para todas sus instancias. Formalmente, se dice que la propiedad θ es rígida si, para todo x que la tiene definida se cumplen:

1. para todo x, θ(x)

2. θ (x)

*Definición 2. Propiedad no rígida*: Una propiedad es no rígida cuando no es esencial para todas sus instancias. Formalmente, se dice que la propiedad θ es no rígida si, para todo x que la tiene definida se cumplen:

1. existe x, θ(x)

2. No θ (x)

*Definición 3. Propiedad anti-rígida*: Una propiedad es anti-rígida cuando es no esencial para ninguno de sus instancias en todos los posibles mundos. Formalmente, se dice que la propiedad θ es anti-rígida si, para todo x que la tiene definida se cumplen:

1. para todo x, θ(x)

2. No θ (x)

**Diseño y construcción de taxonomías**

De acuerdo con (Kremer, 2001), existen tres criterios a tomar en cuenta cuando construimos taxonomías:

- Adecuación descriptiva. La taxonomía debe clasificar la totalidad de los objetos del dominio a modelar (este criterio asume una metodología de construcción bottom-up en términos de nivel de abstracción)

- Simplicidad. La selección de nombres para cada elemento del dominio, así como su clasificación, debería ser una tarea simple.

- Capacidad predictiva. Esta es la propiedad más deseable para buenas taxonomías. Objetos con clasificaciones similares deberían tener propiedades similares.

Una buena taxonomía debe poseer un umbral de discriminación que debe ser lo suficientemente alto para no agrupar diferentes objetos en la misma categoría, y lo suficientemente bajo para permitir la existencia de un número adecuado de categorías a modelar de forma manejable. Sin embargo, existen diferentes problemas relativos al uso de taxonomías. Los problemas más frecuentes derivados del mal uso de esta relación son discutidos en (Guarino, 1999). En particular se distinguen cinco problemas:

A) Reducción de sentido: Este es el caso de situaciones como "una asociación es un grupo".

B) Sobre-generalización. Este es el caso de situaciones como "una localización es un objeto físico".

C) Confusión de sentido: Este es el caso de situaciones como "una ventana es tanto un artefacto como una localización".

D) Conflicto de sentidos: Este es el caso de "una organización es tanto un ser social como un grupo".

E) Enlace sospechoso tipo-rol: Este es el caso de "una manzana es tanto una fruta como una comida".

Estos problemas han sido extraídos del análisis de taxonomías existentes conocidas. En particular, se han tomado ejemplos de las siguientes ontologías: PANGLOS (Knight and Luk, 1994), WORDNET (Miller, 1995), MICROCOSMOS (Mahesh, 1996), y CYC (Lenat et al, 1990).

## Mereología

El término mereología es un término griego que significa “estudio de las partes”. Este término fue usado por el filósofo polaco Lešniewsky para introducir una teoría formal sobre las partes y sus conceptos asociados. En (Simons, 1987) se presenta la Mereología Clásica Extensional. Allí, se ve la mereología como un modelo de organización ontológico. En la siguiente tabla, se presenta la notación usada por Simons, que será usada posteriormente en esta sección.

| Símbolo | Significado | Símbolo | Significado |
|---|---|---|---|
| $\exists$ | Cuantificador Existencial | $+$ | Suma binaria |
| $\forall$ | Cuantificador Universal | $-$ | Diferencia mereológicas |
| $\langle ... \rangle$ | Ámbito de cuantificación | $\cdot$ | Producto binario |
| $=$ | Igualdad | $\sigma$ | Suma general |
| $\approx$ | Identidad | $\pi$ | Producto general |
| $<$ | Parte de | $\perp$ | Conexión |
| $<<$ | Parte propia de | $\rceil\lceil$ | Desconexión |
| $\oplus$ | Superposición | $\times$ | Conexión externa |
| $\mid$ | Disyunción | | |

La mereología va más allá del estudio de relaciones parte-de entre elementos de sistemas comunes. La mereología también se ocupa de aquellos elementos cuyas partes y todos son relevantes. Aquellos elementos se identifican como individuos. Con el objeto de realizar un razonamiento mereológico basado en individuos, se deben enunciar algunos principios y propiedades mereológicas.

Parte-de, parte propia e igualdad

En mereología, la relación parte-de se expresa mediante el operador "<". Esto permite establecer dos significados para cada relación: (1) uno no incluye el concepto de igualdad entre individuos, al que nos referimos como parte propia da ("<<"); (2) uno incluyendo la igualdad, referido como parte de ("<"). Ambos aspectos se pueden formalizar como sigue:

Dados dos individuos, x e y, x es una parte de y, denotado por x<y, sí y sólo si:

i) x << y, ó

ii) x = y

donde x<<y significa que "x es una parte propia de y"

Hay algunas propiedades que definen el significado del concepto universal de parte. Tales propiedades son asimetría y transitividad, de las que surge la no reflexividad. Formalicemos dichas propiedades.

Asimetría: $x << y \rightarrow \text{not } (y << x)$
Transitividad: $(x << y) \text{ and } (y << z) \rightarrow x << z$

Definamos ahora otros conceptos relacionados con relaciones parte-de. La superposición es la afirmación de que dos individuos tienen partes comunes. Formalmente, se puede definir como sigue: Dados dos individuos, x e y, $\exists z$ s.t. $\langle z < x \text{ and } z < y \rangle$. Por otro lado, la disjunción es la negación lógica de la superposición, y se define como not $\langle x \oplus y \rangle$. Se puede definir fácilmente que la superposición es reflexiva y simétrica pero no transitiva. Por otro lado, la superposición propia es la superposición que existe entre dos individuos x e y tales que x es parte de y, o ambos tienen una parte propia común.

Principio de suplementación débil: Si un individuo tiene una parte propia, entonces debe tener una parte que sea distinta de la anterior. Formalmente, $x << y \rightarrow \exists z$ t.q. $\langle (z << y \rangle$ and $(z \mid x)\rangle$.

La parte común de los individuos superpuestos se llama límite inferior. La mereología establece que, dados dos individuos superpuestos x e y, existe un individuo que posee como partes todos los límites inferiores entre dichos individuos. Tal individuo es el producto binario, que se basa en la necesidad de diferenciar los conjuntos de partes solapadas entre x e y de los propios individuos. La suma binaria de dos individuos x e y, denotada por x+y, se define como el individuo que está superpuesto a, al menos, uno de ellos. El supuesto de existencia de la suma binaria para cada par de individuos mereológicos es la propiedad más controvertida en mereología clásica extensional.

Dados dos individuos superpuestos, sus partes compartidas las enlazan para formar un individuo simple. La suma es plausible para individuos de tipos similares pero no para aquellos individuos que sean muy diferentes en términos espacio-temporales o que tienen tipos diferentes. Por otro lado, dados dos individuos x e y que cumplen la superposición propia, su diferencia mereológica, denotada por x-y, puede ser calculada como la parte propia de x más grande que no tiene partes comunes con y. Esto sólo existirá si x no es parte de y. La existencia de la suma o producto para cada par de individuos no garantiza la existencia de la suma para cada clase de individuos, ni la existencia del producto para cada clase de individuos con una parte común, porque tales clases pueden ser infinitas (Simons, 1987). Sin embargo, se puede estudiar la suma o producto de individuos que pertenecen a cierta clase y que satisface cierto predicado. Formalmente, la suma general de cada individuo x perteneciente a la clase F, denotado por $\sigma x\langle Fx\rangle$, se define como sigue:

i) Fx, y

ii) $\forall y$ t.q. $\langle x \oplus y\rangle$, y

iii) $\exists z$ t.q. $\langle Fz \rangle$ and $\langle z \oplus y \rangle$

Dado un individuo que satisface un predicado en particular, se puede decir que, para cada individuo que se solapa con aquél, existe un tercer individuo que satisface los mismos predicados que se solapan con el segundo. Esto permite extender el concepto de suma a suma general. De forma similar, la noción de producto binario puede ser extendida a producto general, de la forma siguiente. El producto general del individuo x de la clase F, denotado por $\pi x \langle Fx \rangle$, cumple las siguientes condiciones:

i) $\forall y$ t.q. $\langle \langle y < x \rangle$ ; y

ii) $\langle \forall z \langle Fz \rangle$ and $\langle y < z \rangle \rangle$

Principio de la Suma General

Este principio permite la completa axiomatización de la mereología clásica extensional. Esto garantiza la existencia de sumas generales. Formalmente:

i) $\exists x \in Fx$, y

ii) $\exists x$ t.q. $\forall y \langle \langle y \oplus x \rangle \rangle$, y

iii) $\exists z$ t.q. $\langle \langle Fz \rangle$ and $\langle y \oplus z \rangle \rangle$

Principio de Extensionalidad

Este axioma se puede derivar de los anteriores, y establece que un individuo sólo puede ser distinguido por sus partes. Ahora entraríamos en la discusión sobre la posibilidad de que dos individuos tengan las mismas partes y que pudieran ser diferentes. Esto no es posible en mereología extensional. Sin embargo, existen distintos tipos de individuos, tales como eventos o clases para los cuales se debería permitir tal supuesto.

Atomismo

En mereología atomista, cada individuo puede ser considerado como construido a partir de individuos menores llamados átomos. Un átomo, denotado por At, es un individuo sin partes propias. Formalmente, las condiciones satisfechas por individuos atómicos son:

i) Atx, y

ii) not $\langle \exists z \text{ t.q. } \langle z << x \rangle \rangle$

Por otro lado, ciertos tipos de individuos son atómicos y otros no. Los intervalos temporales son ejemplos de esta última situación, porque pueden dividirse en intervalos menores. Sin embargo, los componentes de sistemas electrónicos son átomos del sistema. Esta suposición de la atomicidad del individuo simplifica la teoría, porque un individuo sería parte de otro individuo sí y sólo si todos sus átomos son también átomos del último. Esta propiedad también simplifica los axiomas mereológicos. Por ejemplo, la relación de identidad puede ser redefinida como la posesión de los mismos átomos. En resumen, no hay una teoría mereológica única, pero esto ofrece algunos ingredientes que se pueden utilizar para construir nuestras propias propiedades ontológicas. Cada teoría es adecuada para un tipo particular de individuo.

La transitividad de las relaciones parte/todo ha sido discutida por varios autores (Cruse,1979; Winston et al, 1987;Gerstl and Pribbenow,1993). Según (Winston et al,1987), las

relaciones mereológicas pueden ser clasificadas en siete clases diferentes y, en general, no se asume la transitividad entre instancias de diferentes clases. Estas son las siete clases:

- Componente/Objeto: rama/árbol
- Miembro/Colección: árbol/bosque
- Porción/Masa: porción/tarta
- Constituyente/Objeto: aluminio/avión
- Característica/Actividad: pago/compra
- Fase/Proceso: adolescencia/desarrollo
- Lugar/Área: Murcia/España

La mayor parte de la investigación en las relaciones parte-de se ha centrado en el estudio de las partes. Sin embargo, en (Poli, 2001) se presenta una teoría de los todos. En ella se identifican diferentes tipos de todos de acuerdo a las siguientes propiedades:

- Separabilidad de partes y todos.
- Propiedad espacio-temporal de las partes.
- Papel desempeñado por la parte con respecto al todo.
- Homogeneidad de las partes.

Este autor presenta tres tipos de todos: agregados, todos y sistemas. En los agregados existe unidad por proximidad, pero no solidaridad ni unidad dinámica. En los todos podemos encontrar unidad por proximidad y solidaridad, mientras que no existe unidad dinámica. Finalmente, estas tres propiedades se pueden identificar en los sistemas.

#### *1.3.1.2 Otra clasificación alternativa*

Una vez que se han presentados los componentes de las ontologías, la siguiente cuestión es ¿cómo se explicita una ontología? Uschold y Grüninger (Uschold et al., 96) distinguieron cuatro tipos de ontologías según el tipo de lenguaje usado para su implementación:

- Ontologías altamente informales si se escriben en lenguaje natural.
- Ontologías semi-informales si son expresadas en una forma restringida y estructurada de lenguaje natural (p.ej., usando patrones).
- Ontologías semi-formales, que se definen en un lenguaje formal artificial.
- Ontologías rigurosamente formales, que se definen en un lenguaje con semántica formal, teoremas y pruebas de tales propiedades como completitud.

## 1.3.2 Ejemplos de ontologías

En las secciones anteriores se presentan diferentes clasificaciones de las ontologías. A continuación describiremos varios ejemplos de ontologías pertenecientes a dichos tipos. En particular, los ejemplos pertenecen a estas categorías:

- **Ontologías para representar conocimiento** (van Heijst et al., 97): Capturan las primitivas de representación usadas para formalizar conocimiento en los paradigmas de representación de conocimiento. El ejemplo más representativo de este tipo de la Frame-Ontology (Gruber, 93), que captura las primitivas de representación (clases, instancias, ranuras, facetas, etc) usadas en lenguajes basados en frames. Esta ontología define los términos que capturan convenciones usadas en sistemas de representación de conocimiento centrados en objetos. Puesto que estos términos se construyen sobre la semántica de KIF, se puede pensar que KIF y la frame-ontology constituyen un lenguaje especializado de representación. Dicha ontología es la base conceptual para los traductores de Ontolingua. Un propósito de esta ontología es permitir que diferentes sistemas de representación puedan compartir ontologías centradas en objetos. Los traductores de ontologías escritos en KIF usando la frame ontology, tales como los proporcionados por Ontolingua permiten trabajar desde un formato fuente común, y todavía lo siguen usando sistemas actuales de representación.

- **Ontologías generales/comunes** (Mizoguchi et al., 95): Incluyen vocabularios relacionados con cosas, eventos, tiempo, espacio, causalidad, comportamiento, función, etc. A continuación, describiremos las características deseables para este tipo de ontologías (http://www.cyc.com/cyc-2-1/cover.html):

    - Es "universal", esto es, cada concepto imaginable puede ser enlazado correctamente en la ontología top-level en lugares adecuados, sin importar lo general o específico, lo arcano o prosaico, o el contexto en el que esté.

- Es "articulada". Las distinciones hechas en la ontología son necesarias y suficientes para la mayoría de propósitos. Existe una justificación teórica y pragmática para cada clase, para cada predicado y función, para cada individuo. Por suficiente entendemos que se hacen bastantes distinciones para permitir y facilitar la compartición del conocimiento, la integración y limpieza de las bases de datos, etc.

Un ejemplo sería la ontología top-level de Sowa (Sowa, 2000), que incluye las categorías y distinciones básicas derivadas de una variedad de fuentes de la Lógica, Lingüística, Filosofía e Inteligencia Artificial. Esta ontología tiene una estructura de enrejado donde el concepto raíz es el "*universal type"*, y el concepto ínfimo es el "*absurd type"*. Como subtipos del universal tenemos los conceptos primitivos (*independent*, *relative*, *mediating*, *continuant*, *occurrent*). Combinando estos conceptos primitivos obtenemos más conceptos de la ontología. Por ejemplo, *history = proposition* ∩ *occurrent*. Siguiendo con este proceso de combinación de conceptos se obtendría el resto de conceptos. Este enrejado quedaría cerrado por el tipo absurdo.

- **Meta-ontologías**, también llamadas ontologías genéricas o centrales (van Heijst et al., 97), que son reusables en varios dominios. El ejemplo más representativo sería una ontología mereológica (Borst, 97) que incluiría el término parte-de. Otro ejemplo sería la ontología Cyc. La metodología Cyc surge de la experiencia del desarrollo de la base de conocimiento Cyc, que contiene una inmensa cantidad de conocimiento común, y que se construye sobre un núcleo de más de 1.000.000 de axiomas introducidos manualmente para capturar una gran parte de lo que la gente considera conocimiento consensuado sobre el mundo. La codificación de esta base de conocimiento fue realizada en CycL, que es un cálculo de predicados de primer orden aumentado con extensiones para manejar igualdad, razonamiento por defecto y algunas características de segundo orden. Esta base de conocimiento puede ser considerada una ontología porque puede usarse como base para construir sistemas inteligentes así como para comunicación e interoperabilidad.

- **Ontologías del dominio** (Mizoguchi et al., 95;van Heijst et al., 97), que son reusables en un dominio determinado. Proporcionan vocabularios sobre los conceptos y relaciones de un dominio, sobre las actividades que se desarrollan en dicho dominio, y sobre las teorías y principios elementales que gobiernan dicho dominio. Por ejemplo, las ontologías médicas se introducen para resolver problemas tales como la petición de reutilización y compartición de datos de los pacientes, su transmisión y necesidad de criterios basados en semántica con propósitos estadísticos. En este sentido, la comunicación no ambigua de conceptos médicos detallados y complejos es crucial hoy en día para sistemas de información médica. A continuación presentamos dos ejemplos, GALEN y UMLS.

    - GALEN (Rector et al, 1995) incluye un modelo semánticamente válido de terminología clínica representado en un lenguaje formal, y asociado con soporte sofisticado para diferentes lenguajes naturales y conversión entre diferentes esquemas de codificación. GALEN se basa en un modelo semántico sólido de terminología clínica llamado GALEN Coding reference (CORE). Este modelo contiene conceptos clínicos elementales (p.ej., fractura, hueso, etc), relaciones que controlan la combinación de conceptos (p.ej., los huesos pueden tener fracturas), y conceptos complejos (p.ej., fractura de la clavícula).

    - El Unified Medical Language System (UMLS) (Pisanelli et al, 1998; NLM, 1999) es una base de datos diseñada para unificar terminologías biomédicas de fuentes dispares tales como terminologías clínicas, fuentes de drogas, o vocabularios en diversos idiomas. Existen tres fuentes de conocimiento UMLS, a saber: (1) metatesauro, que contiene información semántica sobre conceptos biomédicos, sus nombres y relaciones entre ellos;(2) red semántica, que es una red de categorías generales o tipos semánticos a lo que se asignan todos los conceptos del metatesauro; (3) el lexicón especialista, que contiene información sintáctica sobre términos biomédicos y eventualmente cubrirá la mayoría de términos correspondientes a los nombres de conceptos que aparecen en el metatesauro.

- **Ontologías lingüísticas**. La característica principal de este tipo de ontología es que están limitadas a la semántica de las unidades gramaticales. Forman un grupo heterogéneo de recursos usados principalmente en procesamiento del lenguaje natural. La mayoría de ontologías lingüísticas usan palabras como unidades gramaticales, de hecho, entre las ontologías que hemos analizado, sólo la Generalized Upper Model captura información sobre unidades gramaticales mayores que palabras. Asimismo, algunas de ellas presentan un mapeo uno a uno entre conceptos y palabras en lenguaje natural, mientras que en otras ontologías se facilita un mapeo múltiple. Existen también diferencias con respecto al grado de dependencia del lenguaje. Algunas ontologías lingüísticas dependen totalmente de un idioma en particular; algunas abarcan múltiples lenguajes. Estas ontologías tienen diversos orígenes y motivaciones: bases de datos léxicas on-line; traducción, generación de lenguaje natural, etc. Wordnet (Miller, 1995) es probablemente la ontología lingüística más importante. Es una base de datos léxica para inglés basada en teorías psicolingüísticas. Está organizado en 70.000 conjuntos de sinónimos, llamados synsets, cada uno representando un concepto léxico. Los synsets se enlazan entre sí por medio de relaciones semánticas. Wordnet divide el lexicón en cinco categorías: nombres, verbos, adjetivos, adverbios y palabras con función. Los nombres se organizan en jerarquías temáticas, los verbos a través de relaciones de construcción, y los adjetivos y adverbios en hiperespacios N dimensionales.

## 1.3.3 Criterios para diseñar ontologías

En primer lugar, presentaremos un ejemplo de una ontología de sistemas informáticos tomada de (Borst, 1997):

Define-theory systems_nature

Define-class type_of_system(x)

Instance-of (personal_computer, type_of_system)

Instance-of(DB_application, type_of_system)

Instance-of(casino_videogame, type of system)

Define-relation built(s,n)

Built(personal_computer, physical)

Built(DB_application, logical)

Built(casino_videogame, physical)

Define-relation is-a(s,n)

Is-a (personal_computer, hardware)

Is-a (DB_application, software)

Is-a (casino_videogame, hardware)

En la primera línea se define el nombre de la teoría, que indica el dominio al que pertenece la teoría. La segunda línea contiene la definición de un concepto llamado "type_of_system" del dominio en cuestión. Después de esto se definen diferentes axiomas relativos a la naturaleza de los sistemas, así como sus instancias. Se define una relación llamada "built", que asocia instancias de sistemas con el medio con el que fue construido. Finalmente se clasifican los sistemas de acuerdo con su naturaleza hardware o software.

Es importante señalar que las ontologías definen términos y relaciones entre términos de un dominio. Sin embargo, no se especifica el significado de dichos términos, que se supone

conocido por el grupo de individuos que usarán la ontología. Por lo tanto, la selección de términos para los conceptos y la descripción de la ontología en lenguaje natural son importantes para su uso y comprensión. De esta forma, sería deseable disponer de algún criterio para construir ontologías comprensibles y útiles.

A continuación, se enumerará una serie de criterios de diseño y un conjunto de principios de probada utilidad para el desarrollo de ontologías (Gruber, 1995;Bernaras et al, 1996;Borgo et al, 1996; Arpírez et al, 1998;Gómez-Pérez and Benjamins, 1999):

- **Claridad y objetividad**, que significan que la ontología debería proporcionar el significado de los términos definidos al proporcionar definiciones objetivas y también documentación en lenguaje natural.

- **Completitud**, esto es, que se prefiere una definición expresa da en términos de condiciones necesarias y suficientes a una definición parcial (por ejemplo, sólo en base a condiciones necesarias).

- **Coherencia**, para permitir inferencias consistentes con las definiciones.

- **Extensibilidad monótona máxima**. Significa que los términos nuevos o especializados deben ser incluidos en la ontología de forma que no requiera la revisión de las definiciones existentes.

- **Compromiso ontológico mínimo.** Se refiere a acordar el uso de una terminología compartida de forma coherente y consistente. Garantiza la consistencia, que no la completitud, de la ontología.

- **Principio de Distinción Ontológica** que significa que las clases en una ontología deberían ser disjuntas.

- **Diversificación de jerarquías** para aumentar la potencia proporcionada por los mecanismos de herencia múltiple.

- **Modularidad** para minimizar el acoplamiento entre módulos.

- **Minimización de la distancia semántica entre conceptos hermanos**, que significa que se agrupan los conceptos similares y se representan usando las mismas primitivas.

- **Estandarización de nombres** cuando sea posible [Arpírez et al.; 98].

## 1.3.4 Metodologías para construir ontologías

En la literatura podemos encontrar diferentes metodologías para construir ontologías. Estas metodologías las podemos clasificar de acuerdo a diferentes parámetros. Por un lado, existen metodologías para construir ontologías desde cero. Por otro lado, también existen metodologías para construir ontologías a través de procesos de reingeniería. Finalmente, podemos encontrar metodologías para la construcción cooperativa de ontologías.

### *1.3.4.1 Metodologías para Construir Ontologías a partir de cero*

Sobre 1990, en (Lenat el at.; 1990) se publicaron los pasos generales y algunos puntos interesantes relacionados con el proceso de desarrollo de Cyc. Posteriormente, en (Uschold et al.; 1995) se comentó la metodología empleada para la creación de la ontología TOVE (TOronto Virtual Enterprise) para modelar organizaciones. Al año siguiente, estos autores propusieron unas reseñas metodológicas para construir ontologías (Uschold et al, 1996). En (Bernaras et al, 1996) se presentó un método para construir ontologías para redes eléctricas como parte del proyecto KACTUS. Methontology (Fernández et al.; 1997) apareció simultáneamente y fue extendido posteriormente (Fernández-López et al.; 1999), (Fernández-López et al.; 2000). En 1997, se propuso otra metodología basada en la ontología SENSUS (Swartout et al.; 1997).

Procedamos a continuación a describir un conjunto de diferentes metodologías para la construcción de ontologías a partir de cero.

**Metodología Cyc**

La metodología Cyc (Lenat et al, 1990) consiste en varios pasos. En primer lugar hay que extraer manualmente el conocimiento común que está implícito en diferentes fuentes. A continuación, una vez que tengamos suficiente conocimiento en nuestra ontología, podemos adquirir nuevo conocimiento común usando herramientas de procesamiento de lenguaje natural o aprendizaje computacional. Así se construyó la ontología Cyc. Esta metodología recomienda los siguientes pasos:

- Codificación manual de conocimiento implícito y explícito extraído de diferentes fuentes

- Codificación de conocimiento usando herramientas software

- Delegación de la mayor parte de la codificación en las herramientas

**Metodología de Construcción de Ontologías de Uschold y King**

Esta metodología fue presentada en (Uschold et al.; 95) y propone algunos pasos generales para desarrollar ontologías, a saber: (1) identificar el propósito; (2) capturar los conceptos y relaciones entre estos conceptos y los términos utilizados para referirse a estos conceptos y relaciones; (3) codificar la ontología. La ontología debe ser documentada y evaluada, y se pueden usar otras ontologías para crear la nueva. De esta forma se creó la Enterprise Ontology. Esta metodología recomienda los siguientes pasos:

- Identificar propósito
- Capturar la ontología
- Codificación
- Integrar ontologías existentes
- Evaluación
- Documentación

**Metodología de Construcción de Ontologías de Grüninger y Fox**

En esta metodología, presentada en (Grüninger et al.; 95), el primer paso es identificar intuitivamente las aplicaciones posibles en las que se usará la ontología. Posteriormente, se usa un conjunto de preguntas en lenguaje natural, llamadas cuestiones de competencia, para determinar el ámbito de la ontología. Se usan estas preguntas para extraer los conceptos principales, sus propiedades, relaciones y axiomas, los cuales se definen formalmente en Prolog. Por consiguiente, ésta es una metodología muy formal que se aprovecha de la robustez de la lógica clásica y que puede ser usada como guía para transformar escenarios informales en modelos computables. Esta metodología, que se usó para construir la ontología TOVE, recomienda los siguientes pasos:

- Escenarios motivantes
- Cuestiones informales de competencia
- Terminología formal

- Cuestiones formales de competencia
- Axiomas formales
- Teoremas de completitud

**Metodología KACTUS**

En esta metodología (Bernaras et al, 1996) se construye la ontología sobre una base de conocimiento por medio de un proceso de abstracción. Cuantas más aplicaciones se construyen, las ontologías se convierten en más generales y se alejan más de una base de conocimiento. En otras palabras, se propone comenzar por construir una base de conocimiento para una aplicación específica. A continuación, cuando se necesita una nueva base de conocimiento en un dominio parecido, se generaliza la primera base de conocimiento en una ontología y se adapta para las dos aplicaciones, y así sucesivamente. De esta forma, la ontología representaría el conocimiento consensuado necesario para todas las aplicaciones. Esta metodología ha sido utilizada para construir una ontología para diagnosticar fallos, y recomienda seguir los siguientes pasos:

- Especificación de la aplicación
- Diseño preliminar basado en categorías ontológicas top-level relevantes
- Refinamiento y estructuración de la ontología

**METHONTOLOGY**

Methontology es una metodología para construir ontologías tanto partiendo desde cero como reusando otras ontologías, o a través de un proceso de reingeniería. Este entorno permite la construcción de ontologías a nivel de conocimiento, e incluye: (1) identificación del proceso de desarrollo de la ontología donde se incluyen las principales actividades (evaluación, gestión

de configuración, conceptualización, integración, implementación, etc); (2) un ciclo de vida basado en prototipos evolucionados; y (3) la metodología propiamente dicha, que especifica los pasos a ejecutar en cada actividad, las técnicas usadas, los productos a obtener y cómo deben ser evaluados. Esta metodología está parcialmente soportada por el entorno de desarrollo ontológico WebODE. Esta metodología ha sido usada en la construcción de múltiples ontologías, como una ontología química, ontologías hardware y software, etc. Se proponen los siguientes pasos:

- Especificación
- Conceptualización
- Formalización
- Implementación
- Mantenimiento

**Metodología SENSUS**

La metodología basada en Sensus (Swartout et al, 1997) es un enfoque top-down para derivar ontologías específicas del dominio a partir de grandes ontologías. Los autores proponen identificar un conjunto de términos semilla que son relevantes en un dominio particular. Tales términos se enlazan manualmente a una ontología de amplia cobertura. Los usuarios seleccionan automáticamente los términos relevantes para describir el dominio y acotar la ontología Sensus. Consecuentemente, el algoritmo devuelve el conjunto de términos estructurados jerárquicamente para describir un dominio, que puede ser usado como esqueleto para la base de conocimiento. Esta metodología sirvió para construir la ontología Sensus y recomienda los siguientes pasos:

- Tomar una serie de términos como semillas.
- Enlazarlos manualmente.
- Incluir todos los conceptos en el camino que va de la raíz de Sensus a los conceptos semilla.
- Añadir nuevos términos relevantes del dominio.
- Opcionalmente, añadir para aquellos nodos por los que pasan más caminos su subárbol inferior.

**Metodología On-To-Knowledge**

El proyecto OTK (Staab et al, 2001) aplica ontologías a la información disponible electrónicamente para mejorar la calidad de la gestión de conocimiento en organizaciones grandes y distribuidas. La metodología proporciona guías para introducir conceptos y herramientas de gestión de conocimiento en empresas, ayudando a los proveedores y buscadores de conocimiento a presentar éste de forma eficiente y efectiva. Esta metodología incluye la identificación de metas que deberían ser conseguidas por herramientas de gestión de conocimiento y está basada en el análisis de escenarios de uso y en los diferentes papeles desempeñados por trabajadores de conocimiento y accionistas en las organizaciones. Cada una de las herramientas de la arquitectura de OKT se centra en el desarrollo de aplicaciones dirigidas por ontologías y finalmente describe el uso y la evaluación de la metodología mediante casos de estudio, como por ejemplo, la ontología Proper o AIFB. Los siguientes pasos son recomendados por esta metodología:

- Estudio de viabilidad

- Comienzo
- Refinamiento
- Evaluación

**TERMINAE**

Terminae (Aussenac-Gilles et al, 2002) aporta tanto una metodología como una herramienta para la construcción de ontologías a partir de textos. Se basa en un análisis lingüístico de los textos, el cual se realiza mediante la aplicación de diferentes herramientas para el procesamiento del lenguaje natural. En particular se usan dos herramientas: (1) Syntex para identificar términos y relaciones; y (2) Caméléon para identificar roles o relaciones. Estas herramientas se basan en la misma hipótesis lingüística: el significado de las frases y las palabras es específico para un dominio y puede ser inferido de la observación de regularidades en documentos. La metodología funciona como sigue. Mediante la aplicación de Syntex obtenemos una lista de posibles palabras y frases del texto y algunas dependencias sintácticas y gramaticales entre ellas. Estos datos se usan como entrada para el proceso de modelado junto con el texto original. De esta forma, la identificación de conocimiento se basa en dos tareas que se realizan alternativamente:

- Explorar los resultados Syntex para identificar conocimiento importante o decidir cómo representar alguna información de acuerdo al uso de las palabras en el texto.
- Extraer sistemáticamente del texto tanto conocimiento como sea posible.

Cada pieza de conocimiento puede ser representada en el modelo de conocimiento de Terminae, cuyo lenguaje de representación de conocimiento posee las siguientes primitivas: fichero terminológico (términos), conceptos genéricos (clases), conceptos primitivos (instancias), y roles (relaciones). El siguiente paso es normalizar el conocimiento para obtener

una ontología bien estructurada, donde cada concepto quede justificado por sus relaciones con otros conceptos. Esta metodología sugiere aplicar criterios diferenciadores para hacer explícitas las propiedades comunes y diferentes de un concepto con sus respectivos conceptos padre y hermanos debidas a sus roles. La última etapa es la formalización de la ontología en el lenguaje formal Terminae, que es un tipo de lógica descriptiva. Una función de clasificación sirve para comprobar la corrección de las definiciones de conceptos genéricos, ya que sólo pueden ser definidos si tienen roles diferenciados.

#### *1.3.4.2 Metodologías para Reingeniería Ontológica*

La reingeniería ontológica es el proceso de recuperar y mapear un modelo conceptual de una ontología implementada en otro modelo más adecuado, el cual sería reimplementado. Un método para la reingeniería ontológica fue presentado por el grupo de ontologías del Laboratorio de Inteligencia Artificial de la Universidad Politécnica de Madrid. Dicho método adapta el esquema de reingeniería de Chikofsky al dominio ontológico, teniendo tres actividades principales:

- Ingeniería inversa
- Reestructuración
- Ingeniería hacia adelante

### 1.3.5 Evaluación de ontologías

En (Welty and Guarino, 2001) se presenta una metodología para evaluar taxonomías. Esta metodología hace uso de una serie de principios filosóficos basados en los conceptos de rigidez, unidad, identidad y dependencia. En particular, el usuario realiza anotaciones en cada propiedad de la taxonomía. Por ejemplo, se puede especificar si hay algún criterio que identifique a cada entidad de la propiedad, si la propiedad depende de otra propiedad, etc. De esta forma, el

usuario evalúa la ontología teniendo en cuenta estas anotaciones. Usando esta metodología, el grupo de Guarino ha construido ontologías top-level de particulares y universales. Esta metodología de evaluación también da una serie de recomendaciones para construir ontologías:

- Etiquetar cada propiedad con metapropiedades
- Centrarse en las propiedades rígidas
- Evaluar la taxonomía teniendo en cuenta restricciones entre meta-propiedades
- Considerar las propiedades no rígidas
- Completar la taxonomía con otras propiedades

En (Gómez-Pérez, 2001) se presenta otra metodología de evaluación, ésta referida a la construcción correcta del contenido de la ontología, esto es, asegurar que sus definiciones implementan correctamente los requerimientos y cuestiones de competencia de la ontología. Se proponen los siguientes criterios para evaluar ontologías:

- Consistencia: Se refiere a obtener conclusiones contradictorias a partir de definiciones de entrada válidas. Una definición dada es consistente sí y solo si la definición individual es consistente y no se pueden inferir contradicciones usando otras definiciones y axiomas.
- Completitud: Este es un problema fundamental en las ontologías. Sólo se puede probar la incompletitud de una definición y, por tanto, la incompletitud de la ontología.
- Concisión: Una ontología es concisa si no contiene definiciones innecesarias o inútiles, redundancia explícitas y además no se pueden inferir redundancias usando otras definiciones y axiomas.

- Extensibilidad: Esta propiedad se refiere al esfuerzo necesario para añadir nuevas definiciones a una ontología y más conocimiento a las definiciones sin alterar el conjunto de propiedades bien definidas.

- Sensibilidad: Está relacionada con cómo afectan pequeños cambios al conjunto de propiedades bien definidas.

Esta metodología también se ocupa de los posibles errores realizados al estructurar el conocimiento del dominio en taxonomías, ontologías, y bases de conocimiento: errores de circularidad, errores de particiones de clases exhaustivas y no exhaustivas, errores de redundancia, errores gramaticales, errores semánticos y errores de incompletitud.

Esta metodología sugiere los siguientes pasos para construir ontologías:

- Evaluación de cada definición y axioma individual

- Evaluación de cada colección de definiciones y axiomas explícitamente en la ontología.

- Evaluación de las definiciones importadas de otras ontologías

- Evaluación de las definiciones que se pueden inferir usando otras definiciones y axiomas

### 1.3.6 Papel de las ontologías en el desarrollo de sistemas de información

En Inteligencia Artificial, la mayoría de metodologías existentes para construir ontologías de aplicación incluyen estos pasos:

- Construir la ontología del dominio
- Seleccionar los métodos de resolución de problemas (PSMs) para realizar las tareas
- Relacionar el conocimiento para PSMs con el dominio
- Añadir conocimiento factual del dominio

Algunos pasos (por ejemplo, los dos primeros) se pueden realizar en paralelo y es probable que los tres primeros se realicen varias veces hasta que converjan PSMs y conocimiento del dominio. Para el cuarto paso se suelen emplear herramientas de adquisición automática de conocimiento. El resultado final será la ontología de aplicación para el sistema de información en cuestión.

Por lo que respecta al papel desempeñado por las ontologías en el diseño de Sistemas de Información, dicho papel no es muy diferente al desempeñado por otros enfoques como modelado semántico, metadatos, análisis y diseño de patrones, y librerías de módulos software reusables. Sin embargo, existen diferencias entre esos enfoques y el modelado ontológico. Así, las ontologías de método son similares a módulos software reusables, aunque el modelado ontológico permite especificar relaciones entre tareas genéricas y del dominio, así como decir qué partes del conocimiento del dominio usamos. El modelado ontológico se parece más al modelado semántico, puesto que los diagramas E/R son una especificación conceptual de datos usados por un sistema de información que incluye conceptos, propiedades y relaciones entre entidades, permitiendo la definición de subclases y herencia. Un diagrama E/R puede ser expresado en una ontología, pero las ontologías permiten una especificación más rica del dominio de conocimiento.

Las ontologías también pretenden capturar metaconocimiento y presentan la ventaja frente al modelado en base a metadatos que se puede expresar el conocimiento de forma más estructurada. Por último, el análisis orientado a objetos y el diseño de patrones son similares a las ontologías del dominio, puesto que proporcionan descripciones de modelos usuales de

objetos a nivel de metaconocimiento. Las ontologías tienen la ventaja de que su conocimiento está a un nivel superior y no tienen las mismas limitaciones que los métodos orientados a objetos.

## 1.3.7 Uso e importancia de las ontologías

En esta sección discutiremos los dos usos más importantes de las ontologías (Chandresekaran et al, 1999), su uso como vocabulario y como teoría de contenidos. Por lo que respecta al primer uso, se puede ver la ontología como un vocabulario de representación. Bajo este punto de vista, las ontologías están generalmente especializadas en algún dominio. No es el vocabulario lo que califica a una ontología, sino las conceptualizaciones capturadas por los términos. La representación de un vocabulario proporciona un conjunto de términos con los que describimos hechos del dominio, mientras que el cuerpo de conocimiento que usa este vocabulario es una colección de hechos sobre el dominio. Sin embargo siguen existiendo contradicciones en el uso del término ontología. A veces, los teóricos usan el término para referirse a un conjunto de entidades y relaciones en un dominio.

Por otra parte, las ontologías son vistas como una teoría de contenidos. Uno de los intereses principales en ontologías es la relación entre teoría de contenidos y de mecanismos en IA. Las ontologías son esencialmente teorías de contenidos porque su contribución principal es identificar clases específicas de objetos y relaciones de un dominio.

El análisis ontológico aclara la estructura del conocimiento. Las ontologías son necesarias para obtener vocabularios para representar conocimiento. El primer paso para obtener un vocabulario y un sistema efectivo de representación de conocimiento es realizar un análisis ontológico del dominio. Por lo tanto, la clarificación terminológica permite que la ontología trabaje con fines coherentes y cohesivos. Además, las ontologías permiten la compartición de conocimiento. El resultado de un análisis ontológico incluye términos del dominio, términos generales y términos que describen comportamiento. Las ontologías capturan la estructura conceptual intrínseca del dominio. En un análisis ontológico, los términos se asocian con conceptos y relaciones en la ontología, por lo que necesitamos una sintaxis. Este lenguaje de representación de conocimiento puede ser compartido con otros con necesidades similares para representar conocimiento en el mismo dominio. Las ontologías compartidas pueden formar la base para lenguajes de representación de conocimiento específicos del dominio.

Otra razón para interesarse por las ontologías se muestra en (Berners-Lee, 2001). Las ontologías se consideran necesarias para el desarrollo de la Web Semántica, puesto que pueden mejorar el funcionamiento de la web de varias formas. Las ontologías pueden ser usadas para mejorar la efectividad de búsquedas web, porque el programa de búsqueda puede buscar sólo páginas referidas a conceptos específicos. Las páginas se enlazan a páginas de ontologías que definen información sobre el dominio. Toda la información la procesa un ordenador y podría ser usada para responder a preguntas que implicaran que el usuario rastreara el contenido de varias páginas devueltas por el motor de búsqueda. El uso de ontologías para la Web Semántica también es promovido en (Fensel and Musen, 2001), donde se considera que las ontologías son una tecnología clave para la Web Semántica. En (Maedche and Staab, 2001), los autores establecen que la definición de teorías del dominio compartidas y comunes puede ayudar a personas y máquinas a comunicar de forma concisa, facilitando el intercambio semántico. Así, la rápida y poco costosa construcción de ontologías específicas del dominio es crucial para el éxito y la popularidad de la Web Semántica.

## 1.4 DESARROLLO COLABORATIVO DE ONTOLOGIAS

Recientemente ha habido un creciente interés por las ontologías y en la idea de reutilizar ontologías del dominio existentes (Crow and Shadbolt, 2001). Asimismo, hay un alto nivel de consenso en considerar que se requiere un intenso trabajo organizacional así como costosos recursos humanos para construir ontologías. Por un lado, expertos e ingenieros del conocimiento deben trabajar conjuntamente para crear una ontología. Por otro lado, es necesario coordinar el trabajo de los expertos y gestionar los flujos de información entre los agentes humanos implicados en el proceso de construcción. En este sentido, se requiere un complejo proceso de elicitación entre expertos e ingenieros de conocimiento. Esta situación presenta bastantes problemas, tales como la necesidad de tener reuniones entre expertos e ingenieros de conocimiento, y establecer un consenso en la planificación de la tarea para construir una ontología (Hameed, Sleeman and Preece, 2001). Afortunadamente, la disponibilidad de los expertos puede ser resuelta en cierta medida por la explotación de las posibilidades ofrecidas por redes de comunicación a las que se conectan un conjunto de usuarios. De esta forma, los

nodos de usuarios pueden estar distribuidos espacial y temporalmente mientras resuelven entre ellos un cierto problema.

Dado este problema, creemos que la construcción de ontologías corporativas puede ser mejorada por la reutilización de ontologías ya existentes. Construir ontologías del dominio reutilizables no es tarea sencilla: debe haber un compromiso entre uso específico y reuso. Esto es, cuanto más ligada esté una ontología a un dominio específico y tarea, sus elementos terminológicos se pueden generalizar y reutilizar menos en otros dominios y tareas. Gruber y Guarino usan el término compromiso ontológico de la teoría de la lógica para elegir y compartir elementos de vocabulario de forma coherente y consistente entre tareas. Muchos autores han defendido el concepto de ontologías reutilizables como una meta alcanzable en representación del conocimiento. Varios de ellos han intentado mostrar que las ontologías del dominio pueden ser construidas para sistemas específicos en el cual la ontología se reutiliza en tareas o aplicaciones alternativas.

De esta forma, se reutilizarían las ontologías mediante la aplicación de procesos de integración de ontologías. En (Noy and Hafner, 1997), los procesos de integración son considerados beneficiosos para facilitar la reutilización y compartición del conocimiento. Según (Reimer, 1998), la integración de conocimiento puede ser analizada desde dos puntos de vista: integración de diversas bases de conocimiento e integración de diferentes representaciones del mismo conocimiento a diferentes niveles de formalización. Centrándonos en la segunda perspectiva trataríamos el problema de la integración de ontologías.

En la literatura no encontramos consenso con respecto a la definición de los procesos de integración, aunque la mayoría de autores consideren que la integración de ontologías no es tan solo una actividad sino un proceso completo. Asimismo se han propuesto diferentes definiciones para este proceso. En (Pinto and Martins, 2001), el proceso de integración se divide en los siguientes pasos:

(1) agregación

(2) combinación

(3) unificación de distintas ontologías fuente para crear la ontología resultante, posiblemente después de que haber modificado las ontologías reutilizadas.

Por otro lado, en (McGuiness, Fikes, Rice and Wilder, 2000), la integración de ontologías consiste en la iteración de los siguientes pasos:

(1) encontrar solapamientos en las ontologías;

(2) relacionar conceptos;

(3) chequear la consistencia, coherencia y no redundancia del resultado.

Sin embargo, la comunidad de la Web Semántica parece estar de acuerdo en los objetivos de dicho proceso, esto es, reutilizar las ontologías para construir modelos del dominio más completos para, por ejemplo, mejorar la semántica del contenido web. El trabajo en construcción colaborativa de componentes reutilizables de conocimiento (p.e.j, ontologías) se puede dividir en dos categorías: (1) entornos, algoritmos y herramientas para integración, alineamiento y fusión; y (2) desarrollo cooperativo de una ontología global. Sin embargo, el trabajo cooperativo da lugar a una serie de problemas (Fernández-Breis and Martínez-Béjar, 2000a), como pueden ser:

Información Redundante. Dos expertos describen la misma parte del dominio de la misma forma. Dada esta eventualidad, sería deseable que el sistema fuera capaz de gestionar este hecho para evitar redundancias.

Uso de términos sinónimos para un concepto. Diferentes expertos pueden emplear diferente terminología para definir un mismo concepto. En otras palabras, podría haber una correspondencia entre los términos empleados para un concepto dado (Shaw and Gaines, 1989). Durante el proceso de construcción de la ontología, la información relacionada con el uso de términos sinónimos para un concepto debe ser almacenada y gestionada, puesto que no se debe imponer una terminología a los expertos durante el proceso de adquisición de conocimiento. Sin embargo, una ontología tendería hacia un conocimiento consensuado, esto es, una terminología fija. Los sinónimos son posibles, pero lo ideal sería que se consensuara la terminología.

En el caso de los entornos para integración de diferentes ontologías tenemos un tercer problema a señalar, el de las inconsistencias entre piezas de conocimiento.

### 1.4.1 Sistemas de Integración de Ontologías

En (Shaw and Gaines, 1989) se comparan sistemas conceptuales. Allí, se construyen sistemas conceptuales basados en el acuerdo de los expertos que toman parte en el proceso. Entonces, se le asignan atributos a los conceptos, y a cada atributo cada experto le da valores individualmente. A continuación, los expertos intercambian y comparan sus resultados. Como resultado de esta comparación, se agrupan los conceptos en cuatro categorías, a saber, consenso, conflicto, correspondencia y contraste. Entonces, se deben resolver los conflictos o bien marcarlos como irresolubles. Este es un proceso manual y requiere participación activa de los expertos. Definamos los cuatro tipos de relaciones:

- Consenso: Equivalencia de conceptos.
- Conflictos irreparables: inconsistencias.
- Correspondencia: sinonimia.

- Contraste: aparece cuando los expertos definen partes independientes del mismo dominio.

La fusión de dominios a través de la fusión de ontologías se trata en (Wiederhold, 1994). Para resolver el problema de la sinonimia de conceptos propone un entorno que mantiene las terminologías originales asociadas a ontologías particulares. Sin embargo se asume que los términos nunca significan lo mismo a no ser que se diga explícitamente. Si se decide que dos conceptos significan lo mismo, se debe explicitar mediante una regla. Un conjunto de individuos estará encargado de gestionar dichas reglas, y estos colaboradores deben conocer la terminología asociada a cada posible dominio. Este enfoque se centra en los términos y partes comunes de los dominios implicados en la integración, y utiliza un álgebra con operadores binarios (unión, intersección y diferencia).

En (Visser et al, 1998) se presentó otro enfoque para integrar diferentes fuentes de conocimiento, de forma que se resuelven las diferencias ontológicas de forma manual utilizando un conjunto de funciones de utilidad similar a las reglas del enfoque anterior. Otro sistema es APECKS (Tennison and Shadbolt, 1998), donde se muestran las diferentes ontologías a los expertos para su comparación. De esta forma se muestran todas las ontologías a todos los usuarios, por lo que el usuario puede perderse entre tanta información, por la posible complejidad e inconsistencia eventual del elevado número de conceptos que se le muestran. El sistema propuesto en (Nakata et al, 1998) tiene un problema parecido, puesto que no hay restricciones de usuario para el acceso a partes del sistema en términos de seguridad por lo que cualquiera puede modificar el conocimiento de forma incorrecta.

A continuación presentamos una descripción de los algoritmos y sistemas de integración más importantes. Los tres entornos/herramientas a describir son PROMPT, FCA-Merge, y Chimaera.

### *1.4.1.1 PROMPT*

En (Fridman-Noy and Musen, 1999) se presenta un enfoque para integrar ontologías. En este enfoque se introducen dos procesos diferentes: fusión y alineamiento. La fusión es el proceso por el cual dos ontologías se unen para formar una sola; adicionalmente, ambas ontologías fuente deben cubrir dominios similares o solapados. El alineamiento implica enlazar conceptos entre las dos ontologías fuente. Por otra parte, el algoritmo SMART (Fridman-Noy and Musen, 1999; Musen and Fridman-Noy, 1999) fusiona conceptos cuyos nombres son lingüísticamente iguales o similares (por ejemplo, Unidad Militar y Unidad Militar Moderna). Sin embargo no se tienen en cuenta propiedades internas ni estructurales.

Estos autores presentan un nuevo algoritmo llamado PROMPT en (Fridman-Noy and Musen, 2000), que es una evolución de SMART y tiene en cuenta no sólo similaridades lingüísticas, sino también la estructura de la ontología para detectar conceptos a fusionar. En los sistemas basados en PROMPT, el usuario siempre decide qué conceptos son integrados puesto que fue diseñado para apoyar el proceso de integración, no para realizarlo por sí mismo. PROMPT tiene en cuenta diferentes características de las ontologías fuente para hacer sugerencias y buscar conflictos. Estas características incluyen:

- nombres de clases y ranuras (p.ej., si los frames tienen nombres similares y el mismo tipo entonces son buenos candidatos para la integración)

- jerarquía de clases

- asignación de ranuras a clases

- facetas y sus valores

Además de realizar sugerencias al usuario, PROMPT identifica conflictos, algunos de los cuales son descritos a continuación:

- conflicto de nombres (más de un frame con el mismo nombre),

- referencias obsoletas (un frame hace referencia a otro que ya no existe),

- redundancia en la jerarquía de clases (más de un camino de una clase a uno de sus padres distintos de la raíz),

- restricciones ranura-valor que violen la herencia.

En términos de ayuda al usuario, PROMPT tiene las siguientes características:

- **Establecer ontología preferida**. Suele suceder que las ontologías fuente no tienen la misma importancia o estabilidad, y que el usuario desee resolver todos los conflictos en beneficio de una ontología fuente. El usuario puede escoger una de las ontologías como la preferida. Entonces, cuando ocurre un conflicto de valores el sistema resolverá el conflicto automáticamente.

- **Mantener la posición de usuario**. Supongamos que un usuario une dos ontologías grandes y actualmente está trabajando sobre un área en particular. PROMPT mantiene la posición del usuario usando una lista de sugerencias y se presentan primero los ítems que incluyan frames relacionados con las últimas operaciones realizadas.

- **Informar al usuario**. Para cada sugerencia, PROMPT presenta una serie de explicaciones, empezando por la obtenida en primer lugar. Si PROMPT cambia la posición de una sugerencia en dicha lista, entonces también explicará el motivo de este cambio.

### *1.4.1.2 FCA-Merge*

FCA-Merge (Stumme and Maedche, 2001) es un enfoque bottom-up para integrar ontologías que ofrece una descripción estructural global del proceso de fusión. Para las ontologías fuente se extraen instancias de un conjunto dado de textos específicos del dominio por medio de la aplicación de técnicas de procesamiento de lenguaje natural. Se aplican técnicas basadas en matemáticas tomadas del Análisis Formal de Conceptos (FCA) (Wille, 1982; Ganter and Wille, 1999) para obtener un enrejado de conceptos como estructura final del proceso de FCA-Merge. Este resultado obtenido lo explora y transforma el ingeniero de conocimiento en la ontología integrada.

Esta metodología impone unas condiciones a los datos de entrada:

- Los documentos tienen que ser relevantes para cada una de las ontologías fuente. Un documento del que no se extraigan instancias para cada ontología fuente no es útil.

- Los documentos tienen que cubrir todos los conceptos de las ontologías fuente. Conceptos no cubiertos han de ser tratados manualmente tras el proceso de integración.

- Los documentos deben separar suficientemente los conceptos. Si dos conceptos considerados como diferentes aparecen siempre en los mismos documentos, FCA-Merge los mapeará al mismo concepto en la ontología final (a no ser que el ingeniero de conocimiento establezca otra cosa). Si esta situación se da con frecuencia, el ingeniero de conocimiento podría añadir más documentos para obtener una mejor diferenciación.

Esta metodología toma como datos de entrada las dos ontologías y un conjunto D de documentos en lenguaje natural. Los documentos pueden ser tomados de la aplicación objeto que requiere la ontología final. A partir de los documentos en D se extraen instancias. Este paso de adquisición automática de conocimiento devuelve, para cada ontología, un contexto formal que indica qué conceptos ontológicos aparecen en qué documentos. La extracción de instancias a partir de documentos es necesaria porque no existen generalmente instancias ya clasificadas

por ambas ontologías. Sin embargo, si se da esta situación, se puede obviar el primer paso y usar dicha clasificación de instancias.

El segundo paso de este enfoque contiene el algoritmo FCA-Merge, que unifica dos contextos y calcula un enrejado de conceptos a partir del contexto unificado usando técnicas de FCA. Más precisamente, calcula un enrejado de conceptos que tiene el mismo grado de detalle que las dos ontologías fuente. La extracción de instancias y el algoritmo FCA-Merge son procesos completamente automáticos. El paso final para derivar la ontología fusionada a partir del enrejado de conceptos requiere de la interacción humana. El ingeniero ontológico crea los conceptos y relaciones de la ontología final a partir del enrejado de conceptos y del conjunto de relaciones.

### *1.4.1.3 Chimaera*

El diseño de Chimaera (McGuiness et al, 2000) se basa en la experiencia obtenida al desarrollar otras interfaces para aplicaciones de conocimiento del mismo grupo. Se construyó sobre una plataforma que maneja cualquier sistema de representación basado en OKBC (Chaudhri, et. al, 1998). De esta forma, Chimaera acepta más de quince formatos de entrada, tales como ANSI KIF, Ontolingua, Protégé, CLASSIC, o XOL.

Chimaera sugerirá candidatos potenciales basados en un número de propiedades. Genera una lista de resolución de nombres que pueden ser usados como guía a través de la tarea de mezcla. Cuando se propone una sugerencia, se le muestra al usuario los lugares donde los dos términos aparecen en la jerarquía. El usuario puede explorar la jerarquía en más detalle mediante operaciones como expandir subclases. El usuario también puede ver las definiciones de los términos y obtener los resultados de comparaciones de similaridad y diferencia estructural de las definiciones. El usuario puede entonces elegir la integración de términos mediante una opción del menú.

Tiene un modo que apoya a la resolución taxonómica. Busca relaciones sintácticas tipo <X-Y> e <Y>. Cuando encuentra alguna, entonces busca relaciones de semánticas de

subsunción. Chimaera posee una capacidad analítica que permite a los usuarios ejecutar un conjunto de tests diagnósticos (incompletitud, chequeos sintácticos, análisis taxonómico, chequeo semánticos, etc) sobre las ontologías. La salida de muestra en forma de log interactivo que permite ver y explorar los resultados de estos tests.

### 1.4.2 Construcción Cooperativa de Ontologías

Podemos encontrar diferentes enfoques para la construcción cooperativa de ontologías. Todos ellos tienen en común la construcción de una ontología por un conjunto de usuarios, aunque difiere la forma en que se gestiona la cooperación. En esta sección se describirá un conjunto de dichos enfoques. Finalmente se discutirán en detalle tres de ellos, a saber, Ontolingua (Farquhar et al, 1997), WebOnto (Domingue, 1998) y OntoSaurus (ISX, 1991).

En (Euzenat, 1996) se presentó un sistema para la construcción cooperativa de bases de conocimiento consensuadas llamado CO4 (Cooperative Construction of Consensual knowledge bases). Se construyen ontologías en una jerarquía de bases de conocimiento, cada una representando el conocimiento consensuado por los usuarios suscritos a ese grupo. En la parte inferior de la jerarquía, cada usuario puede tener un número de bases de conocimiento que pueden modificar cuando deseen. Este sistema se basa en que, antes de introducir un elemento de conocimiento en la base de conocimiento, aquélla debe ser aceptada por una comunidad. Si todos los usuarios aceptan la propuesta, entonces se modifica la base de conocimiento. Para ello, se define un protocolo para enviar conocimiento, en el que se requiere una respuesta consensual de los miembros de la comunidad antes de modificar la base de conocimiento. Este principio garantiza la consistencia del conocimiento introducido y dirige el diálogo entre los expertos. Además, los usuarios deben usar la misma terminología.

En (García,1996), se presenta un sistema para construir una ontología de un dominio que se subdivide en subdominios homogéneos. Este enfoque es similar al anterior en cuanto a que la integración se hace de forma jerárquica, y trata el problema de la integración de conceptos similares pertenecientes a ontologías diferentes. La detección y definición de tales tipos de conceptos implica que cada usuario conoce la terminología del resto de usuarios, por lo que, de

esta forma, se pueden establecer enlaces entre estos conceptos. El principal problema de este enfoque es que el consenso necesario para enlazar conceptos similares incrementa el número de mensajes para la colaboración. Por otro lado, este proceso implica que los usuarios conozcan el vocabulario de los otros y que examinen las ontologías de otros.

En el Ontolingua Server (Farquhar et al, 1997), los conceptos se representan internamente como conceptos cualificados con respecto a la ontología a la que pertenecen. Asimismo, cuando un usuario manifiesta su intención de usar una ontología para extenderla, este sistema nunca permite que el usuario final añada información inconsistente con los contenidos de la ontología construida. Cada herramienta de trabajo basado en la cooperación incluye un componente que permite el diálogo entre los agentes implicados. En este sistema se permite que varios usuarios modifiquen la misma ontología a través del concepto de sesión compartida. También proporciona mecanismos que permiten comunicación vía email.

En Tadzebao y WebOnto (Domingue, 1998), y en KARAT (Abecker et al, 1998), el referido diálogo se considera como una de las actividades más importantes. Así, en KARAT se promueve el diálogo mediante la compartición de conocimiento, haciendolo visible a los usuarios. Además, el diálogo se establece en estos sistemas con el propósito de facilitar la discusión entre dos expertos diferentes sobre ontologías creadas con anterioridad.

En (Benjamins and Fensel, 1998) se presentó $(KA)^2$, una iniciativa para el desarrollo cooperativo de una ontología sobre investigación en Adquisición de Conocimiento. En este trabajo se presentó el uso de la World Wide Web (WWW) como un factor importante para el desarrollo cooperativo de una ontología. En primer lugar, la WWW representa la fuente de conocimiento más importante del mundo. En segundo lugar, la tecnología WWW permite la cooperación en procesos de construcción de ontologías, y la ontología está centralizada, mientras que sus instancias están distribuidas por la WWW. Finalmente, ONTOBROKER (Fensel et al, 1998) muestra como las memorias organizacionales pueden ser consultadas a través de la WWW por cualquier agente.

### *1.4.2.1 Ontolingua Server*

El Ontolingua server (Farquhar et al, 1996) fue un sistema de referencia para la comunidad ontológica. El objetivo de este servidor era facilitar el desarrollo cooperativo de ontologías. Como sugiere el nombre, la arquitectura consiste en un servidor central que usa el protocolo HTTP para comunicarse con usuarios remotos a través de un explorador web. Este servidor es un conjunto de herramientas y servicios que facilitan la construcción de ontologías compartidas entre grupos distribuidos y ha sido desarrollado por el Knowledge Systems Laboratory (KSL) de la Universidad de Stanford. La arquitectura del servidor de ontologías proporciona acceso a una librería de ontologías, traductores a lenguajes como Prolog, CORBA IDL, CLIPS, o Loom y un editor para crear y navegar ontologías. Los usuarios remotos pueden explorar y editar las ontologías, y las aplicaciones remotas y locales pueden acceder cualquiera de las ontologías de la librería usando el protocolo OKBC (Open Knowledge Based Connectivity). En Ontolingua, varios usuarios editan la misma ontología de forma simultánea y los cambios realizados por un usuario se hacen inmediatamente visibles para el resto de usuarios. El lenguaje Ontolingua permite que las ontologías se construyan por una de estas tres formas: (1) usando expresiones KIF; (2) usando exclusivamente el vocabulario de la Frame Ontology; y (3) usando ambos lenguajes.

### *1.4.2.2 WebOnto*

WebOnto (Domingue, 1998) fue diseñado para facilitar la navegación, creación y edición cooperativa de ontologías sin sufrir problemas de interfaz. En particular se diseñó WebOnto para proporcionar una interfaz de manipulación directa que muestra expresiones ontológicas. Es fácil de usar y tiene facilidades de escalabilidad. Soporta la navegación, creación y edición colaborativa de ontologías representadas en OCML. El enfoque cooperativo usado en WebOnto es similar al usado en Ontolingua pues los usuarios modifican la misma ontología. Cuando un usuario está modificando una ontología, los otros usuarios sólo pueden explorarla. WebOnto tiene una herramienta auxiliar llamada Tadzebao (Domingue, 1998), que permite realizar discusiones (a)síncronas sobre ontologías. Las características principales de WebOnto son:

- Gestión de ontologías usando un interfaz gráfico.

- Generación automática de instancias a partir de definiciones de clases, soporte al modelado de PSMs.

- Inspección de elementos, teniendo en cuenta la herencia de propiedades y el chequeo de consistencia.

Un interfaz completo que soporta trabajo colaborativo por medio de mensajes y anotaciones (usando Tadzebao).

### *1.4.2.3 Ontosaurus*

Ontosaurus (ISX, 1991) fue desarrollado por el Information Sciences Institute (ISI) de la Universidad de South California. Consiste en dos módulos: un servidor de ontologías, que usa Loom como sistema de representación de conocimiento, y un servidor de exploración de ontologías que crea páginas html dinámicamente para mostrar la jerarquía de ontologías. La ontología puede ser editada mediante formularios HTML. Además, existen traductores de Loom a Ontolingua, KIF, KRSS y C++.

En Ontosaurus, los usuarios cooperan para construir la misma ontología, de forma que el bloqueo es necesario, aunque proporciona la facilidad de enviar emails a otros usuarios. El método de bloqueo difiere del usado en Ontolingua o WebOnto porque los cambios no están disponibles inmediatamente para los otros usuarios, por lo que no pueden explorar la ontología hasta que todas las modificaciones hayan concluido.

Estos tres enfoques tienen cosas en común. Por un lado, requieren la participación de constructores de ontologías mientras se construya la ontología. Por otro lado, explotan la tecnología WWW para ofrecer un servicio colaborativo de edición y navegación de ontologías. Así, para aprovechar las propiedades “en cualquier momento, desde cualquier sitio” de la Web

creemos que un enfoque automático de integración sería más apropiado. Los modelos ontológicos podrían ser construidos a través de procesos de integración usando ontologías accesibles por el sistema sin la participación directa de los constructores de ontologías en el proceso.

## 1.5 LENGUAJES ONTOLÓGICOS

Los lenguajes ontológicos son vehículos para expresar ontologías de forma comprensible por las máquinas. Algunos de ellos han emergido en los últimos años en paralelo a la idea de Web Semántica, de forma que están orientados para tal tecnología. Sin embargo, los diferentes lenguajes han sido propuestos históricamente por los desarrolladores de herramientas ontológicas. En esta sección se presentan y discuten los lenguajes ontológicos más usados en la actualidad. La elección de lenguajes se basa en el estudio aparecido en (Gómez-Pérez and Corcho, 2002). En las siguientes secciones se presentan siete lenguajes ontológicos. Para cada uno de ellos se introducirá su origen y objetivos. A continuación se explicará su modelo de conocimiento subyacente.

### *1.5.1 SHOE*

El Simple HTML Ontology Extension (Luke and Heflin, 2002) fue desarrollado en la Universidad de Maryland como una extensión del HTML que incluye conocimiento semántico en documentos web. La última versión de este lenguaje adapta su sintaxis a XML. Con respecto al modelo ontológico subyacente, SHOE usa la siguiente terminología:

**Ontología**: Descripción de clasificaciones válidas para instancias y relaciones válidas entre instancias y elementos.

**Categoría (Clase):** Estos dos términos son intercambiables. Una categoría es un elemento bajo el cual se pueden clasificar instancias o subinstancias de páginas HTML. Los nombres de

las categorías son nombres de elementos y pueden estar prefijados. Las categorías pueden tener categorías padres, y definen herencia, incluyendo herencia múltiple: si una instancia se clasifica en una categoría, entonces claves clasificadas en esta categoría pueden rellenar posiciones en relaciones definidas por esa categoría o alguna de sus categorías padres.

**Instancia**: Es un objeto que puede ser clasificado en alguna categoría e incluido como argumento de relaciones. Las subinstancias son instancias declaradas dentro de otra instancia. Se puede incluso declarar instancias constantes.

**Relaciones**: Un elemento que define una relación entre instancias y otras instancias o datos. Los nombres de relaciones son nombres de elementos, y pueden ser prefijados. Una relación ocurre entre cero o varios elementos llamados argumentos; si una relación se define para un conjunto de argumentos, éste permite que los documentos SHOE declaren esta relación entre instancias de esos argumentos. Los argumentos se ordenan explícitamente.

**Reglas**: Definen clasificaciones o relaciones válidas en la ontología.

Finalmente, una especificación en SHOE podría ser la siguiente:

```
<DEF-CATEGORY NAME="Organization" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="Person" ISA="base.SHOEEntity">
<DEF-CATEGORY NAME="Student" ISA="Person">
<DEF-CATEGORY NAME="Worker" ISA="Person">
<DEF-CATEGORY NAME="Faculty" ISA="Worker">
<DEF-CATEGORY NAME="Professor" ISA="Faculty">

DEF-RELATION NAME="advisor">
     <DEF-ARG POS="1" TYPE="Student">
     <DEF-ARG POS="2" TYPE="Professor">
 </DEF-RELATION>
 <DEF-RELATION NAME="member">
     <DEF-ARG POS="1" TYPE="Organization">
     <DEF-ARG POS="2" TYPE="Person">
 </DEF-RELATION>
```

### *1.5.2 RDF(s)*

El Resource Description Framework (Lassila and Webick, 1998) fue desarrollado por el W3C con el objetivo de especificar contenido semántico, estandarizado, interoperable y basado en XML. Esta especificación muestra tres representaciones del modelo de datos, como tripletes, como grafo y en XML. Existen equivalencias entre las representaciones. El mapeo entre ellas no pretende limitar la representación interna de cada una de ellas. El modelo de datos RDF se define formalmente como sigue. Existe un conjunto llamado Resource, y también un conjunto de Literals. Hay un subconjunto de Resources llamados Properties. Por otro lado, tenemos un conjunto llamado Statements, donde cada elemento es un triplete de la forma{pred, sub, obj}, donde pred es un miembro de Properties, sub es un Resource y obj es o bien un Resource o un Literal.

Para un procesador RDF, los hechos son tripletes miembros de Statements. De esta forma, el Statement original sigue siendo un hecho a pesar de ser reificado, puesto que el triplete que representa al Statement original permanece en Statements. Se han añadido cuatro tipos de tripletes más. La propiedad "type" se define para proporcionar primitivas tipadas. Se define formalmente como sigue: Existe un elemento de Properties conocido como RDF:type. Los miembros de Statement de la forma {RDF:type, sub, obj} deben cumplir que sub y obk son miembros de Resources. Un RDFSchema impone restricciones adicionales en el uso del tipo.

Podemos definir formalmente la reificación como sigue: Existe un elemento de Resources, no contenido en Properties conocido como RDF:Statement. Existen tres elementos en Properties, conocidos como RDF:predicate, RDF:subject y RDF:object. La reificación del triplete {pred, sub, obj} de Statements es un elemento r de Resources que representa el triplete reificado y los elementos $s_1$, $s_2$, $s_3$, y $s_4$ de Statements tales que:

$s_1$: {RDF:predicate, r, pred} ; $s_2$: {RDF:subject, r, subj}

$s_3$: {RDF:object, r, obj} $s_4$: {RDF:type, r, [RDF:Statement]}

A continuación mostramos un ejemplo de porción de ontología RDF.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/metadata/dublin_core#"
  xmlns:dcq="http://purl.org/metadata/dublin_core_qualifiers#">
  <rdf:Description
about="http://www.dlib.org/dlib/may98/05contents.html">
    <dc:Title>DLIB Magazine - The Magazine for Digital Library
Research
      - May 1998</dc:Title>
    <dc:Description>D-LIB magazine is a monthly compilation of
     contributed stories, commentary, and
briefings.</dc:Description>
    <dc:Contributor rdf:parseType="Resource">
      <dcq:AgentType
      rdf:resource="http://purl.org/metadata/dublin_core_qualifiers#E
ditor"/>
      <rdf:value>Amy Friedlander</rdf:value>
    </dc:Contributor>
    <dc:Publisher>Corporation for National Research
Initiatives</dc:Publisher>
    <dc:Date>1998-01-05</dc:Date>
    <dc:Type>electronic journal</dc:Type>
    <dc:Subject>
      <rdf:Bag>
      <rdf:li>library use studies</rdf:li>
      <rdf:li>magazines and newspapers</rdf:li>
      </rdf:Bag>
    </dc:Subject>
    <dc:Format>text/html</dc:Format>
    <dc:Identifier>urn:issn:1082-9873</dc:Identifier>
    <dc:Relation rdf:parseType="Resource">
      <dcq:RelationType
      rdf:resource="http://purl.org/metadata/dublin_core_qualifiers#I
sPartOf"/>
      <rdf:value resource="http://www.dlib.org"/>
    </dc:Relation>
  </rdf:Description>
</rdf:RDF>
```

### *1.5.3 OML*

El lenguaje Ontology Markup (Kent, 1998) fue desarrollado en la Universidad de Washington y está parcialmente basado en SHOE. Entre los cuatro niveles existentes en OML, el Simple OML es el que mejor se ajusta a este análisis puesto que mapea su contenido directamente a RDF. Las ontologías OML representan conjunto de entidades de todos los tipos.

Así, es natural definir un tipo colección llamado Ontología y especificar ontologías como instancias del tipo Ontología. La ontología OML define este tipo entre otros, permitiendo la especificación de ontologías arbitrarias como instancias. En otras palabras, OML tiene ontologías objetos.

OML es un marco que usa tipos (categorías o clases) e instancias. Las instancias son llamadas tokens en Teoría de Flujos de Información. Instancias y tipos en OML pueden ser relaciones o funciones binarias, objetos, relaciones o valores de datos, colecciones o constantes. Puesto que se usan las ontologías como contenedores de tipos y se usan colecciones como contenedores de instancias, las instancias ontológicas no son sintácticamente similares a las ontologías, aunque semánticamente son equivalentes.

Las ontologías son especificadas en términos de una serialización, que puede ser intercambiada en un entorno distribuido como internet. La semántica de la ontología OML está especificada por el modelo OML básico. Una parte de una ontología OML sería como sigue:

```
<Ontology>
    <extends ontology="http://www.oml.org/ontology/"/>
    <Type.Object name="Label">
      <Type.Function name="name"   target.Type="String"/>
      <Type.Function name="street" target.Type="String"/>
      <Type.Function name="city"   target.Type="String"/>
      <Type.Function name="state"  target.Type="String"/>
      <Type.Function name="name"   target.Type="String"/>
      <Type.Function name="code"   target.Type="String"/>
    </Type.Object>
  </Ontology>
```

### *1.5.4 XOL*

XML-based Ontology Exchange Language (Karp et al, 2002) fue desarrollado por la comunidad bioinformática de EEUU para intercambiar ontologías entre sistemas heterogéneos. La base de este lenguaje es Ontolingua y OML, aunque su sintaxis es la XML. XOL difiere de Ontolingua en que tiene una sintaxis basada en XML; la semántica de OKBC-Lite es similar a

la de Ontolingua. Difiere de OML en que la semántica de OML se basa en grafos conceptuales, que es bastante diferente de OKBC-Lite.

El modelo de conocimiento OKBC-Lite proporciona definiciones precisas para lo que entendemos por conceptos, relaciones, objetos y limitaciones. Este modelo es capaz de expresar esquemas usados en varios tipos de sistemas de gestión de información tales como esquemas de bases de datos relacionales y orientados a objetos, y bases de conocimiento. También incluye tipos de ontologías menos complejos, incluyendo vocabularios controlados y taxonomías. Estos son los elementos principales de conocimiento en este modelo:

**Clases e individuos**: Una clase es un conjunto de entidades. Se dice que cada entidad de una clase es una instancia de la clase. Una entidad puede ser una instancia de otra clase. Una clase que tiene instancias que a su vez son clases se llaman metaclases.

**Ranuras propias, y facetas propias**: Una entidad tiene un conjunto de ranuras propias asociadas, y cada ranura propia de una entidad tiene asociado un conjunto de entidades llamadas valores de ranura. Una ranura propia de una entidad tiene asociada un conjunto de facetas propias, y cada faceta propia tiene asociada un conjunto de entidades llamadas valores de faceta.

**Ranuras plantilla y facetas plantilla**: Una clase tiene asociada una colección de ranuras plantilla que describen los valores de ranuras propias que tiene cada instancia de la clase representada por el frame. Se dice que los valores de las ranuras plantilla son heredados por las subclases y las instancias de una clase. Los valores de una ranura plantilla son heredadas por las subclases como valores de la misma ranura plantilla y por las instancias como valores de la correspondiente ranura propia. Una ranura plantilla de una clase tiene asociada una colección de facetas plantilla que describe los valores de facetas propias de las ranuras propias de cada instancia de la clase. Tal y como sucede con los valores de las ranuras plantilla, los valores de las facetas plantilla son heredadas por las subclases e instancias. Los valores de una faceta plantilla se heredan como subclases como valores de la misma faceta plantilla y en las instancias como valores de la correspondiente faceta propia.

Una muestra de ontología OML sería la siguiente:

```
<class>

<name>person</name>

<documentation>The class of all persons.</documentation>

</class>

<class>

<name>man</name>

<documentation>The class of all persons who define their gender as
male.</documentation>

<subclass-of>person</subclass-of>

</class>

<slot>

<name>year-of-birth</name>

<documentation>An integer that represents the year the person was
born.

</documentation>

<domain>person</domain>

<slot-cardinality>1</slot-cardinality>

<slot-numeric-min>1800</slot-numeric-min>

<slot-value-type>integer</slot-value-type>

</slot>

<slot>

<name>citizenship</name>

<documentation>Describes the current citizenship status of a person
with respect to their primary country of residence.</documentation>
```

```
<domain>person</domain>

<slot-value-type>(set-of citizen resident-alien permanent-
resident)</slot-value-type>

</slot>

<slot>

<name>has-father</name>

<documentation>Has-father(X,Y) holds when the father of X is Y. Has-
father and father-of are inverses, which is why their domain and slot-
value-type are reversed: all fathers are male, but the child of a
father may be of either gender. </documentation>

<domain>person</domain>

<slot-value-type>man</slot-value-type>

<slot-inverse>father-of</slot-inverse>

</slot>

<individual>

<name>John</name>

<instance-of>man</instance-of>

<slot-values>

<name>year-of-birth</name>

<value>1987</value>

</slot-values>

<slot-values>

<name>citizenship</name>

<value>permanent-resident</value>

</slot-values>

<slot-values>

<name>has-father</name>
```

```
<value>Carl</value>
</slot-values>
</individual>
```

### *1.5.5 OIL*

El Ontology Interchange Language (Horrocks et al, 2000) fue desarrollado como parte del proyecto OntoKnowledge. Pretende conseguir la interoperabilidad semántica entre recursos web. Su sintaxis se basa en lenguajes existentes tales como XOL o RDF(S)a. Una ontología OIL es una estructura compuesta por varios componentes, algunos de los cuales pueden ser estructuras, siendo algunos opcionales y pudiendo repetirse alguno. Cuando describimos ontologías en OIL debemos distinguir tres capas diferentes: (1) el nivel de objeto, donde se describen instancias concretas de la ontología;(2) el primer metanivel, donde se proporcionan las definiciones ontológicas actuales; y (3) el segundo metanivel, relacionado con la descripción de características de la ontología como autor, nombre, etc.

Aparte de los encabezamientos encapsulados en su contenedor, una ontología OIL consiste en un conjunto de definiciones tales como:

**Importar**: Lista de referencias a otros módulos OIL.

**Definiciones de Clase**: Una definición de clase **(class-def)** asocia un nombre de clase con una descripción de clase.

**Definiciones de Ranuras**: Una definición de ranura **(slot-def)** asocia un nombre de ranura con una descripción de ranura. Una descripción de ranura especifica las limitaciones globales a aplicar a la ranura.

**Base de Reglas**: Lista de reglas que se aplican a la ontología.

A continuación mostramos un trozo de ontología OIL.

```
ontology-definitions

slot-def eats
inverse is-eaten-by

slot-def has-part
inverse is-part-of
properties transitive

class-def animal

class-def plant
subclass-of NOT animal

class-def tree
subclass-of plant

class-def branch
slot-constraint is-part-of

class-def leaf
slot-constraint is-part-of
has-value branch

class-def defined carnivore
subclass-of animal
slot-constraint eats
value-type animal

class-def defined herbivore
subclass-of animal
slot-constraint eats
value-type
plant OR
(slot-constraint is-part-of has-value

class-def giraffe
subclass-of animal
slot-constraint eats
value-type leaf
class-def lion
subclass-of animal
slot-constraint eats
value-type herbivore

class-def tasty-plant
subclass-of plant
slot-constraint eaten-by
has-value herbivore, carnivore
has-value tree
```

### *1.5.6 DAML+OIL*

El lenguaje DAML+OIL (Horrocks and van Harmelen, 2001) fue desarrollado por un comité formado por miembros de la Unión Europea y los Estados Unidos en el contexto de DARPA Agent Markup Language (DAML). Fue construido sobre RDF(S), y comparte los mismos objetivos que OIL. DAML+OIL divide el universo en dos partes disjuntas. Una parte consiste en los valores que pertenecen a los tipos de datos del esquema XML. Esta parte se llama dominio de tipos de datos. La otra parte consiste en objetos que son considerados miembros de clases descritas en DAML+OIL (o RDF). Esta parte se llama dominio de objetos.

DAML+OIL trata la creación de clases que describen partes del dominio objeto. Tales clases se llaman clases de objetos y son elementos de daml:Class, una subclase de rdfs:Class. DAML+OIL también permite el uso de tipos de datos de esquemas XML para describir parte del dominio de tipos de datos. Estos tipos de datos se usan en DAML+OIL simplemente incluyendo sus URLs en una ontología DAML+OIL. Son elementos implícitos de daml:Datatype. Los tipos de datos no son objetos DAML+OIL individuales. A continuación describiremos los elementos principales del lenguaje

**Elementos de clase**: Un elemento de clase daml:Class contiene parte de la definición de una clase de objetos.

**Enumeraciones***:* Una enumeración es un elemento daml:one of, que contiene una lista de objetos que son las instancias. Esto nos permite definir una clase para enumerar exhaustivamente sus elementos. La clase definida por el elemento oneOf contiene exactamente los elementos enumerados.

**Propiedades**: Un elemento rdf:Property se refiere al nombre de una propiedad (una URL). Las propiedades que se usan en restricciones de propiedades deberían ser propiedades que relacionan objetos, y que son instancias de ObjectProperty, o propiedades de tipos de datos, que relacionan objetos a valores de tipos de datos, y que son instancias de DatatypeProperty.

**Restricciones de propiedad**: Una restricción de propiedad es un tipo especial de expresión de clase. Se define implícitamente una clase anónima, llamada clase de objetos que satisface la restricción.

**Instancias**: Instancias de clases y propiedades se escriben en el esquema sintáctico RDF.

A continuación se presenta un ejemplo de DAML+OIL.

```
<daml:Class rdf:ID="Animal">
        <rdfs:label>Animal</rdfs:label>
        <rdfs:comment>
          This class of animals is illustrative of a number
      of ontological idioms.
        </rdfs:comment>
      </daml:Class>

      <daml:Class rdf:ID="Male">
        <rdfs:subClassOf rdf:resource="#Animal"/>
      </daml:Class>

      <daml:Class rdf:ID="Female">
        <rdfs:subClassOf rdf:resource="#Animal"/>
        <daml:disjointWith rdf:resource="#Male"/>
      </daml:Class>

      <daml:Class rdf:ID="Man">
        <rdfs:subClassOf rdf:resource="#Person"/>
        <rdfs:subClassOf rdf:resource="#Male"/>
      </daml:Class>

      <daml:Class rdf:ID="Woman">
        <rdfs:subClassOf rdf:resource="#Person"/>
        <rdfs:subClassOf rdf:resource="#Female"/>
      </daml:Class>

      <daml:ObjectProperty rdf:ID="hasParent">
        <rdfs:domain rdf:resource="#Animal"/>
        <rdfs:range rdf:resource="#Animal"/>
      </daml:ObjectProperty>

      <daml:ObjectProperty rdf:ID="hasFather">
        <rdfs:subPropertyOf rdf:resource="#hasParent"/>
        <rdfs:range rdf:resource="#Male"/>
      </daml:ObjectProperty>

      <daml:DatatypeProperty rdf:ID="shoesize">
        <rdfs:comment>
```

```
    shoesize is a DatatypeProperty whose range is
xsd:decimal.
    shoesize is also a UniqueProperty (can only have
one shoesize)
  </rdfs:comment>
  <rdf:type
rdf:resource="http://www.w3.org/2001/10/daml+oil#Unique
Property"/>
  <rdfs:range
rdf:resource="http://www.w3.org/2000/10/XMLSchema#decim
al"/>
</daml:DatatypeProperty>

<daml:DatatypeProperty rdf:ID="age">
  <rdfs:comment>
    age is a DatatypeProperty whose range is
xsd:decimal.
    age is also a UniqueProperty (can only have one
age)
  </rdfs:comment>
  <rdf:type
rdf:resource="http://www.w3.org/2001/10/daml+oil#Unique
Property"/>
  <rdfs:range
rdf:resource="http://www.w3.org/2000/10/XMLSchema#nonNe
gativeInteger"/>
</daml:DatatypeProperty>
```

### *1.5.7 Comentarios Generales*

Según (Gómez-Pérez and Corcho, 2002), los lenguajes se pueden clasificar atendiendo al tipo de ontología a representar. Por ejemplo, para representar ontologías ligeras (esto es, sin axiomas) el orden sería: XOL, RDF(S), SHOE, OML, OIL, y DAML+OIL. Sin embargo, para representar ontologías pesadas no sería fácil ordenarlos. En este caso es necesario estudiar los requisitos de la aplicación/entorno para conocer si uno de los lenguajes es suficientemente expresivo y contiene todos los elementos necesarios.

# 1.6 GESTIÓN DE CONOCIMIENTO

## *1.6.1 Definiendo Conocimiento*

En primer lugar, deberíamos proporcionar una definición apropiada de conocimiento. En la literatura podemos encontrar muchas. Sin embargo, desde la perspectiva de Gestión de Conocimiento, se pueden señalar los siguientes aspectos:

*"El conocimiento incluye restricciones implícitas y explícitas entre objetos (entidades), operaciones y relaciones, que permiten recoger heurísticas generales y específicas así como los procedimientos de inferencias relacionados con la situación a modelar". (Sowa, 1984).*

*"El conocimiento es toda la información organizada que es aplicable a la resolución de problemas". (Woolf, 1990).*

*"El conocimiento es información organizada y analizada para hacerla comprensible y aplicable a la resolución de problemas y toma de decisiones" (Turban, 1992).*

*"El conocimiento se compone de verdades y creencias, perspectivas y conceptos, juicios y expectativas, metodologías y know-how" (Wiig, 1993).*

*"El conocimiento está formado por intuiciones, experiencias y procedimientos que son considerados correctos y que dirigen los pensamientos, comportamientos y*

*comunicaciones humanos" (van der Spek and Spijkervet, 1997).*

*"El conocimiento es razonar sobre información y datos para garantizar la operación, resolución de problemas, procesos de toma de decisiones, y aprendizaje en un modo activo" (Beckman, 1997a).*

*"Creencia justificada que incrementa la capacidad de una entidad para la acción efectiva" (Huber, 1991; Nonaka, 1994)*

### *1.6.2 Tipos de conocimiento*

Con el objeto de establecer una tipología sobre el conocimiento, se consideran diferentes aspectos. Así, según (Deutsch, 1999), hay varios factores implicados al describir las características tipológicas del conocimiento, esto es, factores atendiendo a las categorías de conocimiento que se pueden identificar.

Las características de los medios de almacenamiento de conocimiento son el primer factor. Ejemplos pertenecientes a esta categoría podrían ser el conocimiento guardado en la mente humana, documentos, conocimiento proveniente del uso de computadoras para construir sistemas informáticos para formalizar, compartir, estructurar y organizar tal conocimiento de forma apropiada, esto es, sistemas de representación de conocimiento.

La accesibilidad del conocimiento es un segundo factor a considerar. En este sentido, en (Nonaka and Takeuchi, 1995) se divide el conocimiento en dos categorías, tácito y explícito. Según estos autores, el conocimiento puede ser definido, categorizado y descrito atendiendo a factores tales como tipo de conversiones, propiedades estructurales y elementales, usos y propósitos. Así, estos autores sugieren los siguientes tipos fundamentales de conocimiento:

- **Conocimiento tácito**: Este conocimiento se corresponde con el conocimiento obtenido a través de la experiencia (por ejemplo habilidades), conocimiento simultáneo (relativo al aquí y ahora) y conocimiento análogo (tiene que ver con asuntos prácticos).

- **Conocimiento explícito**: Se corresponde con el conocimiento racional (en la mente), conocimiento secuencial (relativo al allí y entonces), y conocimiento digital (aspectos teóricos).

De esta forma, en (Nonaka and Takeuchi, 1995)) se desarrolló el ciclo de vida del conocimiento, que especifica los procesos mediante los cuales el conocimiento se transforma en un tipo de conocimiento diferente:

- **Socialización**: Compartición de conocimiento tácito entre individuos. El conocimiento permanece siendo tácito sin ser transformado en explícito. Este tipo de patrón no es muy interesante debido a su naturaleza tácita (Tácito → Tácito).

- **Articulación**: Alguien transforma el conocimiento tácito en explícito (Tácito → Explícito).

- **Síntesis**: Combinación de conocimiento explícito para crear nuevo conocimiento explícito (Explícito → Explícito).

- **Internalización**: Proceso de transformar conocimiento explícito en tácito (Explícito → Tácito).

En (Wiig, 1993), el flujo de conocimiento organizacional más importante es la transformación del conocimiento tácito en explícito, esto es, la articulación o socialización. Este proceso permite acumular conocimiento explícito, que es compartible y accesible por los miembros de la organización. Por el contrario, la internalización es el proceso natural llevado a cabo a través del aprendizaje individual por parte de los miembros de la organización, esto es, la asimilación de conocimiento. El tercer flujo de conocimiento relacionado con la transformación de conocimiento es la combinación o síntesis. En este caso se transfiere conocimiento a otra forma explícita de conocimiento. Este sería el caso de cambiar el formato de una base de conocimiento. Este tipo de flujo de conocimiento es importante para seleccionar, combinar y distribuir el conocimiento existente con diferentes fines. El cuarto flujo del ciclo de vida del

conocimiento permite transformar conocimiento tácito en otras formas de conocimiento tácito mediante procesos de socialización. Este proceso permite transferir conocimiento tácito de un experto a un ingeniero de conocimiento en una entrevista personal.

Además de esta distinción entre conocimiento tácito y explícito, otros autores proponen categorías adicionales. Por ejemplo, en (Liebowitz and Beckman, 1998) se establece un tercer nivel llamado conocimiento implícito. Concretamente se identifican estos tres niveles de accesibilidad del conocimiento:

- **Conocimiento Tácito** (en la mente humana y en la organización): Se accede de forma indirecta a través de procesos de extracción de conocimiento y observación de comportamientos.

- **Conocimiento Implícito** (en la mente humana y en la organización): Se accede mediante consulta y discusión. Requiere la localización y comunicación previa de conocimiento informal.

- **Conocimiento Explícito** (en documentos y sistemas informáticos): Se accede de forma rápida y documentada en fuentes formales de conocimiento.

En (Spender, 1996), se clasifica el conocimiento atendiendo a la naturaleza del conocimiento y los poseedores de tal conocimiento:

| | INDIVIDUAL | SOCIAL |
|---|---|---|
| EXPLICITO | CONSCIENTE | OBJETIFICADO |
| IMPLICITO | AUTOMATICO | COLECTIVO |

### *1.6.3 Principios de conocimiento*

En (Beckman, 1997a; Beckman, 1997b), podemos encontrar los siguientes principios de conocimiento atendiendo a las dimensiones principales de conocimiento. La primera se refiere al conocimiento en sí mismo y sus propiedades deseadas:

> *"El conocimiento formal y compartido, así como la experiencia son los factores clave para un mejor funcionamiento, agilidad y éxito de la organización"*

Por lo que respecta al conocimiento explícito y tácito,

> *"El conocimiento debe ser formalizado o hecho explícito para que pueda tener un valor importante para la organización"*

> *"El conocimiento formalizado es el único que puede ser electrónicamente representado, almacenado, compartido y aplicado de forma efectiva"*

### *1.6.4. Gestión de Conocimiento como disciplina*

La Gestión de Conocimiento es una disciplina reciente. El término Gestión de Conocimiento se debe a Karl Wiig, quien lo usó por primera vez durante una conferencia en Suiza (Wiig, 1997). Desde entonces se han realizado múltiples definiciones, de entre las cuales señalaremos las siguientes:

> *"La Gestión de Conocimiento aplica enfoques sistemáticos a la búsqueda, comprensión y uso del conocimiento para crear valor" (O'Dell, 1996)*

> *"La Gestión de Conocimiento implica la identificación y análisis del conocimiento necesario y disponible, así como la consiguiente planificación y control de acciones para*

*crear activos de conocimiento para satisfacer los objetivos de la organización" (Macintosh, 1996)*

*"La Gestión de Conocimiento proporciona el conocimiento adecuado para las personas adecuadas en el momento adecuado para que aquellas pueden tomar la mejor decisión" (Petrash, 1996)*

*"La Gestión de Conocimiento es la construcción y aplicación sistemática, explícita y deliberada de conocimiento para maximizar la efectividad organizacional con respecto al conocimiento, por lo que usa sus activos de conocimiento" (Wiig, 1997)*

*"La Gestión de Conocimiento es el proceso de capturar experiencia colectiva organizacional donde ésta resida (por ejemplo, bases de datos, documentos, mentes humanas) y su distribución allá donde pueda ayudar a mejorar los resultados" (Hibbard, 1997)*

*"La Gestión de Conocimiento es la gestión y control explícito del conocimiento en una organización para lograr los objetivos de la organización" (van der Spek and Spijkervet, 1997)*

*"La Gestión de Conocimiento es la formalización y acceso a la experiencia y al conocimiento práctico para crear nuevas capacidades, mejorar el rendimiento, fomentar la innovación y mejorar el valor de mercado" (Beckman, 1997a)*

De las definiciones anteriores podemos extraer la conclusión que la Gestión de Conocimiento implica establecer un proceso de búsqueda, acumulación, difusión y uso de conocimiento relevante para una organización.

Actualmente, la Gestión de Conocimiento (GC) es un factor clave en las organizaciones puesto que se tiende a evolucionar de empleados a trabajadores de conocimiento (Brooking,

1999; Muller and Schappert, 1999). El hecho es que las organizaciones se dan cuenta de que el conocimiento incrementa el valor de sus productos y servicios, además de proporcionar ventaja competitiva. De acuerdo a (Steels, 1993), los objetivos de la GC son facilitar la generación, comunicación y preservación de conocimiento en la organización. Existen varios tipos de conocimiento importantes para una organización. Deberíamos mencionar la identificación de funciones críticas de conocimiento y el conocimiento de quién sabe qué en la organización como factores más importantes.

Con respecto a las actividades implicadas en la GC como proceso, según (Dignum and Heimannsfeld, 1999) podemos enumerar las siguientes: i) identificación y mapeo de bienes intelectuales de la organización; ii) generación de conocimiento nuevo que permita obtener una ventaja competitiva, iii) recopilación accesible de información organizacional; y iv) compartición de buenas prácticas y tecnología, incluyendo técnicas de trabajo en grupo e intranets.

Este conocimiento organizacional debe ser mantenido de alguna forma en la organización, y de esta forma surge el concepto de Memoria Corporativa. El know-how se encuentra generalmente distribuido en la organización, de forma que debe ser integrado de forma coherente para facilitar el acceso al mismo y su reutilización, esto es, expresarlo en forma de memoria corporativa. Las memorias corporativas se consideran un elemento clave para gestionar el conocimiento porque facilitan la conservación, distribución y reutilización del conocimiento. En la literatura reciente podemos encontrar muchas definiciones para las memorias corporativas. En (Van Heijst et al, 1996) se define la memoria corporativa como una "representación de conocimiento e información organizacional explícita y persistente", mientras que en (Nagenda and Plaza, 1996) se define como "los recursos colectivos de datos y conocimiento de una compañía, incluyendo experiencias en proyectos, experiencia en resolución de problemas, etc". En (Abecker et al, 1998), una memoria corporativa es referida como "un contenedor que integra información contextual, documentos e información no estructurada, que facilita su uso, compartición y reutilización".

Su función principal sería mejorar la competitividad de las organizaciones por la forma en que se gestiona su conocimiento. Algunos autores consideran que las memorias corporativas son un mecanismo de enlace entre el conocimiento pasado y futuro en relación a los procesos y actividades que toman lugar en las organizaciones. En particular, en (Stein and Zwass, 1995), la memoria corporativa es vista como "medio a través del cual todo el conocimiento operativo acumulado en el pasado está disponible en el presente para ser usado en las actividades de la organización".

En (Caussanel and Chouraqui, 1999) los autores llegan al concepto de memoria corporativa a través del concepto de actividad en una empresa. Para estos autores, una memoria corporativa se compone de un conjunto de actividades y una actividad se define como en (Grunstein, 1995): "lo que la gente hace, hora tras hora, día tras día: finalmente, los empleados realizan sus tareas porque saben que pueden hacerlas, piensan que deben hacerlas, y todo esto implica la utilización de know-how específico". Por lo tanto, la creación de memorias corporativas es una tarea importante para facilitar la gestión de conocimiento. Un ejemplo de metodología para la construcción de memorias corporativas puede ser encontrado en (Campoy-Gómez,2002). Dicho enfoque se basa en el uso de ontologías y la explotación de la compartibilidad y reusabilidad de las mismas para construir memorias corporativas.

### *1.6.5 Tecnologías del Conocimiento*

Hoy en día vivimos en la Era del Conocimiento, donde el conocimiento se considera un activo fundamental, por lo que sería interesante señalar dos propiedades inherentes a la naturaleza del conocimiento. Según Martín-Rubio (1998), el conocimiento tiene dos propiedades "perversas":

- Si no se explica, permanece tácito
- Si no se comunica y comparte, se pierde

Con el objeto de resolver problemas derivados de estas dos cualidades, y dentro del área de la IA, han aparecido dos áreas tecnológicas, la Ingeniería de Conocimiento y la Gestión de Conocimiento, y ambas comparten una misma tecnología, la Tecnología del Conocimiento.

Sin embargo, la Tecnología del Conocimiento no surge como tal hasta 1982 con la introducción del "Nivel de Conocimiento" (Newell, 1982). En este momento, la Ingeniería del Conocimiento se enfrentó al problema de que la obtención de modelos de conocimiento estaba fuertemente ligada al lenguaje de representación utilizado. Este es, de hecho, un problema para reutilizar tales modelos de conocimiento. El Nivel de Conocimiento implica un nivel de abstracción independiente de la implementación, y permite realizar descripciones conceptuales de comportamiento para resolver problemas y de la estructura de conocimiento que soporta tal comportamiento (Newell, 1982).

Por encima de este nivel apareció un nuevo paradigma tecnológico, el Modelado de Conocimiento, que fue propuesto inicialmente en (Wielinga et al, 1992), y que constituye la base de las Tecnologías del Conocimiento actuales. Este paradigma se basa en cuatro principios que todo desarrollador de sistemas basados en conocimiento debe tener en cuenta. Estos cuatro principios, que describimos a continuación, han sido utilizados en la Ingeniería de Conocimiento moderna para diseñar y construir sistemas inteligentes:

1) **Limitación de Roles**. Supone un mecanismo para organizar el conocimiento limitando los roles que puede desempeñar un componente de conocimiento. Ejemplos de tales roles serían: observable, hipótesis, hipótesis validada, etc.

2) **Identificación de tipos de conocimiento.** Según este principio se identifican cinco tipos de conocimiento:

a) *Tareas*, que son la especificación de algunos objetivos, una meta, que puede ser expresada funcionalmente por medio de descripciones de entradas y salidas, así como dependencias lógicas entre ellas. Ejemplos de tales tareas son: diagnóstico, monitorización,

evaluación y planificación. Por ejemplo, la entrada de una tarea de monitorización podría ser la lectura de la información de estado, y una salida podría ser la activación de una alarma.

b) *Métodos*, que especifican diferentes caminos para resolver un problema particular, esto es, aquellos caminos para conseguir una meta asociada a una tarea particular. Como ejemplos podemos encontrar "proposición y revisión", "generación y testeo" y "especialización e inclusión".

c) *Inferencias,* que permiten describir el proceso de resolución de problemas como pasos de razonamiento elemental que actúan sobre elementos específicos de conocimiento que juegan el papel de entradas, y que generan nuevos componentes de conocimiento, que son las salidas. Por ejemplo, para el método "generación y testeo" aplicado a la tarea "diagnóstico", se podría usar una inferencia abductiva que opera aplicando el método "generación" sobre una pieza de conocimiento de rol "observable" y que permita obtener una nueva pieza de conocimiento que desempeñe el papel de salida.

d) *Ontologías*, que son vistas como modelos conceptuales del dominio en forma de descripciones estructuradas del dominio que se usan para resolver problemas. Puesto que un dominio de conocimiento puede ser conceptualizado de diferentes formas, podemos distinguir diferentes tipos de ontologías, tales como ontologías mereológicas, taxonómicas, topológicas, etc. De acuerdo con (O'Leary, 1999), las ontologías definen la terminología compartida utilizada en sistemas de GC para facilitar la comunicación, búsqueda, almacenamiento y representación de conocimiento. Son particularmente importantes para asegurar que las bases de datos de buenas prácticas pueden comunicar al usuario el rango más amplio posible de prácticas y actividades, así como permitir al usuario reconocer cuando una buena práctica es útil en su organización.

*e) Conocimiento del dominio*, que se entiende como una colección de conceptos, relaciones, y hechos relevantes del dominio para ser usados con diferentes roles por las inferencias en procesos de razonamiento.

3) **Compartibilidad y reusabilidad del conocimiento**. Se aplica a cualquiera de los cinco tipos anteriores de conocimiento. El paradigma de modelado encuentra aquí una limitación crítica, el problema de las interacciones relativas (Bylander and Chandrasekaran, 1998). Según este problema, la representación de conocimiento escogida para un problema en particular viene fuertemente determinada por la naturaleza del problema y de las estrategias de inferencia a aplicar, lo que va en contra de la reusabilidad. Sin embargo existen mecanismos para organizar componentes de conocimiento de forma que sean accesibles para su compartición y reutilización: las librerías de componentes. En esta dirección se han intentado construir, por un lado, librerías de tareas y métodos, y por la otra, librerías de ontologías. Sin embargo, también existen problemas con dichas librerías. En primer lugar, las tareas incluidas en las librerías de tareas no son realmente tareas genéricas, sino una combinación de tareas. Por lo tanto, los resultados dependen de la selección adecuada de una combinación de métodos de resolución. En segundo lugar, las librerías de ontologías para compartición y reuso tienen dos problemas principales: (1) puntos de vista diferentes producen conceptualizaciones diferentes; y (2) una ontología puede contener diferentes grados de especificidad/generalidad. Esto provoca una necesidad de mecanismos para mantener versiones y para integrar conocimiento ontológico. En particular, este último asunto ha sido uno de los motivos principales para la realización de esta tesis.

4) **Uso de modelos esquemáticos**. Con la finalidad de modelar el conocimiento requerido para una aplicación particular, es altamente beneficios disponer de un conjunto de modelos esquemáticos de los elementos de conocimiento implicados en el diseño de tareas, métodos y ontologías.

Aunque la construcción de sistemas inteligentes como sistemas autónomos y automáticos para la resolución de problemas presenta perspectivas interesantes, existen dominios de aplicación para los cuales las tecnologías del conocimiento requieren un enfoque distinto de la resolución de problemas clásica. Hay una creciente necesidad de gestionar grandes cantidades de conocimiento (en sus diferentes manifestaciones y fuentes) manejadas por las organizaciones.

### *1.6.6 Gestión de Conocimiento desde una Perspectiva Tecnológica*

Existen diferentes opiniones con respecto al valor de aplicar Tecnologías de la Información y la Comunicación (TIC) a Gestión del Conocimiento. En particular, algunos trabajos (Beckman (1997a); Liebowitz and Beckman (1998); Beckman (1998b)) afirman la importancia de aplicar tecnologías del conocimiento a gestión del conocimiento usando conceptos tales como representación de conocimiento, repositorios de conocimiento y transformación automática de conocimiento.

#### *1.6.6.1 Infraestructura TIC*

Según Beckman (1999), se necesita una infraestructura en TIC para facilitar el intercambio de conocimiento. Tal infraestructura permitiría un acceso rápido y comprensible al conocimiento y a la experiencia, de forma que ambos puedan ser usados. En este sentido, en (Tobin, 1998) se sugirió la construcción de una red de conocimiento usando TICs que incluyeran los siguientes componentes:

- Repositorio de conocimiento, generalmente una base de datos
- Directorio de fuentes de conocimiento
- Directorio de recursos de aprendizaje
- Herramientas de trabajo en grupo

Por otro lado, Beckman (1998b) presenta un modelo correspondiente a TICs está compuesto de los siguientes elementos:

- Una arquitectura y un estándar en TICs

- Una plataforma TIC hardware
- Una infraestructura de comunicaciones
- Interfaces
- Información y datos
- Aplicaciones software
- Soporte al usuario

A pesar de las diferencias existentes en el papel desempeñado por expertos y sistemas inteligentes en Gestión de Conocimiento, existe consenso en considerar el elevado valor de redes globales de computadores y tecnología groupware para compartir conocimiento, que es uno de los objetivos de esta tesis, proporcionar una herramienta por medio de la cual grupos de trabajo puedan construir componentes de conocimiento compartibles y reutilizables de forma cooperativa.

### *1.6.6.2 Esquemas de Representación de Conocimiento*

En (Liebowitz and Beckman, 1998) se presenta un esquema de representación de conocimiento que incluye los cinco niveles introducidos en las secciones anteriores. Estas estructuras de conocimiento pueden ser usadas para facilitar repositorios de conocimiento y lo que el autor llama Integrated Performance Support Systems. Por lo que respecta a los componentes del esquema de representación, en Beckman (1998b) podemos encontrar que el conocimiento puede ser representado en un sistema experto como casos, reglas y modelos. En el caso del razonamiento basado en casos, el conocimiento se representa en forma de eventos, problemas y soluciones para los casos específicos. En razonamiento basado en modelos se crea un marco general usando una terminología de objetos para representar y organizar dominios de

conocimiento en términos de atributos, comportamientos y relaciones entre objetos, así como procesos de simulación. Un último esquema sería la satisfacción de restricciones, donde se representa el conocimiento según una combinación de los enfoques anteriores.

#### *1.6.6.3 Repositorios de Conocimiento*

En la literatura podemos encontrar diferentes definiciones para los repositorios de conocimiento. En (Tobin, 1998), un repositorio de conocimiento organiza la información organizacional básica y proporciona a sus empleados productos, servicios, clientes y procesos. En (Beckman, 1998b), un repositorio de conocimiento es un almacén computerizado que tiene experiencias, prácticas, y documentación sobre cierto dominio. Para su creación se recupera, recoge e integra conocimiento, usando para ello todas las fuentes posibles. Se puede decir que existen pocos prototipos de repositorios de conocimiento que contienen diferentes tipos de conocimiento. Sin embargo, existen excepciones tales como trabajo en memorias corporativas (Jovanovic, 1998), donde diferentes estructuras de conocimiento tales como imágenes, sonido, texto, hipertexto, datos relacionales, casos, reglas, jerarquías de objetos, procesos y modelos.

#### *1.6.6.4 Integrated Performance Support Systems (IPSS)*

Además de la aplicación de sistemas expertos a áreas reducidas de aplicación, es posible diseñar Integrated Performance Support Systems para facilitar repositorios múltiples de conocimiento, con el objetivo de satisfacer las necesidades de usuario. Así, en (Winslow and Bramer, 1994) se cataloga este tipo de sistemas de acuerdo a los siguientes servicios:

- Infraestructura: Organización y estructura del entorno de trabajo.
- Control: Monitorización, coordinación y control.
- Navegación: Interacción hombre-máquina.

- Presentación: Posibilidad para personalizar datos y servicios.

- Adquisición: Captura conocimiento, casos, opiniones, aprendizaje y datos sensoriales en diferentes medios y su transformación en formato interno.

- Consultoría: Consultar servicios, asistencia y recordatorios.

- Instrucción: Ayuda y entrenamiento.

- Aprendizaje: Aplicación de técnicas de descubrimiento de conocimiento y minería dedatos.

- Evaluación: Valoración y certificación basada en medidas del rendimiento y la calidad.

- Referencia: Constituyen fuentes de conocimiento y experiencia para la organización.

### *1.6.6.5 Transformación de Conocimiento*

El último aspecto relacionado con la Gestión de Conocimiento desde un punto de vista tecnológico es la transformación automatizada del conocimiento usando un IPSS para incrementar el valor de los repositorios de conocimiento. Según (Beckman, 1997a), el conocimiento puede ser transformado por medio de la integración. Por ejemplo, datos y casos pueden ser transformados en reglas haciendo uso de técnicas de aprendizaje computacional y minería de datos. Sin embargo existen otras transformaciones posibles. Por ejemplo, en (Zarri, 1996) se muestra como texto no estructurado puede ser transformado en un formato tipo metadocumento, representado e indexado como redes semánticas, frames, y grafos conceptuales.

## 1.6.7 Sistemas de Gestión de Conocimiento

En IA, las bases de conocimiento son generadas para ser consumidas en sistemas expertos y basados en conocimiento, donde las computadoras usan inferencias para responder a cuestiones de usuario. Aunque es importante la adquisición de conocimiento para inferencias computacionales, en los desarrollos más recientes en Gestión del Conocimiento, el conocimiento queda disponible para consumo humano directo o para desarrollar software que procese dicho conocimiento.

Históricamente, la Gestión de Conocimiento se ha centrado en un único grupo a través de lo que generalmente se ha conocido como sistema de información ejecutiva (EIS), que contiene un conjunto de herramientas para acceder a bases de datos, generar alertas, etc para apoyar el proceso de toma de decisiones. Más recientemente, se ha comenzado a diseñar sistemas de Gestión de Conocimiento para organizaciones completas. Si los ejecutivos necesitan acceder a la información y al conocimiento, es probable que sus empleados tengan interés en esa información,

De acuerdo con O´Leary (1999), las principales funciones de un sistema de gestión de conocimiento son facilitar:

- La conversión de datos y texto en conocimiento;
- La conversión de conocimiento individual y de grupo en conocimiento accesible;
- La conexión de individuos y conocimiento a otros individuos y otro conocimiento;
- La comunicación de información entre diferentes grupos;
- La creación de nuevo conocimiento útil para la organización;

De esta forma, uno de los objetivos de esta tesis es proporcionar un sistema que facilite los procesos de gestión de conocimiento. Por lo tanto, el sistema tendrá que cubrir estos requisitos.

## 1.7 PROBLEMA A RESOLVER EN ESTE TRABAJO

Dentro de las tecnologías del conocimiento, las ontologías se están erigiendo en la tecnología más importante para representar el conocimiento. Una muestra de la importancia actual de las ontologías es la cantidad de metodologías que han surgido para su creación, como se ha podido ver en las secciones anteriores. Por un lado, existen metodologías para construir ontologías de forma individual, bien partiendo desde cero o por reingeniería. Por otro lado, también se pueden encontrar metodologías para la construcción cooperativa de ontologías, que a su vez se pueden dividir en desarrollo común de una misma ontología y desarrollo de una misma ontología común a partir de ontologías individuales, esto es, mediante integración de ontologías. La integración de ontologías se está convirtiendo en una actividad importante con diversas finalidades. El creciente interés en esta actividad se debe principalmente a dos motivos: (1) la necesidad de sistemas para la gestión de conocimiento; y (2) el desarrollo de la Web Semántica. Ambos campos comparten la necesidad de agrupar de alguna forma el conocimiento que se encuentra distribuido, tanto espacialmente como en cuanto a formato, en ambos contextos.

Por lo que respecta a los sistemas de gestión de conocimiento, el conocimiento en una organización reside tanto en documentos como en mentes humanas y sistemas informáticos. En cuanto a la Web Semántica, el conocimiento se encuentra distribuido principalmente en páginas web, las cuales pueden adoptar múltiples formatos. Por lo tanto, disponer de mecanismos de unificación/integración de ese conocimiento distribuido (representado mediante ontologías) es una necesidad en dichos ámbitos.

Sin embargo, si analizamos las metodologías disponibles de integración, nos encontramos con diversos problemas que impiden su implantación como soluciones plausibles. En primer lugar nos referiremos al modelo de conocimiento. La mayoría de metodologías de integración de ontologías utiliza relaciones taxonómicas exclusivamente, dejando de lado el resto de relaciones semánticas interconceptuales. Asimismo, en el caso de metodologías que permiten el uso de diferentes relaciones, su semántica no se expresa de manera formal. Consideramos que el modelo de conocimiento de una metodología para la integración de conocimiento debería contener más relaciones que únicamente la taxonomía. Además, la semántica y propiedades de las relaciones deben estar formalizadas.

Por otro lado, estas metodologías siguen teniendo inconvenientes importantes, a saber, la necesidad, durante todo el proceso de creación de la ontología, de la participación de los expertos, quienes puede que no estén disponibles en todo momento. Esto dificulta los procesos de creación cooperativa de conocimiento explícito. Por tanto, sería deseable disponer de una metodología que minimizara la implicación de los expertos en el proceso. Una solución podría contemplar la posibilidad de incluir mayor automatización en el proceso. Además, las metodologías para desarrollar una misma ontología presentan la desventaja de no disponer de una forma personalizada de acceder al conocimiento incluido en la ontología generada de forma cooperativa. Esto implica que se impone a los expertos que usan dichos sistemas una terminología en particular. Esto es inevitable en los sistemas que desarrollan una misma ontología común, puesto que los expertos modifican una misma ontología, de forma que se puede decir que se emplea una terminología conocida por todos ellos, porque todos los expertos la conocen al tener acceso a la ontología, aunque tampoco es una terminología consensuada (que sería el caso ideal). Por lo que respecta a los sistemas de integración, no contienen ninguna política para personalizar el conocimiento incluido en la ontología integrada. Sin embargo, sería deseable contar con un sistema que permitiera que cada experto accediera al conocimiento de manera personalizada, esto es, que se respetara la terminología empleada por dicho experto en la vista de la ontología construida cooperativamente.

Otro problema que tienen los sistemas de construcción cooperativa de ontologías es que el conocimiento incluido en la ontología resultante no se muestra a los usuarios de forma amigable, puesto que requieren ciertos conocimientos acerca de los que son las ontologías para

acceder al conocimiento. Sería deseable que este tipo de sistemas mostraran el conocimiento contenido en la ontología en una terminología cercana al usuario no docto en terminología ontológica.

En base a las carencias detectadas en las metodologías y sistemas de integración de ontología disponibles, el trabajo presentado en esta tesis tiene los siguientes objetivos:

- Definición y formalización de un modelo de conocimiento para representar conocimiento ontológico. Este modelo de conocimiento incluirá los siguientes elementos:

    - conceptos: son las entidades que definen el dominio de conocimiento.

    - atributos: son las propiedades que definen la estructura interna de los conceptos. Pueden tener una doble naturaleza: específicos o heredados. Los atributos específicos son aquellos atributos que se definen explícitamente en un concepto dado. Los atributos heredados son aquéllos que se derivan de las relaciones con otros conceptos del dominio.

    - relaciones interconceptuales: especifican la estructura externa de los conceptos, esto es, establecen qué conceptos del dominio están relacionados entre sí. Dos tipos de relaciones serán definidos y formalizados en este modelo de ontologías: mereología (o relación parte-de) y taxonomía (o relación es-un). Ambas implican dependencias jerárquicas entre los conceptos que toman parte de las mismas, de forma que dichos conceptos jugarán el rol de concepto padre o hijo de una relación taxonómica o mereológica. Además, la relación taxonómica provoca herencia de atributos, de forma que el concepto hijo heredará todos los atributos del concepto padre.

    - axiomas estructurales: son los axiomas del dominio que se pueden extraer directamente de la estructura de la ontología.

- Definición y formalización de una metodología para integrar ontologías especificadas según el modelo ontológico anterior. Este modelo de integración, apoyado por el modelo de ontologías se propone subsanar las carencias detectadas en los sistemas disponibles de integración de conocimiento. Por un lado, el modelo de conocimiento incluye dos tipos de relaciones interconceptuales formales. Asimismo, la participación de expertos del dominio en los procesos de integración se limita a proporcionar una ontología privada, puesto que las ontologías son integradas automáticamente. De esta forma, los expertos tampoco tendrán acceso directo al conocimiento creado por otros expertos. Además, los resultados de los procesos de integración serán adaptados a cada experto en particular. Esto implica que el proceso de integración tenga que tener en cuenta qué usuario ha realizado la petición de integración para adaptar los resultados a las preferencias de dicho usuario.

- Diseño e implementación de una aplicación software para el desarrollo cooperativo de ontologías. Este sistema se diseñará para cumplir con los requisitos que implica el entorno e integración descrito anteriormente, esto es, (1) los expertos se encuentran geográficamente distribuidos, por lo que utilizarán el sistema de integración de conocimiento a través de una intranet/internet; (2) la construcción de la ontología global se obtiene a través de procesos automáticos de integración; (3) cada experto obtendrá una visión personalizada del conocimiento incluido en la ontología resultante.

- Validación del entorno y sistema anteriores. La metodología de validación se puede resumir a continuación: (1) validación de la metodología de construcción de ontologías; (2) evaluación de la calidad de las ontologías obtenidas a través de los procesos de integración;(3) evaluación de la utilidad de los procesos de integración.

## 1.8 RESUMEN

En este capítulo se ha descrito el estado del arte en ingeniería ontológica. La primera parte se centra en el origen de la noción de ontología. Se presentaron varias definiciones desde puntos de vista diferentes. Las definiciones relacionadas con filosofía son más ambiciosas, puesto que las ontologías filosóficas consideran conceptos puros del entendimiento o de la razón. En IA se distinguen dos corrientes principales. Por un lado, se considera que las ontologías definen los términos básicos y relaciones de un dominio particular. Por otro lado, la comunidad filosófica de la IA defiende la noción filosófica original de ontología y usan las ontologías de manera más formal. De hecho, la ontología tiene un origen filosófico, puesto que Tales de Mileto fue el primero en mencionar esta idea al buscar una respuesta a la pregunta ¿de qué están compuestas las cosas? Desde entonces se han propuesto muchas teorías sobre la existencia. Se puede decir que la teoría ontológica más importante fue la teoría del ser de Aristóteles, propuesta en su Metafísica. Filósofos modernos tales como Leibniz o Kant hicieron aportaciones importantes para popularizar el concepto de ontología. Sin embargo, hasta la década de los noventa no se introdujo la ontología en IA. Los investigadores se dieron cuenta de la necesidad de formalizar el conocimiento para desarrollar sistemas inteligentes. Actualmente, las ontologías son la tecnología más extendida para representar conocimiento en IA, porque proporcionan una representación estructura y formal del conocimiento, y las representaciones ontológicas son compartibles y reutilizables.

Asimismo, se han propuesto diferentes clasificaciones de ontología siguiendo diferentes criterios. Una de tales clasificaciones distingue entre ontología formal y descriptiva. Mientras que las ontologías descriptivas están relacionadas con la recogida de información, las ontologías formales filtran, codifican y organizan los resultados de las ontologías descriptivas. En IA, las ontologías son pesadas cuando incluyen axiomas, y son ligeras en otro caso. Los axiomas son uno de los elementos usados para especificar ontologías. Otros elementos ontológicos importantes son: clases, relaciones, funciones e instancias. Se han propuesto diferentes modelos ontológicos. Dichos modelos se diferencian fundamentalmente en dos aspectos: (1) el tipo de ontología a modelar; y (2) la riqueza del modelo ontológico. Las relaciones son un elemento

importante del modelo ontológico puesto que son instrumentos para relacionar entidades ontológicas. Entre las posibles relaciones hemos distinguido dos como las más importantes, taxonomía (teoría de clasificación) y mereología (teoría de partes).

Debido a la creciente importancia de las ontologías, se han propuesto diferentes mecanismos y metodologías para diseñar y construir ontologías. Algunos de ellos fueron desarrollados para obtener ontologías particulares, mientras que otras metodologías tenían un propósito más general. Asimismo, se han tratado metodologías para la evaluación de la calidad de las ontologías construidas, puesto que el objetivo es diseñar ontologías buenas y útiles. Un aspecto importante a señalar es el papel, uso e importancia de las ontologías en el desarrollo de sistemas de información. Las ontologías se han usado principalmente como: (1) vocabulario y contenido a discutir; (2) teoría de contenido; (3) análisis ontológico para clarificar la estructura del conocimiento; y (4) estructurar conocimiento en la Web Semántica.

La parte más relevante de este capítulo para esta tesis es la revisión de diferentes metodologías para el desarrollo cooperativo de ontologías. Se identifican dos corrientes principales. Por un lado, diferentes agentes desarrollan cooperativamente la misma ontología. Esto tiene varias desventajas puesto que los agentes han de ser bloqueados cuando otros están modificando la ontología común, por lo que se requieren mecanismos de sincronización. Por otro lado, diferentes agentes desarrollan su propia ontología y las diferentes ontologías individuales son integradas. En este caso, los participantes no son bloqueados y se deben proporcionar mecanismos para integrar ontologías privadas. La generación de la ontología común es más difícil que en el enfoque anterior, puesto que posiblemente los agentes usen terminologías diferentes. La terminología está más controlada en el enfoque anterior puesto que se trabaja sobre una misma ontología.

La integración de ontologías es un proceso de importancia creciente en los últimos años. En un principio fue considerada como una actividad, mientras que actualmente se considera un proceso completo. Del análisis de los sistemas de integración de ontologías disponibles se desprende que ninguno de ellos es capaz de realizar integración automática, sino que es el usuario quien toma la decisión acerca de qué conceptos integrar. La mayoría de estos sistemas

sólo manejan relaciones taxonómicas, por lo que su modelo de conocimiento no es muy realista. Así, uno de los objetivos de esta tesis es proporcionar un marco para desarrollar ontologías de forma cooperativa a través de procesos de integración de ontologías. El desarrollo cooperativo de ontologías contribuye a obtener mejores ontologías puesto que éstas incorporarían conocimiento de varias fuentes expertas. Por otro lado, los lenguajes ontológicos más extendidos han sido brevemente descritos en este capítulo. En particular, los lenguajes escogidos han sido: XOL (XML-based Ontology Exchange Language), SHOE (Simple HTML Ontology Extension), OML (Ontology Markup Language), RDF(S) (Resource Description Framework), OIL (Ontology Interchange Language), y DAML+OIL. Finalmente, se han comentado aspectos relacionados con la Gestión del Conocimiento, puesto que las ontologías son una tecnología clave para realizar actividades de gestión de conocimiento y desarrollar sistemas para la gestión de conocimiento, fin último de esta tesis.

# CAPÍTULO 2

# FORMALIZACIÓN DEL MODELO ONTOLÓGICO

Este capítulo proporciona la formalización del modelo ontológico que ha sido propuesto y utilizado en esta tesis. El proceso de formalización se realizará en diferentes pasos. En primer lugar se presentarán los supuestos en los que se basa este modelo ontológico. Por lo tanto, se formalizarán todas las partes importantes de una ontología (atendiendo a este modelo ontológico). Dicha formalización es de elevada importancia para este trabajo puesto que las ontologías son el elemento utilizado para representar el conocimiento. A continuación, se formaliza todo lo descrito en las líneas precedentes.

## 2.1 SUPUESTOS BÁSICOS

Una ontología es vista en este trabajo como una especificación de una conceptualización de un dominio de conocimiento (van Heijst, Schreiber and Wielinga, 1997). Además, las ontologías serán representadas aquí mediante jerarquías múltiples de dominio restringido (MHRD) de forma similar al empleado por otros autores (ver, por ejemplo, Eschenbach and Heydrich (1995)). Más concretamente, denominamos MHRD al conjunto de conceptos que cumple las siguientes propiedades:

- Están definidos a través de un conjunto de atributos, de forma que no se tendrá en cuenta la presencia de axiomas entre dichos atributos.

- Existen relaciones taxonómicas entre los conceptos, de forma que se permite la herencia (múltiple) de atributos.

- Existen relaciones mereológicas entre conceptos.

Además, el esquema de representación ontológico usado en este trabajo incluye axiomas estructurales, esto es, axiomas extraídos de las relaciones concepto **tiene** atributo, concepto 1 **es**

**un tipo de** concepto 2 o concepto 1 **es una parte del** concepto 2. Este esquema también incluye otro tipo de axiomas (referidos como propiedades) derivados de algunas propiedades relacionadas con relaciones interconceptuales en organizaciones taxonómicas y mereológicas.

El hecho de definir las ontologías sin axiomas no estructurales no significa que dichos axiomas no puedan ser definidos como parte de la especificación de una conceptualización. Lo que se ha hecho en este modelo es dividir la definición clásica de ontología, que incluye axiomas estructurales y no estructurales, en dos partes de forma que llamamos ontología a la especificación completa de la conceptualización excluyendo axiomas no estructurales.

En lo concerniente a relaciones mereológicas entre conceptos, existe una gran variedad de teorías para construir una ontología mereológica que define la relación part-of y sus propiedades. Entre estas teorías, una de las más populares en la comunidad de Adquisición de Conocimiento es la Mereología Clásica Extensional (Simons, 1987; Borst, 1997). Desde el punto de vista de esta teoría, la transitividad supone que cuando un individuo es una parte propia de un segundo individuo que, a su vez, en una parte propia de un tercero, entonces el primero es también una parte propia del tercero. Sin embargo, la transitividad parece inapropiada para situaciones como "la cabeza de María es parte de María", y "María es parte de la Facultad de Informática". En este caso, podríamos deducir que "la cabeza de María es parte de la Facultad de Informática".

Por lo tanto, no se permite la transitividad en este modelo para organizaciones mereológicas, aunque asumiremos la presencia de otras propiedades mereológicas en el marco propuesto posteriormente en este trabajo. Así, las ontologías mereológicas que usaremos serán irreflexivas (nada es una parte de sí mismo), y asimétricas (si una cosa es una parte de otra, entonces la segunda no es parte de la primera). Por otro lado, aunque existen propiedades que pueden estar presentes en ontologías mereológicas (p.ej., consistencia, superposición, y disyunción), nos restringiremos a analizar organizaciones mereológicas tomando en cuenta las anteriores propiedades mereológicas y algunas otras propiedades nuevas basadas en atributos. Estas propiedades serán introducidas posteriormente.

## 2.2. FUNCIONES ONTOLÓGICAS

Con el objeto de asegurar que todos los conceptos de una jerarquía conceptual que conforman una ontología están organizados en jerarquías (múltiples) por medio de relaciones es-un (is-a) o parte-de (part-of) se proponen las siguientes funciones. Las definiciones serán ilustradas mediante la ontología que aparece en la Figura 2.1.

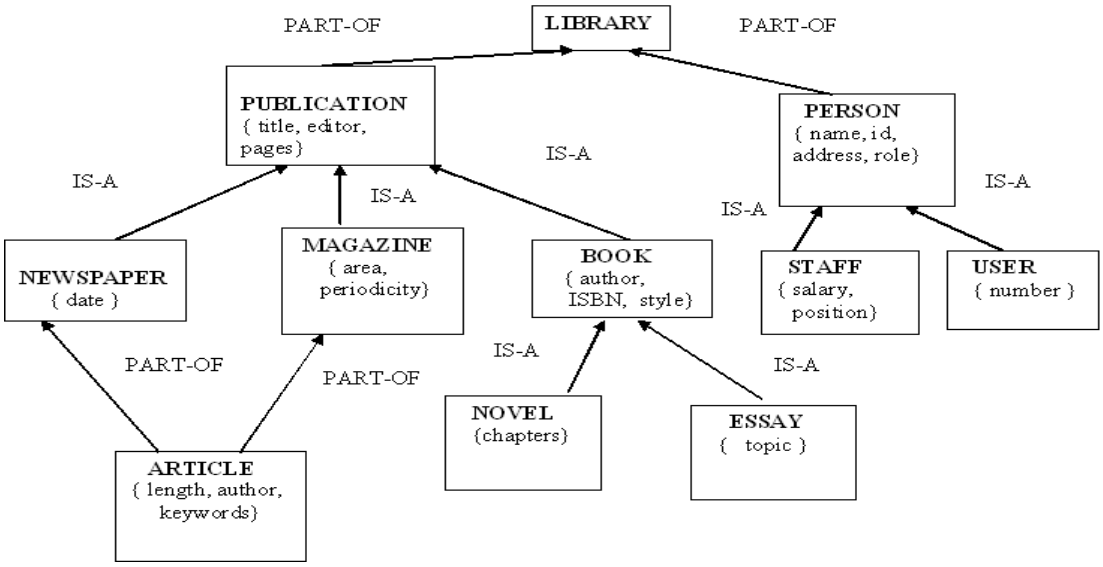


**Figura 2.1 Ontología de referencia**

En primer lugar, podemos definir una función para calcular las relaciones taxonómicas entre conceptos.

**Función TMHRD(t)**. Sea MHRD(t) un dominio múltiple jerárquico restringido hasta el instante t. El dominio múltiple jerárquico restringido taxonómico hasta el instante t, denotada por TMHRD(t), se define como el conjunto de conceptos que están interasociados en múltiples jerarquías por medio de relaciones IS-A. Formalmente, se define TMHRD(t) como el conjunto de conceptos que cumple:

[Cardinal(TMHRD(t)) ≥ 2] ⇔ [for all $c_i(t) \in$ TMHRD(t) exists $c_j(t) \in$ TMHRD(t) such that IS-A($c_i(t)$, $c_j(t)$) v IS-A($c_j(t)$, $c_i(t)$)], donde IS-A(a, b) significa que 'a es un tipo de b'.

En el ejemplo anterior (ver Figura 2.1), **TMHRD(t)=** {publication, person, newspaper, magazine, book, staff, user, novel, essay} porque existen las siguientes relaciones: IS-A(newspaper, publication); IS-A(magazine, publication); IS-A(book, publication); IS-A(staff, person); IS-A(user, person); IS-A(novel, book); IS-A(essay, book).

Análogamente se puede definir la siguiente función para calcular las relaciones mereológicas existentes:

**Función $M^2$HRD(t)**. Sea MHRD(t) un dominio múltiple jerárquico restringido hasta el instante t. El dominio múltiple jerárquico restringido mereológico hasta el instante t, denotado por $M^2$HRD(t), se define como el conjunto de conceptos que están interasociados en múltiples jerarquías por medio de relaciones PART-OF. Formalmente, se define $M^2$HRD(t) como el conjunto de conceptos que cumple:

[Cardinal($M^2$HRD(t))≥ 2] ⇔ [for all $c_i(t) \in M^2$HRD(t) exists $c_j(t) \in M^2$HRD(t) such that PART-OF($c_i(t)$, $c_j(t)$) v PART-OF($c_j(t)$, $c_i(t)$)], donde PART-OF(a, b) significa que 'a es una parte de b'.

Según el ejemplo anterior, $M^2$HRD(t)= {library, publication, person, newspaper, magazine, article} pues existen las siguientes relaciones: PART-OF(publication, library); PART-OF(person, library); PART-OF(article, newspaper); PART-OF(article, magazine).

En esta situación es sencillo establecer una función para calcular los conceptos padres e hijos taxonómicos de un concepto.

**Función T-parents/T-children**. Sea $c_i(t)$ un concepto perteneciente a un TMHRD(t) no vacío. El conjunto de *padres taxonómicos de $c_i(t)$ hasta el instante t*, denotado por *T-parents ($c_i(t)$),* se define como el conjunto $\{c_j(t) \in \text{TMHRD(t) tal que IS-A}(c_i(t), c_j(t))\}$. La función inversa para calcular los conceptos hijos se puede definir de la siguiente forma:: El conjunto de *conceptos hijos taxonómicos de* $c_i(t)$ *hasta el instante t*, denotado por *T-children($c_i(t)$),* se define como el conjunto $\{c_k(t) \in \text{TMHRD(t) tal que IS-A}(c_k(t), c_i(t))\}$.

Tomemos el concepto 'book' del ejemplo anterior como referencia para ilustrar ambas funciones, y calculemos las funciones T-parents (book) y T-children(book):

T-parents(book) = {publication} puesto que la relación IS-A (book, publication) existe en el dominio.

T-children(book) = {novel, essay} puesto que existen las siguientes relaciones en la ontología:

IS-A(novel, book); IS-A(essay, book)

A continuación podemos proceder de forma análoga para definir los conceptos padres e hijos en organizaciones mereológicas.

**Función M-parents/M-children**. Sea $c_i(t)$ un concepto perteneciente a un $M^2$HRD(t) no vacío. El conjunto de *padres mereológicos de $c_i(t)$ hasta el instante t*, denotado por *M-parents ($c_i(t)$),* se define como el conjunto $\{c_j(t) \in M^2\text{HRD(t) tal que PART-OF}(c_i(t), c_j(t))\}$.

M-parents(publication) = {library} puesto que PART-OF(publication, library).

La función inversa para conceptos hijos puede definirse de la siguiente forma: El conjunto de *conceptos mereológicos hijos de $c_i(t)$ hasta el instante t*, denotado por *M-children($c_i(t)$),*se define como el conjunto $\{c_k(t) \in M^2HRD(t)$ tal que PART-OF($c_k(t)$, $c_i(t)$)$\}$.

M-children(library) = {publication, person} puesto que PART-OF(publication, library) y PART-OF(person, library).

En general, un concepto puede pertenecer a organizaciones taxonómicas y mereológicas. Dicha posibilidad queda reflejada en la siguiente función:

**Función PMHRD(t)**. Sea TMHRD(t) un dominio múltiple jerárquico restringido taxonómico hasta el instante t. Sea $M^2HRD(t)$ un dominio múltiple jerárquico restringido mereológico hasta el instante t. El *dominio múltiple jerárquico restringo parcial hasta el instante t*, denotad por *PMHRD(t),* se define como el conjunto que resulta de TMHRD(t) $\cup$ $M^2HRD(t)$.

Según el ejemplo usado como referencia en esta sección:

PMHRD(t) = {publication, person, newspaper, magazine, book, staff, user, novel, essay} $\cup$ {library, publication, person, newspaper, magazine, article} = {publication, person, newspaper, magazine, book, staff, user, novel, essay, article}

Independientemente de las relaciones de un cierto concepto, puede ser necesario conocer los atributos conceptuales que no han sido heredados. Con tal propósito se usará la siguiente función:

**Función SPE**. Sea $c_i(t)$ un concepto perteneciente a un PMHRD(t) no vacío. El *conjunto de atributos específicos hasta el instante t*, denotado por *SPE($c_i(t)$)*, se define como el conjunto de atributos de $c_i(t)$ que no ha sido heredado de ningún otro concepto. Por ejemplo, el conjunto de atributos específicos del concepto 'book' es:

**SPE(book) =** {author, ISBN, style}**.**

De manera análoga, en algunas ocasiones se necesita conocer cuáles son los atributos heredados por un concepto. Para ello definiremos la siguiente función.

**Función INH-T**. Sea $c_i(t)$ un concepto perteneciente a un PMHRD(t) no vacío y sea $C_{ta}(t)$ un subconjunto de T-parents($c_i(t)$). Entonces, el *conjunto de atributos heredados por $c_i(t)$ de $C_{ta}(t)$*, denotado por *INH-T($c_i(t)$, $C_{ta}(t)$)* se obtiene de la siguiente forma:

$$\text{INH - T}(c_i(t), C_{ta}(t)) = \bigcup_{\substack{j=1 \\ c_j(t) \in C_{ta}(t)}}^{\text{Cardinal}(C_{ta}(t))} \text{SPE}(c_j(t))$$

En nuestro ejemplo, el conjunto de atributos heredados por el concepto 'book' es:

**INH-T(book) =** {title, editor, pages}**.**

Teniendo en cuenta ambos tipos de atributos, heredados y específicos, que un concepto puede tener en este esquema de representación de conocimiento, es sencillo obtener la siguiente función:

**Función ATT**. Sea $c_i(t)$ un concepto perteneciente a un PMHRD(t) no vacío. El *conjunto de atributos de $c_i(t)$ hasta el instante t*, denotado por *ATT($c_i(t)$),* se define como el resultado de INH-T($c_i(t)$, T-parents($c_i(t)$)) $\cup$ SPE($c_i(t)$).

El conjunto completo de atributos del concepto 'book' es:

**ATT(book) =** {title, editor, pages} $\cup$ {author, ISBN, style} = {author, editor, ISBN, pages, style, title}

Llegados a este punto es adecuado hacer un resumen de las funciones ontológicas definidas hasta el momento. La Tabla 2.1 contiene las funciones que pueden ser aplicadas a una única ontología para obtener sus distintas subestructuras. La Tabla 2.2 resume las funciones que pueden ser aplicadas a un concepto de una ontología.

| Función | Descripción informal |
|---|---|
| **TMHRD** | **Calcula el conjunto de conceptos relacionados taxonómicamente** |
| **$M^2$HRD** | **Calcula el conjunto de conceptos relacionados mereológicamente** |
| **PMHRD** | **Calcula el conjunto de conceptos relacionados taxonómica o mereológicamente** |

**Tabla 2.1 Funciones para una ontología**

| Función | Descripción informal |
|---|---|
| **T-parent** | **Calcula los padres taxonómicos de un concepto** |
| **T-children** | **Calcula los hijos taxonómicos de un concepto** |
| **M-parent** | **Calcula los padres mereológicos de un concepto** |
| **M-children** | **Calcula los hijos mereológicos de un concepto** |
| **SPE** | **Calcula los atributos específicos de un concepto** |
| **INH-T** | **Calcula los atributos heredados por un concepto** |
| **ATT** | **Calcula todos los atributos (heredados y específicos) de un concepto** |

**Tabla 2.2 Funciones para un concepto**

Finalmente, se puede definir una función que calcula el número de atributo que dos conceptos tienen en común:

**Función degree_of_overlapping**. Sean $c_i(t)$ y $c_j(t)$ dos conceptos diferentes pertenecientes a un PMHRD(t). El *grado de solapamiento entre $c_i(t)$ y $c_j(t)$ en el instante t*, denotado por *degree_of_overlapping($c_i(t)$,$c_j(t)$),* se define como el cardinal del conjunto resultado de $ATT(c_i(t)) \cap ATT(c_j(t))$.

Ahora podemos proceder a calcular al grado de solapamiento entre los conceptos 'book' y 'person'.

**ATT(book) =** {author, editor, ISBN, pages, style, title}

**ATT(person) =** {name, id, address, role}

degree_of_overlapping(book, person)= {author, editor, ISBN, pages, style, title} ∩ {name, id, address, role} = ∅. Por lo tanto, el grado de solapamiento entre ambos conceptos es 0.

Calculemos a continuación el grado de solapamiento entre 'book' y 'newspaper'.

**ATT(book) =** {author, editor, ISBN, pages, style, title}

**ATT(newspaper) =** {date, editor, pages, title}

degree_of_overlapping(book, newspaper)= {author, editor, ISBN, pages, style, title} ∩ {date, editor, pages, title}= {editor, pages, title}. Por lo tanto, el grado de solapamiento entre ambos conceptos es 3.

Algunas propiedades de las relaciones interconceptuales que existen en una ontología pueden ser definidas a partir de algunas de las funciones definidas hasta el momento:

- For all c(t), c'(t) $\in$ $M^2$HRD(t), c(t) $\neq$ c'(t), [c'(t) $\in$ M-parents(c(t)) $\rightarrow$ c(t) $\notin$ M-parents(c'(t))] (asimetría mereológica)

Esta propiedad implica que si existe la relación PART-OF(Article, Newspaper), entonces la relación PART-OF(Newspaper, Article) no es válida.

- For all c(t), c'(t) $\in$ TMHRD(t), c(t) $\neq$ c'(t), [c'(t) $\in$ T-parents(c(t)) $\rightarrow$ c(t) $\notin$ T-parents(c'(t))] (asimetría taxonómica)

Esta propiedad significa que la existencia de la relación IS-A(Essay, Book) imposibilita la existencia de la relación IS-A(Book, Essay).

- For all c(t), c'(t), c''(t) $\in$ TMHRD(t), c(t) $\neq$ c'(t) $\neq$c''(t), [(c'(t) $\in$ T-parents(c(t))) $\wedge$ (c''(t) $\in$ T-parents(c'(t)))$\rightarrow$ (c''(t) $\in$ T-parents(c(t)))] (transitividad taxonómica)

Según esta propiedad, si existen las relaciones IS-A(Essay, Book) e IS-A(Book, Publication), entonces se puede inferir la relación IS-A(Essay, Publication).

- For all c(t), c'(t), c''(t) $\in$ $M^2$HRD(t), c(t) $\neq$ c'(t) $\neq$c''(t), not[(c'(t) $\in$ M-parents(c(t))) $\wedge$ (c''(t) $\in$ M-parents(c'(t)))$\rightarrow$ (c''(t) $\in$ M-parents(c(t)))] (intransitividad mereológica)

De acuerdo con esta propiedad, la existencia de las relaciones PART-OF(Page, Article) y PART-OF(Article, Newspaper) no implica la existencia de la relación PART-OF(Page, Newspaper).

- For all c(t) ∈ $M^2$HRD(t), c(t) ∉ M-parents(c(t)) (irreflexividad mereológica)

Esta propiedad implica que la relación PART-OF(Article, Article) no es válida.

- For all c(t) ∈ TMHRD(t), c(t) ∉ T-parents(c(t)) (irreflexividad taxonómica)

Esta propiedad implica que la relación IS-A(Article, Article) no es válida.

A continuación definiremos cuatro axiomas que limitan la existencia de tipos diferentes de relaciones entre dos conceptos. Formalmente:

- For all c(t), c'(t) ∈ PMHRD(t), [c'(t) ∈ T-parents(c(t)) → c'(t) ∉ (T-children(c(t)) U M-parents(c(t)) U M-children(c(t)))] (axioma de exclusividad de predecesor taxonómico)

Esta propiedad establece que si existe la relación IS-A(Essay, Book), entonces las relaciones IS-A(Book, Essay), PART-OF(Essay,Book) y PART-OF(Book, Essay) no son válidas.

- For all c(t), c'(t) ∈ PMHRD(t), [c'(t) ∈ T-children(c(t)) → c'(t) ∉ (T-parents(c(t)) U M-parents(c(t)) U M-children(c(t)))] (axioma de exclusividad de sucesor taxonómico)

De acuerdo con esta propiedad, si existe la relación IS-A(Essay, Book), entonces las relaciones IS-A(Book, Essay), PART-OF(Essay,Book) y PART-OF(Book, Essay) no pueden existir.

- For all c(t), c'(t) ∈ PMHRD(t), [c'(t) ∈ M-parents(c(t)) → c'(t) ∉ (M-children(c(t)) U T-parents(c(t)) U T-children(c(t)))] (axioma de exclusividad de predecesor mereológico)

Según esta propiedad, la existencia de la relación PART-OF(Article, Newspaper) imposibilita la de las relaciones IS-A(Article, Newspaper), IS-A(Newspaper,Article) y PART-OF(Newspaper, Article).

- For all c(t), c'(t) $\in$ PMHRD(t), [c'(t) $\in$ M-children(c(t)) $\rightarrow$ c'(t) $\notin$ (M-parents(c(t)) U T-parents(c(t)) U T-children(c(t)))] (axioma de exclusividad de sucesor mereológico)

Esta propiedad establece que si existe la relación PART-OF(Article, Newspaper), las relaciones IS-A(Article, Newspaper), IS-A(Newspaper,Article) y PART-OF(Newspaper, Article) no son posibles.

Llegados a este punto, podemos definir una propiedad que establezca que si dos conceptos están relacionados mediante una relación part-of, y ambos conceptos están relacionados a un tercer concepto $c_k$ por medio de una relación is-a, entonces ambos conceptos heredan las propiedades definidas en $c_k$. Esto se puede expresar formalmente de la siguiente manera:

- For all c(t), c'(t) $\in$ PMHRD(t) [c'(t) $\in$ M-parents(c(t)) $\wedge$ (T-parents(c(t)) $\cap$ T-parents(c'(t)) $\neq \varnothing$) $\rightarrow$ (degree_of_overlapping(c(t), c'(t)) > 0]

En este caso, si existen las relaciones PART-OF(Page, Article), PART-OF(Author, Article), IS-A(Page, Component) e IS-A(Author, Component), entonces degree of overlapping(Page, Author) >0, puesto que tendrán algún atributo común (al menos los heredados de Component).

Cuando dos conceptos no tienen ningún padre taxonómico común ni ningún atributo específico común, se puede decir que no tienen nada común. Formalmente:

- For all c(t), c'(t) ∈ PMHRD(t) [(SPE(c(t)) ∩ SPE(c'(t)) = ∅) ∧ (T-parents(c(t)) ∩ T-parents(c'(t)) = ∅) → (degree_of_overlapping(c(t), c'(t)) = 0)]

## 2.3 UNA FORMA COMPLEMENTARIA DE DEFINIR ONTOLOGIAS

Las nociones de ontología y concepto se pueden expresar de la siguiente forma:

**Definición 1**: Una **ontología**, denotada por O(t), es vista como el triplete < C,R, P >, donde:

- C = PHMRD(t), esto es, un conjunto no vacío de conceptos.
- R = {∅, IS-A, PART-OF}, esto es, el conjunto de relaciones que se pueden establecer entre dos conceptos.
- P: CxC → R, esto es, una función que establece las relaciones entre dos conceptos.

Estas relaciones C, R, y P pueden ser identificadas en la ontología mostrada en la Figura 2.1. Concretamente, R permanence fijo para todas las ontologías de este modelo.

C = {library, publication, person, newspaper, magazine, book, staff, user, novel, essay, article}

P= {(publication, library, PART-OF); (person, library, PART-OF); PART-OF(article, newspaper, PART-OF); (article, magazine, PART-OF); (newspaper, publication, IS-A); (magazine, publication, IS-A); (book, publication, IS-A); (staff, person, IS-A); (user, person, IS-A); (novel, book, IS-A); (essay, book, IS-A)}.

**Definición 2:** Un **concepto**, denotado por c(t) es visto como el par < A, Re >, donde:

- A=ATT(c(t)), esto es, su conjunto de atributos.
- Re= {c'(t) ∈ C / P(c(t), c'(t))≠ ∅ }, un conjunto de conceptos con los que está relacionado.

Una vez realizado esto podemos proceder a definir cada concepto de la ontología usando este formalismo.

Library

A= { }

Re= { publication, person}

Publication

A= {title, editor, pages}

Re= { library, newspaper, magazine, book}

Person

A= {name, id, address, role}

Re= {library, staff, user}

Newspaper

A= { date, title, editor, pages}

Re= { publication , article}

Magazine

A={area, periodicity, title, editor, pages}

Re={publication, article}

Book

A={ author, ISBN, style, title, editor, pages}

Re={publication, novel, essay}

Staff

A={ salary, position, name, id, role, address}

Re={ person}

User

A={ number, name, id, role, address}

Re={ person}

Novel

A={ chapters, author, ISBN, style, title, editor, pages}

Re={ book }

Essay

A={ topic, author, ISBN, style, title, editor, pages}

Re={ book }

Article

A= { length, author, keywords}

Re={newspaper, magazine}

## 2.4 ONTOLOGÍAS COMO SISTEMAS DE INFORMACIÓN

Según (Winskel, 1994), un sistema de información o dominio de Scott, consiste en un conjunto de tokens (p.ej., asertos, proposiciones) que se puedan idear para una computación, y que estén relacionados por consistencia y encadenamiento. Formalmente, un sistema de información se define como la estructura A= (A, Con, $\vdash$), donde A es un conjunto contable, Con es un conjunto no vacío de subconjuntos finitos de A, denotado por fin(A) ( fin(A)={X / X $\subseteq^{fin}$ A} ), y $\vdash$ es un subconjunto de (Con\{$\varnothing$}) x A que satisface los siguientes axiomas:

1. $X \subseteq Y \in Con \Rightarrow X \in Con$

2. $a \in A \Rightarrow \{a\} \in Con$

3. $X \vdash a \Rightarrow X \cup \{a\} \in Con$

4. $X \in Con \ \& \ a \in X \Rightarrow X \vdash a$

5. $(X,Y \in Con \ \& \ \forall b \in Y.\ X \vdash b \ \& \ Y \vdash c) \Rightarrow X \vdash c.$

Los elementos, |A|, de un sistema de información A= (A, Con, $\perp$) son aquellos subconjuntos x de A que cumplen las siguientes propiedades:

1. no vacíos : $x \neq \emptyset$.

2. consistentes: $X \subseteq^{fin} x \Rightarrow X \in Con$.

3. $\perp$-cerrado: $X \subseteq x \ \& \ X \perp a \Rightarrow a \in x$.

De esta forma, un sistema de información determina una familia de conjuntos. Estas familias forman cpos (conjuntos parcialmente ordenados) cuando se ordenan por inclusión. Además, los sistemas de información se pueden considerar clases especiales de locales para topologías "pointless" en las que se toman como elementos primarios vecindarios en vez de puntos. Esta visión se usa tanto en topología como en lógica.

Una vez introducidos tanto los sistemas de información como nuestro modelo ontológico, ya podemos demostrar que las ontologías construidas de acuerdo a este modelo son sistemas de información. Esta propiedad facilitaría la explotación de las ontologías construidas. Con ese propósito se han formulado las siguientes cuatro proposiciones que identifican la estructura A=(A, Con, $\perp$) en nuestro modelo. Las proposiciones 1-3 prueban que las ontologías construidas con nuestro modelo no están vacías, son consistentes y $\perp$-cerradas, mientras que la proposición 4 prueba que estas ontologías son sistemas de información.

**Proposición 1**: Sea PHMRD(t) un conjunto contable. Además, PHMRD(t) es el conjunto de los conceptos ontológicos de una ontología dada. Entonces, PHMRD(t) es no vacío.

*Prueba:* Los subconjuntos de PHMRD(t) son conceptos ontológicos simples. Entonces, por definición, un concepto ontológico no puede ser vacío por lo que PHMRD(t) no puede contener elementos vacíos.

**Proposición 2**: Sea PHMRD(t) un conjunto contable no vacío. Entonces, PHMRD(t) es consistente.

*Prueba*: Sea Con un subconjunto no vacío de fin(PHMRD(t)), donde fin(PHMRD(t)) es el conjunto que contiene todos los subconjuntos finitos de PHMRD(t). Cada subconjunto estará compuesto por los propios conceptos y por los conjuntos de conceptos que cumplen la siguiente propiedad:

$[\forall x \in X, \forall y \in (X\backslash\{x\})$ relation $(x,y) \Rightarrow X \subseteq Cons\}$, donde el predicado relation (x,y) significa que "x está relacionado con y".

De esta forma, dado que los elementos de PHMRD(t) son conceptos ontológicos, PHMRD(t) cumple la propiedad de consistencia.

**Proposición 3**: Sea PHMRD(t) un conjunto contable no vacío. Entonces, un PHMRD(t) cumple la propiedad de $\perp$-cierre.

*Prueba*: Definamos $\perp$ como sigue:

$\perp$: Dado $X \in$ Cons, $a \in A$ , $X \perp a \Leftrightarrow [\exists x \in X$ tal que relation (x, a) ].

Definiendo $\perp$ de esta forma y procediendo de forma similar como para la proposición 2, entonces se cumple la propiedad de $\perp$-cierre.

Una vez que se ha encontrado la estructura A=(A, Con, $\perp$) en el PHMRD(t) definido por una ontología definida atendiendo a nuestro modelo ontológico, ya podemos probar que este tipo de ontologías son sistemas de información.

**Proposición 4**: Una ontología es un sistema de información.

*Prueba*: Para ello, tenemos que probar que se cumplan los axiomas necesarios para que una ontología pueda ser considerada un sistema de información.

1. $X \subseteq Y \in Con \Rightarrow X \in Con$. Cada subconjunto de un conjunto consistente es también consistente por la propia definición de conjunto consistente.

2. $a \in A \Rightarrow \{a\} \in Con$. Este axioma queda garantizado por la forma en la que se definen los conjuntos consistentes. El elemento a es un concepto ontológico por lo que el conjunto formado por este concepto pertenece a A.

3. $X \perp a \Rightarrow X \cup \{a\} \in Con$. La relación de vinculación ocurre cuando el elemento 'a' está relacionado con cada elemento de 'X'. Por lo tanto, el nuevo conjunto que se crea como resultado de esta unión también es un conjunto consistente.

4. $X \in Con \ \& \ a \in X \Rightarrow X \perp a$. Este axioma se prueba trivialmente debido a la forma en la que se han definido los conjuntos consistentes.

5. $(X,Y \in Con \ \& \ \forall b \in Y.\ X \perp b \ \& \ Y \perp c) \Rightarrow X \perp c$. El cumplimiento de este axioma es evidente debido a la manera en la que se han definido los conjuntos consistentes.

Por lo tanto, las ontologías que resultan de nuestro modelo ontológico pueden considerarse sistemas de información. Esta propiedad facilitará la explotación de las mismas.

## 2.5 Un Lenguaje Sencillo para Especificar Ontologías

En esta tesis se ha desarrollado un lenguaje sencillo para construir ontologías conforme al modelo ontológico presentado en secciones anteriores. En esta sección se introduce dicho lenguaje. La especificación de los elementos ontológicos incluidos en el modelo, esto es, conceptos, relaciones y atributos debe ser facilitada por el lenguaje. A continuación pasamos a describir la especificación de este pseudo-lenguaje. El fichero puede ser visto como un fichero de texto que contiene la especificación de cada concepto incluido en la ontología. Se define cada concepto por medio de su nombre, sus atributos y sus conceptos padre, tanto taxonómicos como mereológicos. Las siguientes líneas explican con más detalle la información a especificar para cada concepto.

- Concepto
    - Nombre del concepto
    - Nombres alternativos del concepto. Formato:
        - Cantidad de nombres alternativos.
        - Lista de nombres alternativos.
    - Atributos específicos.
    - Cantidad de atributos específicos.
    - Lista de nombres de cada atributo específico. Para cada atributo:
        - Número de valores

- Lista de valores

- Atributos heredados.
  - Número de atributos heredados.
  - Lista de nombres de cada atributo heredado. Para cada atributo:
  - Número de valores
  - Lista de valores
- Padres taxonómicos
  - Número de padres taxonómicos
  - Lista de nombres de los padres taxonómicos
- Padres mereológicos
  - Número de padres mereológicos
  - Lista de nombres de los padres mereológicos.
- Lista de especializaciones
  - Número de especializaciones

- Lista de especializaciones. Una especialización es un par (nombre de atributo, nombre de padre taxonómico) de forma que el concepto actual es una subclase del padre taxonómico a través del atributo de la especialización. Cuando un concepto especializa al concepto padre por medio de más de un atributo, entonces se deberá crear una especialización por cada uno de los atributos.

## 2.5.1 Ejemplo

La Figura 2.2 corresponde a la representación gráfica de una ontología. La especificación de esta ontología parcial en el lenguaje anterior se muestra a continuación.

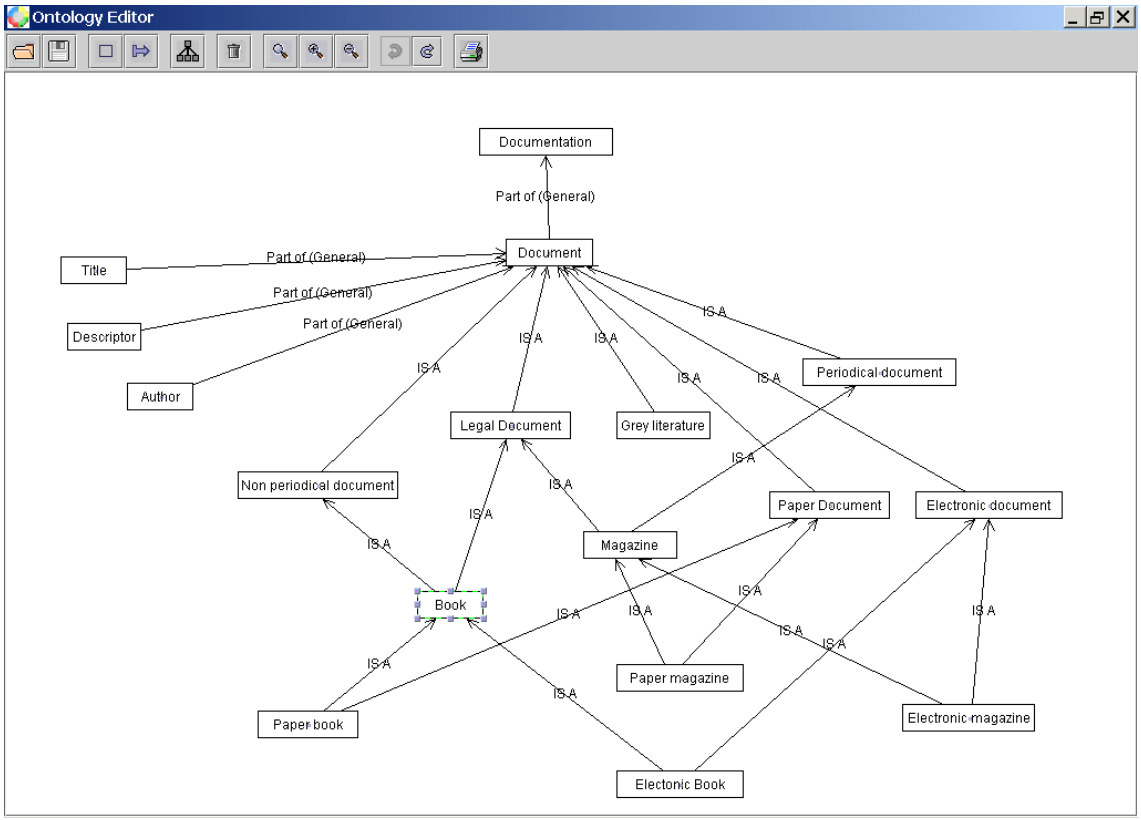


**Figura 2.2 Un ejemplo de ontología parcial**

```
DOCUMENTATION
0
0
0
0
0
0
DOCUMENT
0
5
authors
0
topic
0
legal ID
0
medium
0
periodicity
0
0
0
1
DOCUMENTATION
0
TITLE
0
2
original
0
translated
0
0
0
1
DOCUMENT
0
DESCRIPTORS
0
1
keywords
0
0
0
1
DOCUMENT
0
AUTHOR
0
2
name
0
surname
0
0
0
1
DOCUMENT
0
LEGAL DOCUMENT
0
1
ID
2
ISSN
ISBN
5
authors
0
topic
0
legal ID
0
medium
0
periodicity
0
1
DOCUMENT
0
1
legal ID,DOCUMENT
GREY LITERATURE
0
0
5
authors
0
topic
0
legal ID
0
medium
0
periodicity
0
1
DOCUMENT
0
1
legal ID,DOCUMENT
PAPER DOCUMENT
0
0
5
authors
0
topic
0
legal ID
0
medium
0
periodicity
0
1
DOCUMENT
0
1
medium,DOCUMENT
ELECTRONIC DOCUMENT
0
0
5
authors
0
topic
0
legal ID
0
medium
0
periodicity
0
1
DOCUMENT
0
1
medium,DOCUMENT
NON PERIODICAL
DOCUMENT
0
0
5
authors
0
topic
0
legal ID
0
medium
0
periodicity
0
1
DOCUMENT
```

```
0                       legal ID,LEGAL          ISBN
1                       DOCUMENT                O
periodicity,DOCUMEN     periodicity,PERIODI     medium
T                       CAL DOCUMENT            0
PERIODICAL DOCUMENT     MAGAZINE                2
0                       0                       BOOK
1                       1                       ELECTRONIC DOCUMENT
type                    ISSN                    0
0                       0                       2
5                       3                       ISBN,BOOK
authors                 legal ID                medium,ELECTRONIC
0                       0                       DOCUMENT
topic                   ID                      PAPER MAGAZINE
0                       0                       0
legal ID                periodicity             0
0                       0                       2
medium                  2                       ISSN
0                       LEGAL DOCUMENT          O
periodicity             PERIODICAL DOCUMENT     medium
0                       0                       0
1                       2                       2
DOCUMENT                legal ID,LEGAL          MAGAZINE
0                       DOCUMENT                PAPER DOCUMENT
1                       periodicity,PERIODI     0
periodicity,DOCUMEN     CAL DOCUMENT            2
T                       PAPER BOOK              ISSN,MAGAZINE
BOOK                    0                       medium,PAPER
0                       0                       DOCUMENT
1                       2                       ELECTRONIC MAGAZINW
ISBN                    ISBN                    0,0,0,0
0                       O                       0
3                       medium                  0
legal ID                0                       2
0                       2                       ISSN
ID                      BOOK                    O
0                       PAPER DOCUMENT          medium
periodicity             0                       0
0                       2                       2
2                       ISBN,BOOK               MAGAZINE
LEGAL DOCUMENT          medium,PAPER            ELECTRONIC DOCUMENT
NON PERIODICAL          DOCUMENT                0
DOCUMENT                ELECTRONIC BOOK         2
0                       0                       ISSN,MAGAZINE
2                       0                       Medium,ELECTRONIC
                        2                       DOCUMENT
```

## 2.5.2 El Lenguaje de Definición XML

En esta sección se presenta el DTD del lenguaje XML. Dicha definición de DTD incluye todos los aspectos cubiertos por el modelo ontológico y por el lenguaje anterior, esto es, conceptos, relaciones y atributos.

```
<!ELEMENT ontology (concept+, relation*, axioms?)>
<!ATTLIST ontology
    domain CDATA #REQUIRED
    author  CDATA #IMPLIED
    comment CDATA #IMPLIED
>

<!ELEMENT concept (alternative-names*, specific-attributes*, inherit-
attributes*)>
<!ATTLIST concept name CDATA #REQUIRED>
<!ELEMENT alternative-names (name+)>
<!ELEMENT specific-attributes (attribute+)>
<!ELEMENT inherit-attributes (attribute+)>
<!ELEMENT atrribute (value*)>
<!ATTLIST attribute
          name    CDATA #REQUIRED
          comment CDATA #IMPLIED
          type    CDATA #REQUIRED>

<!ELEMENT value ( intervalo | #PCDATA)>
<!ELEMENT intervalo (from, to) >
<!ELEMENT from (#PCDATA) >
<!ELEMENT to (#PCDATA) >
<!ELEMENT        relation        (relation_type,        concept_name,
specialization_list?)>
```

```
<!ELEMENT relation_type (property*)>
<!ATTLIST relation_type name CDATA #REQUIRED>
<!ELEMENT specialization_list (attribute+)>
<!ELEMENT axioms axioma*>
<!ELEMENT axioma (#PCDATA)>
<!ATTLIST axioma  type  CDATA  #REQUIRED>
<!ELEMENT concept_name (#PCDATA)>
<!ELEMENT property (#PCDATA)>
<!ELEMENT name (#PCDATA)>
```

## 2.6 RESUMEN

En este capítulo se ha propuesto y formalizado un modelo ontológico que tiene el objetivo de especificar ontologías del dominio. Este modelo ontológico contiene los siguientes elementos de conocimiento: (1) conceptos, que representan entidades del dominio; (2) atributos, que son propiedades de los conceptos y que pueden ser específicos o heredados; (3) relaciones, que establecen jerarquías en el dominio puesto que los dos tipos de relaciones incluidas en el modelo, taxonomía y mereología son relaciones jerárquicas; y (4) axiomas estructurales, que son axiomas derivados de la propia estructura de la ontología. También se ha formalizado un conjunto de supuestos, axiomas y propiedades relativas a la estructura de las ontologías. Además, se ha definido un conjunto de funciones ontológicas para disponer de mecanismos para trabajar con dichas ontologías. Finalmente, se ha demostrado que estas ontologías se pueden considerar dominios de Scott o sistemas de información. Este hecho facilita la explotación de las ontologías porque, por ejemplo, se podrían definir espacios topológicos en las ontologías, lo cual aumenta el conjunto de operaciones que pueden ser realizadas con las mismas. Todo eso nos sirve para especificar ontologías desde un punto de vista teórico.

En el capítulo anterior se han descrito los lenguajes ontológicos más utilizados en la actualidad. Más concretamente se analizaron seis lenguajes para especificar ontologías: XOL (XML-based Ontology Exchange Language), SHOE (Simple HTML Ontology Extension),

OML (Ontology Markup Language), RDF(S) (Resource Description Framework), OIL (Ontology Interchange Language), y DAML+OIL. Se describió el modelo de conocimiento representable en cada lenguaje y se mostraron partes de ontologías en dichos lenguajes. Sin embargo se encontró que ninguno de esos lenguajes es óptimo para especificar las ontologías construidas conforme al modelo propuesto en este capítulo. Por lo tanto, se ha propuesto un lenguaje sencillo para especificar nuestras ontologías. Dicho lenguaje fue diseñado con el único objetivo de facilitar la construcción de nuestras ontologías, por lo que no es capaz de especificar otros tipos de ontologías. Por lo tanto, para facilitar la interoperabilidad de nuestras ontologías se ha desarrollado un DTD, el cual también se ha presentado en este capítulo.

# CAPÍTULO 3

# UN ENTORNO COOPERATIVO PARA LA INTEGRACIÓN DE ONTOLOGÍAS

Este capítulo proporciona la formalización del entorno de integración de conocimiento. La integración de conocimiento es vista en este trabajo como integración de ontologías, puesto que se representa el conocimiento en este trabajo por medio de ontologías. Nuestro entorno proporciona una infraestructura para permitir la integración de ontologías es proporcionada. Los procesos de integración de ontologías se basan en la comparación de las ontologías a integrar. Es más, se debe superar algunos problemas debidos a la naturaleza cooperativa de estos procesos de integración como son la sinonimia, la inconsistencia y la equivalencia. El entorno propone soluciones a estos problemas tomando en cuenta tanto propiedades estructurales de las ontologías como criterios relacionados con atributos.

Este capítulo se estructura como sigue. En primer lugar, se describirán los problemas derivados del trabajo cooperativo. Después se formalizará el entorno de integración. Este proceso será dividido en diferentes partes de acuerdo al tipo de entidades ontológicas involucradas en cada de grupo de definiciones y su papel en el mecanismo de integración. Finalmente se presentará un enfoque algorítmico del proceso de integración y se harán algunos comentarios finales.

## 3.1 COOPERACION Y SUS PROBLEMAS

Entendemos por cooperación el fenómeno por el cual distintas personas contribuyen a la creación de una ontología común. La naturaleza de sus contribuciones es ontológica, esto es, cada uno crea una ontología privada. De la unificación de las piezas privadas de conocimiento se obtiene una ontología común y construida de forma cooperativa. Sin embargo, tales piezas individuales de conocimiento representan puntos de vista individuales, lo cual puede provocar que existan problemas para unificar estos puntos de vista particulares. Estos problemas son conocidos como problemas derivados de la cooperación, y entre ellos podemos destacar los siguientes:

a)**Información redundante**. Dos ontologías diferentes pueden intentar describir la misma parte del dominio de conocimiento de la misma forma. Debido a ello, sería deseable que el entorno de integración controlara tal situación, de forma que se pudieran evitar las redundancias. El ejemplo más simple de redundancia es la existencia de dos ontologías iguales. Otras situaciones de redundancia se derivan del descubrimiento de la existencia de conceptos sinónimos (ver siguiente epígrafe).

b)**Uso de términos sinónimos para un concepto**. Diferentes ontologías pueden emplear diferente terminología para un mismo concepto. En otras palabras, puede haber una correspondencia entre los términos empleados para un concepto dado (Shaw and Gaines, 1989). Durante el proceso de construcción de ontologías, la información relativa al uso de términos sinónimos para un concepto debería ser almacenada y gestionada, puesto que no se debe imponer una terminología específica durante el proceso de adquisición de conocimiento. Sin embargo, una ontología deriva hacia un conocimiento consensuado, esto es, una terminología fija. Los sinónimos son posibles, pero, idealmente, la terminología debería ser común.

c)**Conocimiento inconsistente**. Este aspecto posee una doble naturaleza. Una ontología puede ser interna o externamente inconsistente. Podemos decir que una ontología es internamente inconsistente cuando una parte de ella es inconsistente con otra parte de la misma ontología. Por ejemplo, una ontología es internamente inconsistente si hay al menos una propiedad relativa a relaciones entre conceptos no satisfecha. La Figura 3.1 es un ejemplo de inconsistencia interna. En esta figura, los conceptos A y B son a la vez padres e hijos mereológicos del otro, lo cual no es posible según los axiomas mereológicos. Por otro lado, una ontología puede ser externamente inconsistente con respecto a otra ontología, esto es, ambas descripciones del mismo dominio pueden ser incompatibles. Esta situación queda representada por la Figura 3.2, donde se muestran dos partes de dos ontologías diferentes. En la primera ontología, A es hijo mereológico de B mientras que en la segunda, B es hijo mereológico de A, y esta situación no es consistente con respecto a los axiomas mereológicos.

Asumiremos que el sistema de integración recibe ontologías libres de inconsistencias internas. Podemos asumir que, debido a los axiomas estructurales impuestos a nuestro modelo

ontológico, las ontologías construidas según este modelo son internamente consistentes. Esto queda garantizado por el uso de un formato específico, descrito en el capítulo anterior para el formato de fichero ontológico. De ahora en adelante, cuando nos refiramos a inconsistencia nos estaremos refiriendo a inconsistencia externa.

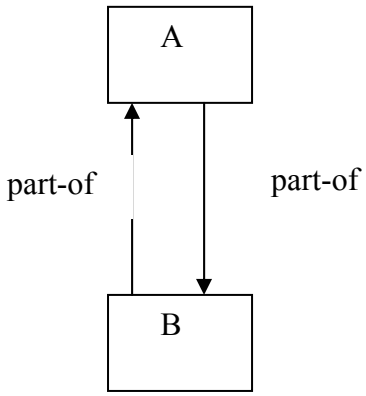


**Figura 3.1 Inconsistencia Interna**

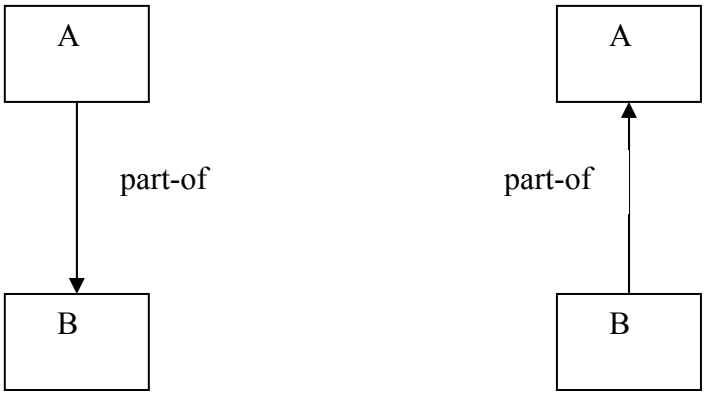


**Figura 3.2 Inconsistencia externa**

## 3.2 EL ENTORNO PARA INTEGRAR CONOCIMIENTO

Una vez presentado formalmente el modelo ontológico debemos formalizar el entorno para integrar conocimiento. La integración de conocimiento es vista aquí como integración de ontologías, puesto que las ontologías son la tecnología usada para representar conocimiento. Por lo tanto, el entorno debe proporcionar una infraestructura que permita la integración de ontologías. Así, el punto de partida del entorno de integración es un conjunto de ontologías que pertenecen a diferentes usuarios. Un usuario puede tener más de una ontología sobre un mismo tema, y en este caso, la ontología más antigua será considerada obsoleta por lo que se descartará del proceso de integración. Cuando se lanza un proceso de integración, será necesario realizar una serie de pasos para cumplimentar la tarea.

Una operación fundamental del proceso de integración es la comparación del conocimiento contenido en ontologías diferentes. Por lo tanto, se necesitan diferentes funciones para comprobar la equivalencia entre conceptos y/o ontologías, así como funciones para encontrar inconsistencias entre ontologías. En este trabajo se consideran las equivalencias y las inconsistencias desde dos puntos de vista, a saber, (1) atributos, y (2) estructura organizativa de la ontología. Otra función necesaria permitirá decidir cuando dos conceptos son sinónimos, lo cual permitirá que los usuarios puedan utilizar su propia terminología.

El marco usado para integrar un conjunto de ontologías ya existentes será construido a partir de una red local y cooperativa (por ejemplo, un conjunto de usuarios conectados a Internet o a alguna intranet). Esta red será considerada un mecanismo de transmisión de ontologías asociadas a usuarios-nodo. Se deben realizar algunas consideraciones respecto a los atributos y su representación en este entorno de integración. Un atributo es identificado únicamente por su nombre porque no se tiene en cuenta ni su estructura interna ni sus valores asociados. Los atributos son únicamente sus nombres, por lo que el atributo “leg” de una persona sería equivalente al atributo “leg” de una eliminatoria de Copa de la UEFA si ambos atributos son comparados de forma aislada. Por lo tanto, las funciones ontológicas presentadas en las secciones siguientes sólo usan nombres de atributos para hacer comparaciones entre ellos.

Por otro lado, si existen inconsistencias entre partes del conocimiento perteneciente a la ontología correspondiente a un usuario-nodo particular en el instante t, denotado por $O_i(t)$, y la ontología resultante de la integración de todas las ontologías disponibles en el instante t, denotado por $O_{int}(t)$, (asumiendo una definición de inconsistencia ontológica basada en la estructura de la ontología (p.ej., taxonomías/mereologías) y en la definición conceptual (a través del conjunto de atributos), se asumirá como válido el conocimiento contenido en $O_i(t)$. Esta política ha sido adoptada tomando en cuenta que los usuarios expertos pueden realizar consultas a $O_{int}(t)$ mientras construyen sus ontologías privadas.

El conjunto de funciones definidas en el Capítulo 2 es necesario puesto que trabajar con varias ontologías requiere comparar el conocimiento almacenado en cada una de ellas con la finalidad de integrarlas. Ya podemos proceder a presentar el proceso de integración. Todas las ontologías a integrar deben cubrir el mismo dominio. Esta es la primera consideración a tener en cuenta cuando deseemos integrar varias ontologías usando este entorno. El siguiente paso es seleccionar qué ontologías van a ser finalmente integradas, puesto que puede haber ontologías equivalentes e inconsistentes en el grupo de ontologías a integrar.

Para realizar esta operación es importante saber si dos ontologías cualesquiera reúnen los requisitos lógicos para la integración. No pueden integrarse dos ontologías si son inconsistentes o equivalentes. Es obvio que no pueden integrarse ontologías inconsistentes porque ello incorporaría conocimiento contradictorio en la ontología derivada de la integración. Por otro lado, las ontologías equivalentes no son adecuadas para la integración porque incorporarían redundancia en la ontología derivada de la integración, lo cual haría improductivo alguno de los esfuerzos anteriores.

Después de calcular el conjunto de ontologías compatibles para la integración (para cada ontología fuente), se debe escoger el mejor de ellos. Esta decisión depende del objetivo a alcanzar. En nuestro caso, el sistema intenta aumentar al máximo la calidad y cantidad de información suministrada a los usuarios. Por tanto, nuestro criterio será que el conjunto con mayor número de ontologías es el mejor. Una vez tomada esta decisión puede comenzar el proceso de integración. Se generará una nueva ontología, pero este conocimiento no se crea

desde cero sino que se reusan las ontologías fuente. La nueva operación a realizar es colocar el conjunto previamente seleccionado de ontologías como parte de la nueva ontología. Esto se hace en la práctica insertando cada ontología fuente en la ontología derivada de la integración como hijo mereológico del concepto raíz, cuyo nombre es el dominio que está siendo integrado. La formalización de este proceso aparece en la Sección 3.2.2.

Hemos alcanzado un punto en el que el conjunto inicial de ontologías ha sido filtrado (eliminando aquéllas inconsistentes y equivalentes), y hemos colocado a las ontologías seleccionadas en una ontología común.

La próxima fase es transformar la terminología de cada ontología fuente de forma que se comparta la terminología. Esta propiedad sería deseable para los pasos siguientes del proceso de integración. La ontología obtenida de este proceso de unificación terminológica se llama ontología integrada e instanciada. Para hacer esta unificación de vocabulario, se debe tomar una de las ontologías que toman parte en el proceso de integración como referencia. Concretamente, la terminología de la ontología de referencia debe tomarse como referencia para el proceso. Ahora debemos decidir qué ontología debe ser escogida como referencia para completar esta unificación terminológica. En este trabajo, la terminología de referencia depende de qué usuario ha solicitado la integración. Si el usuario tiene una ontología privada en el dominio, esta ontología será considerada la referencia terminológica. De otra forma, la decisión depende del criterio que escojamos. En nuestro caso, el criterio es la ontología con el mayor número de conceptos, para minimizar potencialmente el número de cambios a realizar. El proceso de unificación puede describirse brevemente como sigue. Para cada concepto C de cada ontología incluida en la ontología derivada de la integración, si existe un concepto C' en la ontología referencia tal que ambos conceptos son sinónimos, entonces se cambia el nombre de C al nombre de C'.

Llegamos a la fase final del proceso de integración. Una vez unificada la terminología, el sistema puede generar la ontología final y transformada a partir de la ontología integrada e instanciada. Para ello, nuestro enfoque tomará una de las sub-ontologías que son hijas mereológicas de la raíz de de ontología integrada e instanciada como el esqueleto para realizar

la transformación. Además, se agregarán nuevos conceptos, atributos y relaciones a ésta para finalmente obtener la ontología final que será mostrada al usuario. En este caso, también es importante conocer si la ontología referencia pertenece al usuario que solicitó la integración del tema, así que el criterio usado aquí es seleccionar la ontología con el mayor número conceptos como la primera referencia, o la referencia en caso de pertenecer al usuario que solicitó la integración. De esta forma, todas las ontologías se ordenan según el número de conceptos que tienen, y serán fusionadas con la ontología transformada en el orden establecido La intención es minimizar el número de modificaciones a realizar en la ontología transformada, puesto que conforme más ontologías hayan sido incluidas, más conocimiento contendrá la ontología transformada. Al adoptar esta política es más probable que el conocimiento de las ontologías que aún no han sido tratadas ya haya sido incluido en la ontología transformada.

La siguiente fase es la transformación de la ontología esqueleto insertando los conceptos, relaciones y atributos que pertenecen a las otras ontologías que toman la parte del proceso. Este mecanismo procesa una ontología por ciclo, y esta ontología se compara con la ontología transformada hasta ese momento para descubrir qué nuevo conocimiento puede agregarse a la ontología transformada. Cada ontología se procesa concepto a concepto, verificando la existencia de algún concepto equivalente o sinónimo en la ontología transformada. En este caso, deben unificarse los atributos y relaciones de ambos conceptos. Si no hay ningún concepto equivalente o sinónimo en la ontología transformada, el concepto debe agregarse a la ontología transformada en su lugar correspondiente, que vendrá dictaminado por sus relaciones.

Antes de formalizar el mecanismo de integración descrito en esta sección, mostraremos las ontologías que son utilizadas en este capítulo para ilustrar las diferentes definiciones. Concretamente, se han usado cuatro ontologías para este propósito, las cuales son mostradas en las Figuras 3.3-3.6.

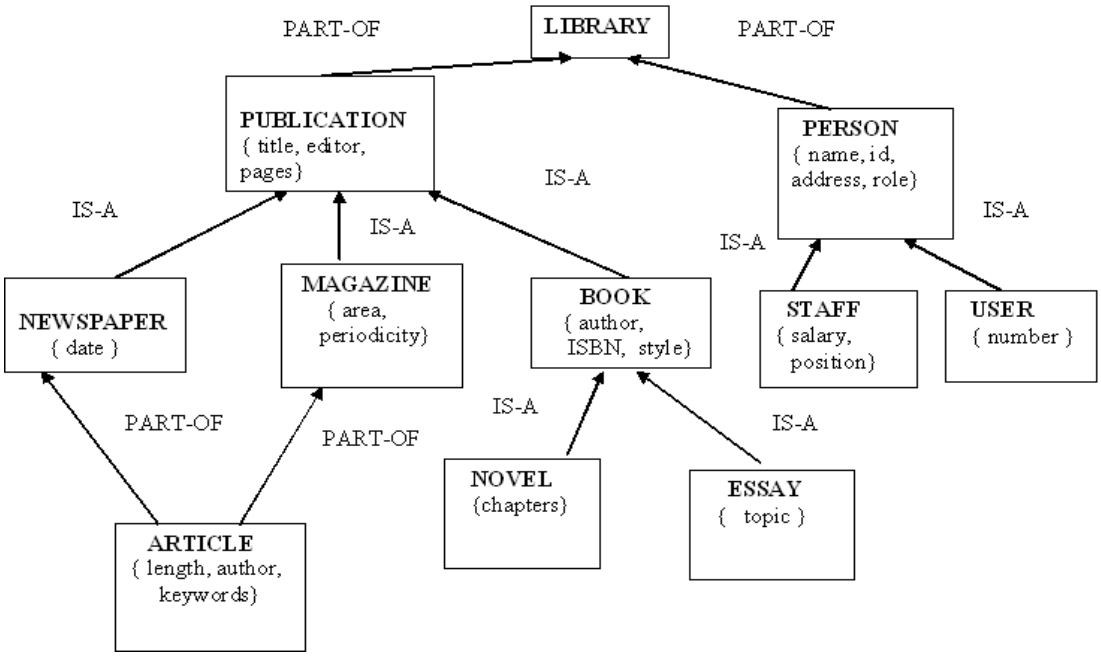


**Figura 3.3. Ontología del experto A**

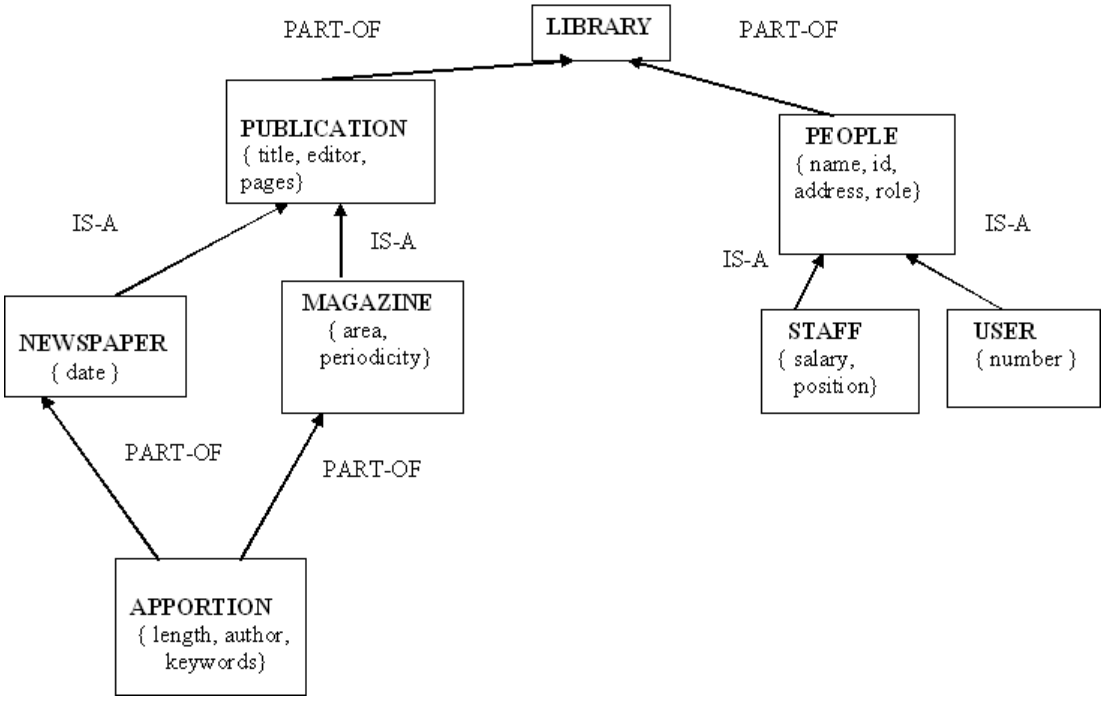


**Figura 3.4. Ontología del experto B**

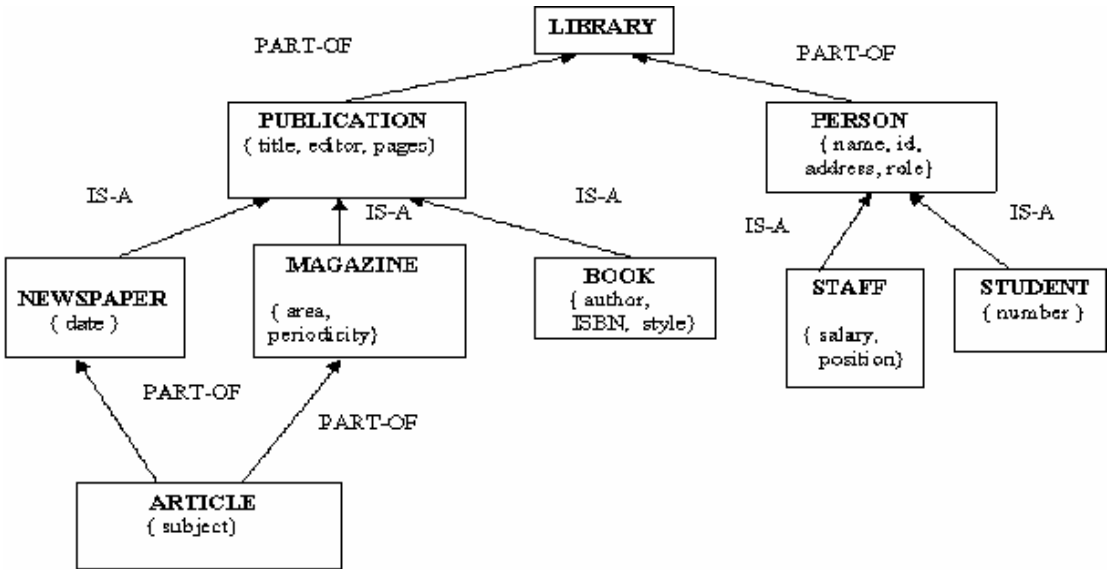


**Figura 3.5. Ontología del experto C**

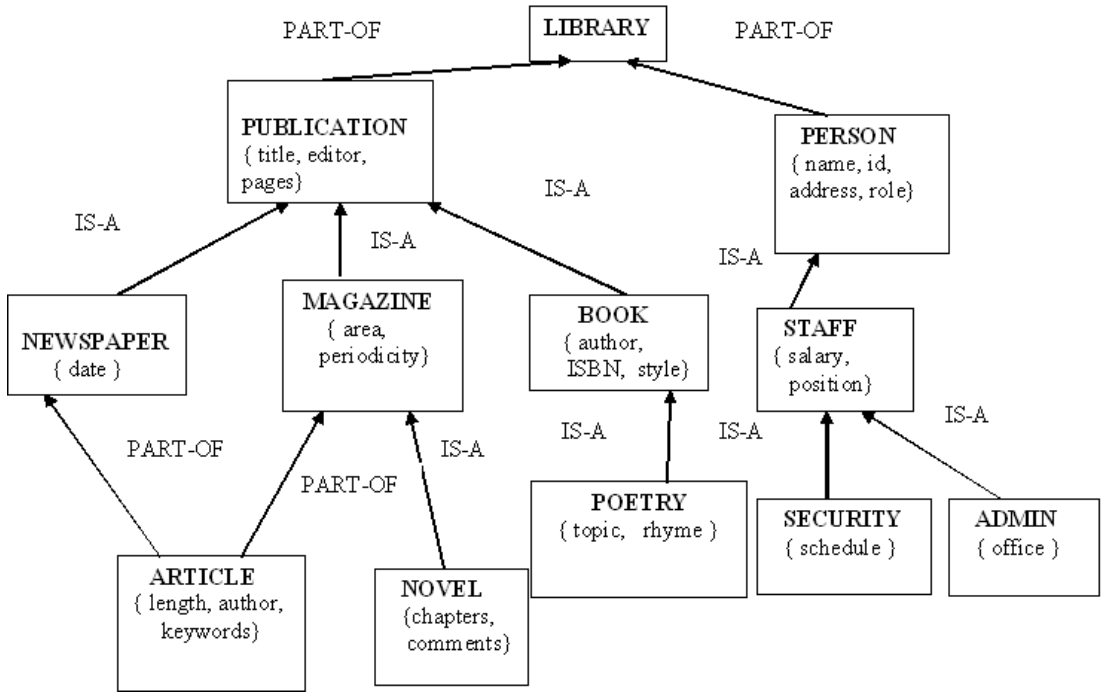


**Figura 3.6. Ontología del experto D**

## 3.2.1 Principios de integración

Como se mencionó anteriormente, un objetivo de este trabajo es diseñar un entorno para la construcción de ontologías a partir de aquéllas suministradas por un conjunto de constructores de ontologías (p.ej., usuarios-nodo). Consecuentemente, el sistema debe resolver los posibles conflictos de consistencia que aparezcan entre las ontologías a integrar en el instante t. Cada vez que un creador de conocimiento inserta o modifica conocimiento en su ontología privada, tal conocimiento debe ser incorporado a $O_{int}(t)$. Debe ser subrayado que varios usuarios-nodo podrían enviar conocimiento para construir la ontología integrada al mismo tiempo, por lo que es necesario distinguir entre piezas de conocimiento suministradas por diferentes usuarios. A continuación se detalla el modus operandi del entorno para integrar un conjunto de ontologías propuesto en este trabajo.

Sea $O_i(t)$ una ontología suministrada por el i-ésimo usuario-nodo hasta el instante t; i=1,...,n; n = número de usuarios activos en el insatnte t. Entonces, el primer principio relativo al modelo de integración aquí propuesto es el siguiente.

### Principio 1: integración guiada por el usuario

Todo el conocimiento contenido en $O_{int}(t_j)$ debe ser consistente con $O_i(t_i)$ para todo i en $\{1,2,...,n\}$ y $t_j > t_i$. Esto asegura que $O_{int}(t)$ siempre contendrá la última versión de las ontologías de los usuarios.

Aunque se pueda pensar que ayudar a los usuarios a construir sus propias ontologías es una poderosa utilidad de una ontología integrada, su principal función debe ser suministrar conocimiento sobre un dominio a las peticiones de los usuarios, que no tienen por qué ser constructores de conocimiento. Para mostrar la información al usuario de forma compacta, no parece una buena solución mostrar al usuario las ontologías de cada constructor, sino que hemos adoptado la decisión de mostrar únicamente el resultado de la integración de todas las ontologías. Sin embargo, esta integración puede ser muy compleja puesto que, en general, los

mismos conceptos (incluso con diferentes atributos) pueden darse en las distintas ontologías contenidas en $O_{int}(t)$.

Para solucionar este problema seguiremos un algoritmo basado en el principio "cada usuario desea obtener la cantidad máxima de información acerca de un dominio". En particular, para construir la ontología que será mostrada al usuario, comenzaremos por la ontología privada que tenga el mayor número de conceptos. Formalmente, este principio puede expresarse de la siguiente forma:

### Principio 2: máximo contenido informativo

Sea $O_g(t)$ una ontología global obtenida como resultado de integrar algunas ontologías hasta el instante t y sea $O_i(t)$ la i-ésima ontología incluida en $O_g(t)$ hasta el instante t, i=1,2...,n; n=número de usuarios-nodo cuyas respectivas ontologías son parte de $O_g(t)$ en el instante t. Entonces, la información suministrada al usuario contendrá, al menos, tantos conceptos como tiene $O_m(t)$, donde $O_m(t)$ es una ontología de $O_g(t)$ tal que:

$$\text{Cardinal}(PMHRD_m(t)) = \max_i \{ \text{Cardinal}(PMHRD_i(t)) \},\ i = 1..n$$

donde $PMHRD_x(t)$ representa el dominio jerárquico múltiple parcial restringido hasta el instante t correspondiente a la ontología $O_x(t)$, $x \in \{m, i\}$.

## 3.2.2 Un Entorno para Integrar Ontologías

Considerando las funciones ontológicas ya introducidas, y suponiendo un criterio de igualdad basado en nombres, la siguiente definición compara elementos diferentes de dos ontologías. Decimos que dos conceptos de dos ontologías diferentes son equivalentes desde el punto de vista de los atributos que corresponden a cada concepto si tienen el mismo conjunto de atributos. Formalmente:

**Definición 1: equivalencia conceptual basada en atributos**

Sean $PMHRD_i(t)$ y $PMHRD_j(t)$ dos PMHRD(t), y sean c(t) y c'(t) dos conceptos de forma que $c(t) \in PMHRD_i(t)$ y $c'(t) \in PMHRD_j(t)$. Se dice que c(t) y c'(t) son *equivalentes desde el punto de vista de los atributos*, denotado por *A_equivalency(c(t), c'(t))*, sí y sólo si ATT(c(t)) = ATT(c'(t)).

Ilustremos esta definición con el ejemplo siguiente. Tomemos los conceptos "staff" y “article” de las ontologías de los expertos A (Figura 3.3) y C (Figura 3.5).

**$ATT(staff_A)$** = {salary, position, name, id, address, role}

**$ATT(staff_C)$** = {salary, position, name, id, address, role}

Los conceptos "$staff_A$" y "$staff_C$" tienen los mismos atributos de forma que los conceptos cumplen la A_equivalency. Otro ejemplo pueden ser los conceptos “article” de las mismas ontologías.

**$ATT(article_A)$** = {length, author, keywords}

**$ATT(article_C)$** = {subject }

Los conceptos "$article_A$" y "$article_C$" no tienen los mismos atributos, por lo que no son A_equivalentes.

Similarmente, podemos extender la definición anterior para tratar con conjuntos de conceptos. Dos conjuntos de conceptos son equivalentes desde el punto de vista de los atributos si ambos conjuntos son vacíos, o si para cada concepto de cada conjunto existe un concepto en el otro conjunto tal que ambos conceptos son A_equivalentes. Formalmente:

Definición 2: Equivalencia conceptual extendida basada en atributos

Sean $C_i$ y $C_j$ dos conjuntos de conceptos. $C_i$ y $C_j$ son dos conjuntos de conceptos equivalentes desde el punto de vista de los atributos, denotado por EA_equivalency ($C_i$, $C_j$), sí y sólo si se cumple:

[$C_i = C_j = \varnothing$] or [(for all c(t) in $C_i$, exists c'(t) in $C_j$ such that A_equivalency(c(t),(c'(t))) and (for all c(t) in $C_j$, exists c'(t) in $C_i$ such that A_equivalency(c(t), c'(t))]

Tomemos las mismas ontologías para presentar un ejemplo para esta definición. Supongamos que tenemos el siguiente conjunto de conceptos:

$Set_B$={people, staff, user}

$Set_C$= {person, staff, user}

**ATT($people_B$)** = {name, id, address, role}

**ATT($staff_B$)** = {salary, position, name, id, address, role}

**ATT($user_B$)** = {number, name, id, address, role}

**ATT(person$_C$)** = {name, id, address, role}

**ATT(staff$_C$)** = {salary, position, name, id, address, role}

**ATT(user$_C$)** = {number, name, id, address, role}

Si aplicamos la definición, podemos ver que existen las siguientes A_equivalencias:

A_Equivalency(people$_B$, person$_C$)

A_Equivalency(user$_B$, user$_C$)

A_Equivalency(staff$_B$, staff$_C$)

Consideremos ahora que dos conceptos tienen los mismos padres e hijos. Consideremos que también puede ocurrir que esos dos conceptos no tengan ningún atributo en común. Entonces podemos establecer lo siguiente. Dos conceptos son inconsistentes desde el punto de vista de los atributos si se cumplen las siguientes condiciones: los conceptos tienen el mismo término como nombre; no tienen atributos en común; los respectivos conjuntos de conceptos padres e hijos mereológicos/taxonómicos, si los tienen, son equivalentes desde el punto de vista de los atributos. Formalmente:

Definición 3: inconsistencia conceptual basada en atributos

Sean PMHRD$_i$(t) y PMHRD$_j$(t) dos PMHRD(t), y sean c(t) y c'(t) dos conceptos tales que c(t) $\in$ PMHRD$_i$(t) y c'(t) $\in$ PMHRD$_j$(t). Se dice que c(t) y c'(t) son *inconsistentes conceptualmente desde el punto de vista de los atributos*, denotado por *A_inconsistency (c(t), c'(t))*, sí y sólo si se cumple lo siguiente:

(NAME(c(t)) = NAME(c'(t))) and (ATT(c(t)) ∩ ATT(c'(t)) = ∅) and EA_equivalency(M-parents(c(t)), M-parents(c'(t))) and EA_equivalency(T-parents(c(t)), T-parents(c'(t))) and EA_equivalency(M-children(c(t)), M-children(c'(t))) and EA_equivalency(T-children(c(t)), T-children(c'(t))) donde NAME(x) es la etiqueta lingüística asignada al concepto x en su correspondiente PMHRD(t), $x \in \{c(t), c'(t)\}$.

Un ejemplo para esta definición lo encontramos con los conceptos "$article_A$" y "$article_C$", puesto que tienen el mismo nombre, y $ATT(article_A) \cap ATT(article_C) = \varnothing$. Por lo tanto, ambos conceptos son inconsistentes desde el punto de vista de los atributos.

**$ATT(article_A)$** = {length, author, keywords}

$ATT(article_C)$ = {subject}

Si hacemos uso de la definición 2, en caso de tener dos conjuntos de conceptos sin conceptos equivalentes desde el punto de vista de los atributos, podemos decir que ambos conjuntos de conceptos son disjuntos. Esto se puede formalizar de la forma siguiente:

Definición 4: conjuntos de conceptos disjuntos

Sean $C_i$ y $C_j$ dos conjuntos de conceptos. Se dice que $C_i$ y $C_j$ son dos *conjuntos de conceptos disjuntos*, denotado por *dis_concept_sets($C_i$, $C_j$)*, sí y sólo si:

($C_i \neq \varnothing$ or $C_j \neq \varnothing$) and [for all c(t) in $C_i$, not (exists c'(t) in $C_j$ such that A_equivalency(c(t), c'(t)))]

Sea $Set_A$= {publication, magazine, article} un conjunto de conceptos perteneciente a la ontología del experto A; Sea $Set_C$= {article, book, person} un conjunto de conceptos perteneciente a la ontología del experto C. Para ningún concepto perteneciente a $Set_A$ existe un concepto A_equivalente en $Set_C$ y viceversa, por lo que ambos conjuntos son disjuntos.

Dados dos conceptos pertenecientes a organizaciones taxonómicas o mereológicas, podemos hacer uso de propiedades estructurales para establecer la equivalencia y la inconsistencia conceptual entre estos conceptos. Dos conceptos son equivalentes desde el punto de vista de la estructura si se dan las dos condiciones siguientes. Primero, los respectivos conjuntos de conceptos padres mereológicos/taxonómicos son equivalentes desde el punto de vista de los atributos. Segundo, los respectivos conjuntos de conceptos hijos mereológicos/taxonómicos son equivalentes desde el punto de vista de los atributos. Formalmente:

Definición 5: equivalencia conceptual basada en la estructura de la ontología

Sean $PMHRD_i(t)$ y $PMHRD_j(t)$ dos PMHRD(t); sean c(t) y c'(t) dos conceptos tales que $c(t) \in PMHRD_i(t)$ y $c'(t) \in PMHRD_j(t)$. Se dice que c(t) y c'(t) son *conceptualmente equivalentes desde el punto de vista de la estructura*, denotado por *OS_equivalency(c(t), c'(t))*, sí y sólo si:

EA_equivalency(M-parents(c(t)), M-parents(c'(t))) and EA_equivalency(T-parents(c(t)), T-parents(c'(t))) and EA_equivalency(M-children(c(t)), M-children(c'(t))) and EA_equivalency(T-children(c(t)), T-children(c'(t))) .

"People" y "person" son dos conceptos OS_equivalentes, pertenecientes a $O_B(t)$ y $O_A(t)$ respectivamente . Veamos cómo se cumplen todas las condiciones.

M-parents (people) = {library};

M-parents (person) = {library}. Por lo tanto,

EA_equivalency (M-parents(people), M-parents (people)) = true;

T-parents(people)=T-parents(person) = ∅. Por lo tanto,

EA_equivalency (T-parents(people), T-parents (people)) = true.

M-children(people)=M-children(person) = ∅. Por lo tanto,

EA_equivalency (M-children(people), M-children (people)) = true.

T-children(people) = {$staff_B$, $user_B$};

T-children (person) = {$staff_C$, $user_C$}

**EA_equivalency (T-children(people), T-children (person))** = true puesto que A_equivalent($staff_B$, $staff_C$) y A_equivalent($user_B$, $user_C$).

Procediendo de manera similar, dos conceptos son inconsistentes desde el punto de vista de la estructura si se dan las siguientes condiciones:

Los conceptos son equivalentes desde el punto de vista de los atributos;

Sus respectivos conjuntos de padres taxonómicos/mereológicos son disjuntos.

Sus respectivos conjuntos de hijos taxonómicos/mereológicos son disjuntos.

Podemos expresar formalmente lo anterior:

Definición 6: Inconsistencia conceptual basada en la estructura

Sean $PMHRD_i(t)$ y $PMHRD_j(t)$ dos PMHRD(t), sean c(t) y c'(t) dos conceptos tales que $c(t) \in PMHRD_i(t)$ y $c'(t) \in PMHRD_j(t)$. Se dice que c(t) y c'(t) son *conceptualmente inconsistentes desde el punto de vista de la estructura*, denotado por *OS_inconsistency (c(t), c'(t))*, sí y sólo si:

A-equivalency(c(t), c'(t)) and dis_concept_set(M-parents(c(t)), M-parents(c'(t))) and dis_concept_set(T-parents(c(t)), T-parents(c'(t))) and dis_concept_set(M-children(c(t)), M-children(c'(t))) and dis_concept_set(T-children(c(t)), T-children(c'(t))).

Usemos las ontologías $O_A(t)$ (Figura 3.3) y $O_D(t)$ (Figura 3.6) para ilustrar esta definición. Tomemos los conceptos "$novel_A$" y "$novel_D$".

Condición 1:

$ATT(novel_A)$ = { chapters, author, ISBN, style};

$ATT(novel_D)$ = { chapters, area, periodicity}. Por lo tanto,

A-equivalency($novel_A$, $novel_B$) = false.

Condición 2:

M-parents($novel_A$)= ∅;

M-parents($novel_D$)={ magazine }. Por lo tanto,

dis_concept_set(M-parents($novel_A$), M-parents($novel_D$)) = true.

Condición 3:

T-parents($novel_A$)={ book };

T-parents($novel_D$)= ∅. Por lo tanto,

dis_concept_set(T-parents($novel_A$), T-parents($novel_D$)) = true.

Condición 4:

M-children($novel_A$)= ∅;

M-children($novel_D$)= ∅. Por lo tanto,

dis_concept_set(M-children($novel_A$), M-children($novel_D$)) = true.

Condición 5:

T-children($novel_A$)= ∅;

T-children($novel_D$)= ∅. Por lo tanto,

dis_concept_set(T-children($novel_A$), T-children($novel_D$)) = true.

Por lo tanto, puesto que los conceptos no son A-equivalentes no pueden ser considerados inconsistentes desde el punto de vista de la estructura de la ontología.

Basándonos en las definiciones previas, podemos establecer que dos conceptos son sinónimos si son equivalentes desde el punto de vista de la estructura y de los atributos. Formalmente:

Definición 7: conceptos sinónimos

Sean $PMHRD_i(t)$ y $PMHRD_j(t)$ dos PMHRD(t), y sean c(t) y c'(t) dos conceptos tales que $c(t) \in PMHRD_i(t)$ y $c'(t) \in PMHRD_j(t)$. Se dice que c(t) y c'(t) son *conceptos sinónimos*, denotado por s*yn_concepts(c(t),c'(t))*, sí y sólo si:

A_equivalency(c(t),c'(t)) and OS_equivalency(c(t), c'(t))

Si miramos las Figuras 3.4 y 3.5, los conceptos "$people_B$" y"$person_C$" son sinónimos porque son A y OS equivalentes.

$ATT(people_B)$ = {name, id, address, role};

$ATT(person_C)$ = {name, id, address, role}. Por lo tanto,

A-equivalency ($people_B$, $person_C$) = true.

Por otra parte, como se ha demostrado en la definición 5, ambos son OS_equivalentes, ya que **OS_equivalency($people_B$, $person_C$) = true**.

Por otra parte, diremos que dos conceptos diferentes son incompatibles si existe alguna inconsistencia basada en los atributos o en la estructura entre ellos. Esta definición se puede formalizar de la siguiente forma:

Definición 8: conceptos incompatibles

Sean $PMHRD_i(t)$ y $PMHRD_j(t)$ dos PMHRD(t), y sean c(t) y c'(t) dos conceptos tales que $c(t) \in PMHRD_i(t)$ y $c'(t) \in PMHRD_j(t)$. Se dice que c(t) y c'(t) son *conceptos incompatibles*, denotado por *incompatible_concepts (c(t),c'(t))*, sí y solo si:

A_inconsistency(c(t),c'(t)) or OS_inconsistency(c(t), c'(t))

Un ejemplo para esta definición puede ser encontrado en los conceptos "$article_A$" y "$article_C$", puesto que son inconsistentes desde el punto de vista de los atributos.

$ATT(article_A)$ = {length, author, keywords};

$ATT(article_C)$ = {subject}. Therefore,

A_inconsistency($article_A$, $article_C$) = true.

Esto implica que incompatible_concepts($article_A$, $article_C$) = true.

A partir de las dos últimas definiciones no se puede deducir que la negación de sinonimia entre dos conceptos implique la afirmación de la incompatibilidad entre ellos, puesto que existen varios estados posibles entre dos predicados, aunque ambos predicados no pueden ser verdad al mismo tiempo. Ahora podemos establecer cuando dos ontologías son equivalentes o inconsistentes. Dos ontologías son equivalentes si para cada concepto de una de ellas existe en la otra ontología un concepto equivalente. Formalmente:

Definición 9: ontologías equivalentes

Sean $PMHRD_i(t)$ y $PMHRD_j(t)$ dos PMHRD(t) correspondientes a $O_i(t)$ y $O_j(t)$ respectivamente, consideradas en el instante t. Se dice que $O_i(t)$ y $O_j(t)$ son *dos ontologías equivalentes*, denotado por *equivalent_ontologies* $(O_i(t), O_j(t))$, sí y solo si:

[for all c(t) in $PMHRD_i(t)$, exists c'(t) in $PMHRD_j(t)$ such that syn_concepts(c(t), c'(t))]

**and** [for all c(t) in $PMHRD_j(t)$, exists c'(t) in $PMHRD_i(t)$ s. t syn_concepts(c(t), c'(t))]

Entre las cuatro ontologías usadas en este ejemplo no existen dos ontologías equivalentes. Sin embargo, si consideramos subontologías sí podemos encontrar equivalencias. Tomemos una subontología de $O_B(t)$ y otra de $O_C(t)$ . Para ello, tendremos en cuenta únicamente el concepto raíz (Library) y sus hijos mereológicos "people" y "person" respectivamente, y sus respectivos descendientes mereológicos y taxonómicos y así sucesivamente. En otras palabras, consideraremos las ramas "person"/"people" de cada ontología, pero no la rama "publication" (ver Figuras 3.7-3.8).

Estas subontologías serían equivalentes puesto que para cada concepto de una ontología se encontraría en la otra un concepto sinónimo.

synonym_concepts($library_B$, $library_C$)= true.

synonym_concepts($people_B$, $person_C$)= true.

synonym_concepts($staff_B$, $staff_C$)= true.

synonym_concepts($user_B$, $user_C$)= true.

Dos ontologías son inconsistentes si existe algún par de conceptos, uno perteneciente a cada ontología de forma que ambos conceptos son inconsistentes, o bien desde el punto de vista de los atributos o bien desde el punto de vista de la estructura. Esta definición se puede formalizar de la siguiente forma:

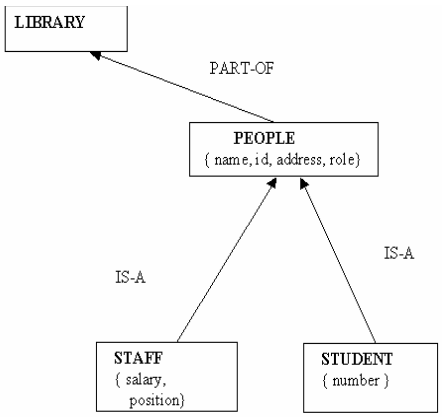


**Figura 3.7. Subontología de la ontología B**

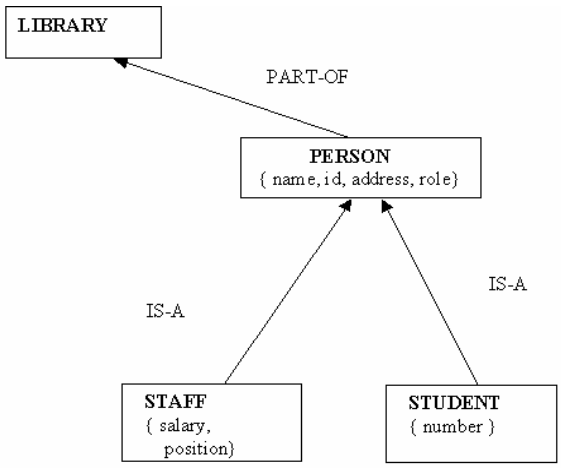


**Figura 3.8. Subontología de la ontología C**

Definición 10: ontologías inconsistentes

Sean $PMHRD_i(t)$ y $PMHRD_j(t)$ dos PMHRD(t) correspondientes a las ontologías $O_i(t)$ y $O_j(t)$, respectivamente, consideradas en el instante t. Se dice que $O_i(t)$ y $O_j(t)$ son *ontologías inconsistentes*, denotado por *inconsistent_ontologies* $(O_i(t), O_j(t))$, sí y solo si:

c(t) exists in $PMHRD_i(t)$, c'(t) exists in $PMHRD_j(t)$ such that incompatible_concepts(c(t),c'(t)), donde $PMHRD_i(t)$ y $PMHRD_j(t)$ representan los PMHRD(t) correspondientes a las ontologías $O_i(t)$ y $O_j(t)$, respectivamente.

Las ontologías correspondientes a las Figuras 3.3 y 3.5 ($O_A(t)$ y$O_C(t)$ ) son inconsistentes puesto que hay dos conceptos incompatibles (uno de cada ontología), como son, "$article_A$" y "$article_C$". Se dice que dos ontologías son compatibles sí y sólo si no son equivalentes ni inconsistentes. Formalmente:

Definición 11: ontologías compatibles

Sean $O_i(t)$ y $O_j(t)$ dos ontologías. Se dice que $O_i(t)$ y $O_j(t)$ son *ontologías compatibles*, denotado por *compatible_ontologies*, sí y sólo si not (equivalent_ontologies($O_i(t)$,$O_j(t)$)) and not (inconsistent_ontologies($O_i(t)$, $O_j(t)$)).

Las ontologías correspondientes a las Figuras 3.3 y 3.4 ($O_A(t)$ y $O_B(t)$ respectivamente) son dos ontologías compatibles, puesto que no son equivalentes ni inconsistentes. No son equivalentes porque no hay conceptos en $O_B(t)$ que sean sinónimos a los siguientes conceptos $O_A(t)$: { book, novel, essay }. Por otro lado, tampoco son inconsistentes porque no podemos encontrar dos conceptos, uno de cada ontología, que sean incompatibles. Por consiguiente, las dos ontologías son compatibles. En este momento resulta conveniente presentar un resumen de las distintas funciones definidas hasta ahora en esta sección. La Tabla 3.1 describe las funciones aplicables a dos conceptos. La Tabla 3.2 describe las funciones diseñadas para conjuntos de conceptos mientras que la Tabla 3.3 presenta funciones para dos ontologías. Sin embargo, antes

de ver estas tablas definiremos la noción de ontología inicializada. Consideramos que una ontología está inicializada si se dan las siguientes condiciones:

*topic* es el dominio en cuestión, que viene contenido en la raíz, que no tiene hijos mereológicos y tiene un atributos llamado *number_of_active_user_nodes*;

cada ontología perteneciente a un usuario tiene un sufijo añadido a su nodo raíz que corresponde con el identificador del nodo.

| Función | Descripción informal |
|---|---|
| degree of overlapping | Calcula el grado de solapamiento entre dos conceptos |
| A_equivalency | Chequea si dos conceptos son equivalentes desde el punto de vista de los atributos |
| A_inconsistency | Chequea si dos conceptos son inconsistentes desde el punto de vista de los atributos |
| OS_equivalency | Chequea si dos conceptos son equivalentes desde el punto de vista de la estructura |

| | |
|---|---|
| OS_inconsistency | Chequea si dos conceptos son inconsistentes desde el punto de vista de la estructura |
| synonym concepts | Chequea si dos conceptos son equivalentes |
| incompatible concepts | Chequea si dos conceptos pueden tomar parte del mismo proceso de integración |

**Tabla 3.1. Funciones para dos conceptos**

| **Función** | **Descripción informal** |
|---|---|
| **EA_equivalency** | **Chequea si dos conjuntos de conceptos son equivalentes desde el punto de vista de los atributos** |
| **Dis_concept_sets** | **Chequea si dos conjuntos de conceptos son disjuntos** |

**Tabla 3.2. Funciones para conjuntos de conceptos**

| **Función** | **Descripción informal** |
|---|---|
| **equivalent ontologies** | **Chequea si dos ontologías son equivalentes** |
| **inconsistent ontologies** | **Chequea si dos ontologías son inconsistentes** |
| **compatible ontologies** | **Chequea si dos ontologías pueden tomar parte del mismo proceso de integración sin introducir redundancia o inconsistencia** |

**Tabla 3.3. Funciones para dos ontologías**

Definición 12: ontología derivada de la integración

Sea *topic* el dominio sobre el que el usuario requiere información ontológica; sea *number_of_active_user_nodes* un atributo del concepto raíz de una ontología, denotado por $O_{init}(t)$, estableciendo el número de usuarios-nodo que participan en el proceso de integración de ontologías; y sea $O_i$ la ontología correspondiente al i-ésimo usuario-nodo, i = 1,…,val(n) donde val(n) es el valor del atributo number_of_active_user_nodes. Sea candidates(t) el conjunto de ontologías a integrar. El primer paso es eliminar de este conjunto aquellas ontologías que introducen redundancia o inconsistencia en el proceso. Con este propósito, sea $compatible_i(t)$ el conjunto de ontologías $O_j(t)$ pertenecientes a candidates(t) que son compatibles con $O_i(t)$ y con los elementos de $compatible_i(t)$, i=1..val(n). Así, esta operación calcula los conjuntos de ontologías que pueden tomar parte conjuntamente en el proceso de integración. Sea subset(t) el mejor subconjunto de todos los conjuntos $compatible_i(t)$, i=1..val(n).

Consecuentemente, la ontología derivada de la integración $O_{init}(t)$ es la que cumple las siguientes propiedades:

(root($O_{init}$(t)) = *topic*) and (ATT(root($O_{init}$(t))) = {*number_of_active_user_nodes*}) and (for all i, root($O_i$(t))=root($O_i$(t))_user_node$_i$) and

$$(\text{M - children}(\text{root}(\text{Oinit}(t))) = \bigcup_{i=1}^{val(n)} \{\text{root}(O_i(t))\})$$

i=1,.., card(subset(t)).

donde root($O_i$(t)) representa el concepto raíz de la i-ésima ontología en el instante t, y user_node$_i$ representa el identificador del i-ésimo usuario-nodo.

Por ejemplo, supongamos que hay cuatro usuarios, a los que nos referimos como experto A ($O_A$(t)), experto B ($O_B$(t)), experto C ($O_C$(t)), y experto D ($O_D$(t)), quienes están construyendo una ontología sobre el dominio Library. Supongamos también que sus respectivas ontologías privadas son las mostradas en las Figuras 3.3-3.6.

Veamos cómo procede el sistema para inicializar las ontologías para el proceso de integración. El primer paso es construir el conjunto de ontologías candidatas a tomar parte en el proceso de integración:

candidates(t) ={ $O_A$(t), $O_B$(t), $O_C$(t), $O_D$(t) }

Ahora podemos calcular las ontologías compatibles con $O_A$(t) (ver Tabla 3.6).

compatible$_{Oa(t)}$(t) = { $O_A$(t), $O_B$(t) } porque:

compatible_ontologies ($O_A$(t), $O_A$(t)) = true; y

compatible_ontologies ($O_A$(t), $O_B$(t)) = true; y

compatible_ontologies ($O_A(t)$, $O_C(t)$) = false; y

compatible_ontologies ($O_A(t)$, $O_D(t)$) = true.

La Tabla 3.4 muestra las compatibilidades para cada ontología candidata. En el próximo capítulo se desarrollará completamente este ejemplo. Allí se describirá el proceso completo para obtener la Tabla 3.4.

| compatibilities | $O_A$ | $O_B$ | $O_C$ | $O_D$ |
|---|---|---|---|---|
| $O_A$ | X | X | | X |
| $O_B$ | X | X | | X |
| $O_C$ | | X | X | |
| $O_D$ | X | X | | X |

**Tabla 3.4. Ontologías candidatas compatibles**

Llegados a este punto, el sistema debe elegir qué conjunto de ontologías compatibles se va a utilizar para la integración propiamente dicha. Supongamos que la petición de integración haya sido realizada por el experto A. Entonces, $O_A(t)$ debe estar incluida en el subconjunto elegido, de forma que los subconjuntos candidatos son tres: $compatible_{Oa(t)}(t)$, $compatible_{Ob(t),}$ y $compatible_{Od(t)}$, que contienen las mismas ontologías, de forma que subset = { $O_A(t)$, $O_B(t)$, $O_D(t)$ }. Sin embargo, en caso que diferentes subconjuntos contengan diferentes ontologías debe haber un criterio para decidir cuál es el mejor subconjunto. Se pueden adoptar diferentes criterios. En nuestro caso usaremos como criterio "el subconjunto con mayor número de ontologías es el mejor". Otros criterios podrían ser: (1) el subconjunto con el mayor número de entidades de conocimiento, esto es, la suma de conceptos, relaciones y atributos; (2) el subconjunto con mayor profundidad media; (3) el subconjunto que optimiza cierta función predefinida; u otros. Ahora podemos obtener la ontología derivada de la integración, que aparece en la Figura 3.9.

Mirando la definición anterior se puede notar que las contribuciones ontológicas están enlazadas al concepto raíz a través de una relación mereológica. El motivo para adoptar esta política es que asumimos que cada contribución es una pieza de conocimiento que forma parte de un conjunto llamado ontología derivada de la integración.

## 3.2.3 Usando ontologías derivadas de la integración

De acuerdo con los contenidos de secciones anteriores, cuando los usuarios están construyendo sus ontologías privadas podrían estar interesados en acceder a la ontología integrada, denotada por $O_{int}(t)$. Esto puede ser útil cuando un usuario desea conocer lo que han enviado los otros usuarios a $O_{int}$ (t). Por otro lado, deberíamos comentar que, dado un dominio sobre el que se va a construir la ontología integrada, no todos los usuarios saben lo mismo sobre el dominio. Además, el uso de términos diferentes para expresar el mismo concepto podría complicar el proceso de construcción de $O_{int}(t)$. Así, una política para gestionar términos sinónimos es un elemento razonable e interesante que ya ha sido incorporada en otros sistemas como WordNet (Miller, 1990).

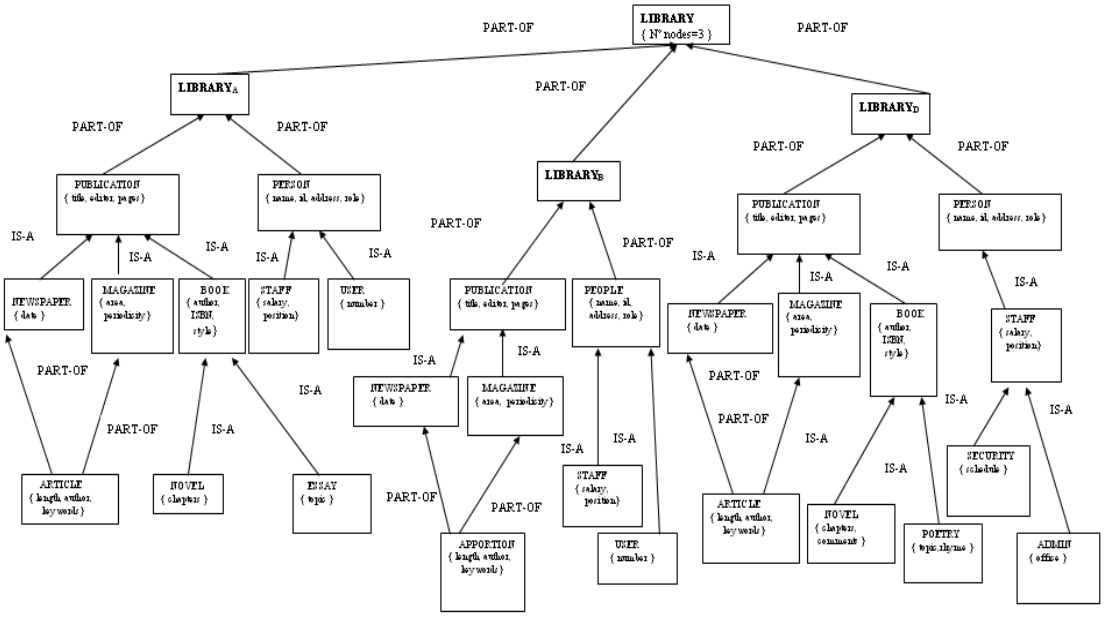


**Figura 3.9. Ontología derivada de la integración**

En el entorno propuesto en este artículo, cada vez que un usuario envía una consulta a $O_{int}(t)$, este usuario verá una $O_{int}(t)$ virtual. En esta ontología, los conceptos identificados como sinónimos de conceptos contenidos en la ontología del usuario que solicita la integración serán reemplazados por los términos correspondientes usados por el usuario en cuestión. Esta estrategia se puede expresar de la forma siguiente.

Dada una ontología derivada de la integración, $O_{int}(t)$, y una ontología correspondiente a un usuario, $O_i(t)$, ambas ontologías consideradas en el instante t, la ontología instanciada e integrada para ese usuario será la que cumpla las siguientes propiedades:

sus enlaces topológicos son los mismos que los existentes en $O_{int}(t)$; y

cada concepto debe pertenecer al PMHRD(t) correspondiente a la ontología $O_i(t)$ (si existe al menos un concepto perteneciente al PMHRD(t) correspondiente a $O_{int}(t)$ que sea equivalente al concepto en cuestión desde el punto de vista de los atributos y de la estructura) o bien al PMHRD(t) perteneciente a $O_{int}(t)$; y

cada concepto perteneciente al PMHRD(t) correspondiente a $O_{int}(t)$ debe pertenecer al PMHRD(t) correspondiente a $O_i(t)$ (si hay al menos un concepto perteneciente a la ontología integrada e instanciada que sea equivalente al concepto en cuestión desde el punto de vista de los atributos y de la estructura) o bien a la ontología integrada e instanciada.

La ontología $O_i(t)$ es la ontología de referencia para el proceso de instanciación puesto que la terminología de la ontología derivada de la integración se adapta a la terminología de $O_i(t)$. La selección de la ontología de referencia merece una explicación más detallada. El entorno maneja dos tipos de usuarios, expertos y normales. La diferencia principal es que los usuarios normales no pueden construir ontologías, por lo que sólo puede solicitar la integración del conocimiento relativo a un dominio determinado. Por lo tanto, cuando un usuario solicita información sobre un tema, si ha construido una ontología sobre ese dominio, dicha ontología será la utilizada como ontología referencia. En el supuesto de que el experto no tenga ninguna ontología en ese dominio, el experto será tratado como un usuario normal y la ontología de

referencia será determinada en función del principio de máximo contenido informativo introducido anteriormente.

Podemos formalizar esta idea como sigue:

Definición 13: ontología instanciada e integrada

Sea $O_i(t)$ una ontología considerada en el instante t y sea $O_{int}(t)$ la ontología derivada de la integración hasta el instante t. La *ontología instanciada e integrada con respecto a* $O_i(t)$, denotado por $O_{int}^{i}(t)$ *o IDO(t)*, se define como una ontología que cumple las siguientes condiciones:

(1) sus enlaces topológicos son los mismos que los de $O_{int}(t)$;

(2) $\forall\ c'(t) \in PMHRD'(t)$ se cumple que:

$$c'(t) \in \begin{cases} PMHRD_i(t) & \text{iff exists } c_{int}(t) \in PMHRD_{int}(t) \text{ s.t.} \\ & syn\_concepts(c'(t), c_{int}(t)) \\ PMHRD_{int}(t) & \text{otherwise} \end{cases}$$

(3) $\forall\ c_{int}(t) \in PMHRD'(t)$ se cumple que:

$$c_{int}(t) \in \begin{cases} PMHRD_i(t) & \text{iff exists } c'(t) \in PMHRD'(t) \text{ s.t.} \\ & syn\_concepts(c_{int}(t), c'(t)) \\ PMHRD'(t) & \text{otherwise} \end{cases}$$

donde

$PMHRD_x(t)$ representa el PMHRD(t) correspondiente a la ontología $O_x(t)$, $x \in \{int, i\}$;

PMHRD'(t) representa el PMHRD(t) correspondiente a la ontología $O^i_{int}(t)$.

Según el ejemplo seguido en este capítulo, podemos construir la ontología instanciada e integrada con respecto a $O_B(t)$, denotada por $O_{int}{}^B(t)$, como se muestra en la Figura 3.10, donde el identificador escrito como superíndice en $O_{int}{}^B(t)$ ha sido omitido por cuestiones de legibilidad. Esa figura representa la ontología instanciada e integrada correspondiente a la ontología derivada de la integración mostrada en la Figura 3.9. Supongamos que la petición de integración fue emitida por el experto B, de forma que $O_B(t)$ (Figura 3.3) es la ontología de referencia para instanciar la ontología. Ilustremos la aplicación de la definición 13 a la ontología derivada de la integración que aparece en la Figura 3.9. Dicha ontología se compone de tres ontologías de expertos ($O_A(t)$, $O_B(t)$ y $O_D(t)$ ). Cada concepto de cada ontología debe ser procesado para buscar conceptos sinónimos en el resto de ontologías. Cada ontología tiene los siguientes conceptos:

**$PHMRD_A(t)$** = {publication, newspaper, magazine, book, article, novel, essay, person, staff, user}

**$PHMRD_B(t)$** = {publication, newspaper, magazine, apportion, people, staff, user}

**$PHMRD_D(t)$** = {publication, newspaper, magazine, book, article, novel, poetry, person, staff, security, admin}

Los conceptos pertenecientes a $PHMRD_B(t)$ no cambiarán su nombre puesto que $O_B(t)$ es la ontología referencia. Por lo tanto, este paso solo se aplicaría a los conceptos de $PHMRD_A(t)$ y $PHMRD_D(t)$. Cuando buscamos conceptos sinónimos, el algoritmo puede responder de dos maneras, de forma afirmativa o negativa. Si la respuesta es negativa, no se hace nada porque no

se ha encontrado ningún concepto sinónimo. Si la respuesta es afirmativa pueden ocurrir dos cosas: (1) tienen el mismo nombre (en esta caso no se haría nada; es el caso de "publication", "newspaper", "magazine", "staff" y "user"); (2) tienen nombres distintos.

Dada esta situación, el nombre del concepto que no pertenece a $PHMRD_B(t)$ es sustituido por el nombre del concepto sinónimo en la ontología referencia. Por ejemplo, el nombre del concepto $person_A$ es cambiado a "people". Sin embargo, no se pierde el nombre asignado por el experto A, sino que se mantiene como nombre alternativo. Esta información se mantiene en ambos conceptos. La Figura 3.10 muestra la ontología instanciada y derivada de la integración.

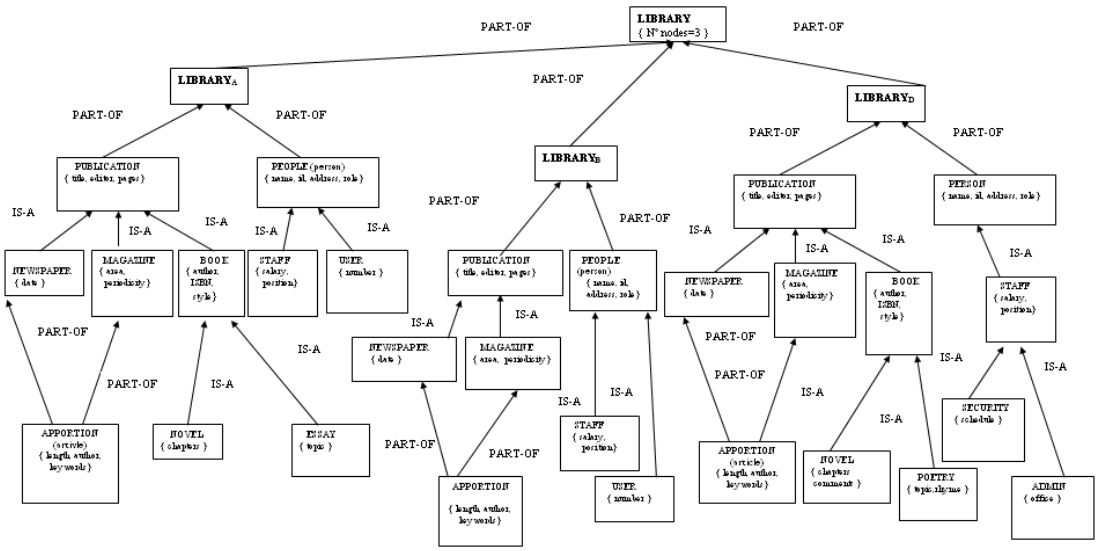


**Figura 3.10. Ontología instanciada e integrada**

A continuación podemos describir un algoritmo para generar ontologías transformadas usando las definiciones introducidas hasta el momento. Por transformada entendemos que no se pretende una traducción precisa de la información contenida en $O_g(t)$, sino una integración de esa porción de información, esto es, el conocimiento inferido de $O_g(t)$ que es consistente con los principios explicados en capítulos anteriores. Cada vez que un usuario final envíe una solicitud de información al sistema, dicho usuario recibirá una ontología que cumpla el principio de máximo contenido informativo.

Además, los atributos de cada concepto de esta ontología final son obtenidos a partir de la ontología derivada de la integración. Para ello se toma en cuenta la contribución de todos los usuarios. Estas consideraciones han sido utilizadas en las siguientes definiciones. Si el conjunto de ontologías pertenecientes a IDO(t) se clasifica en orden decreciente de número de conceptos, entonces la primera ontología de este conjunto se denomina personalised_ontology en el instante t. Podemos expresarlo a través de la siguiente definición:

Definición 14: ontología personalizada

Sea IDO(t) una ontología derivada de la integración hasta el instante t; y sea Onto_set(t) un conjunto no ordenado conteniendo las ontologías pertenecientes a IDO(t) y que cumple que Cardinal($PMHRD_i(t)$)≥ Cardinal($PMHRD_j(t)$) sí y sólo si i<j, donde $PMHRD_k(t)$ representa el PMHRD(t) correspondiente al k-ésimo elemento de Onto_set(t), denotado por $Onto\_set_k(t)$ $k \in \{i,j\}$, $1 \leq i, j \leq$ Cardinal(M-children(root(IDO(t))). La *ontología personalizada hasta el instante t correspondiente a Onto_set(t)*, denotado por *PO(t)*, es definido como $Onto\_set_1(t)$.

Esta definición puede ser vista como la inicialización de la ontología transformada derivada de la integración, que es el resultado final del proceso. PO(t) puede ser considerado como el esqueleto usado para fusionar las ontologías de diferentes expertos, así que se incluirán nuevos conceptos, atributos y relaciones para generar la ontología transformada y derivada de la integración. La definición 14 explica la estructura de Onto_set(t) cuando no hay ontología de referencia explícita para el proceso, de forma que se ordenan las ontologías según el máximo contenido informativo. Cuando el proceso tiene una ontología referencia, Onto_Set(t) se ordena de forma diferente:

$Onto\_set_1(t)$= ontology referencia.

El resto de Onto_set (t) es un conjunto ordenado y no vacío que contiene las ontologías pertenecientes a IDO(t) y que cumple que Cardinal($PMHRD_i(t)$)≥ Cardinal($PMHRD_j(t)$), tal que $1< i, j \leq$ Cardinal(M-children(root(IDO(t))), puesto que $Onto\_set_1(t)$ es la ontología referencia.

Tomando en cuenta el ejemplo anterio, podemos afirmar que en este caso Onto_set(t)={$O_D$(t), $O_A$(t), $O_B$(t)}, y por lo tanto PO(t)=$O_D$(t). Se debe decir que la ontología asignada a PO(t) será siempre la ontología referencia cuando ésta exista. En caso de no existir se aplicará el principio de máximo contenido informativo. Si la petición de integración es realizada por el experto B, entonces PO(t) sería $O_B$(t) y Onto_set(t)={$O_B$(t), $O_D$(t), $O_A$(t)}.

Un concepto es una versión actualizada de otro si los términos dados a ambos conceptos son el mismo y los atributos específicos del segundo son un subconjunto de los atributos específicos del primero. Formalmente:

Definición 15: concepto actualizado

Sea $PMHRD_{po}$(t) el PMHRD(t) correspondiente a una ontología personalizada hasta el instante t; sea $PMHRD_i$(t) un PMHRD(t) correspondiente a la ontología en proceso; sean $c_1$(t) y $c_2$(t) dos conceptos pertenecientes a $PMHRD_i$(t) y $PMHRD_{po}$(t) respectivamente. Entonces, se dice que $c_2$(t) es un *concepto actualizado de $c_1$(t)*, denotado por *updated_concept($c_1$(t), $c_2$(t))*, si (NAME($c_1$(t)) = NAME($c_2$(t))) and(SPE($c_1$(t)) $\subseteq$ SPE($c_2$(t))) donde NAME(x) representa al término dado al concepto x in $PMHRD_j$(t), x $\in$ { $c_1$(t), $c_2$(t)} , j $\in$ { po, i }.

A continuación presentamos cinco nuevas definiciones: ontología personalizada actualizada (PUO), ontología libre de ambigüedad (AFO), ontología libre de conflictos semánticos (SCFO), ontología transformada (TO), y ontología transformada y derivada de la integración (TIDO). Hasta ahora, se ha ilustrado cada definición con un ejemplo de aplicación, lo cual no sucederá con este grupo de definiciones. Existen algunas dependencias entre estas definiciones así que, para simplificar la comprensión y con el propósito de usar el mismo ejemplo en toda la sección, la aplicación práctica de estas definiciones aparecerá tras la definición TIDO.

Se dice que una ontología personalizada PO(t) es una ontología personalizada actualizada si, dado un concepto de una ontología entrante, hay algún otro concepto en aquélla tal que existe un concepto padre de aquel concepto presente en la última ontología, cumpliendo que son

conceptos sinónimos y que el conjunto de atributos específicos del nuevo concepto están incluidos en los del concepto presente en PO(t). (En este caso, el concepto de la ontología entrante será incorporado como hijo taxonómico/mereológico del concepto de la ontología personalizada). Esto se puede expresar formalmente de la siguiente forma:

Definición 16: ontología personalizada actualizada

Sea $PMHRD_{po}(t)$ el PMHRD(t) correspondiente a la ontología personalizada hasta el instante t, denotado por PO(t); sea $c_{po}(t)$ un concepto perteneciente a $PMHRD_{po}(t)$; y sea $c_{in}(t)$ un concepto perteneciente a una ontología candidata parcial, denotada por $O_{in}(t)$. Entonces, la *ontología personalizada actualizada construida a partir de PO(t), $c_{po}(t)$, $c_{in}(t)$ y $O_{in}(t)$*, denotada por *PUO(PO(t), $c_{po}(t)$, $c_{in}(t)$, $O_{in}(t)$)* se define como la modificación de PO(t) tal que:

T-children($c_{po}(t)$) = T-children($c_{po}(t)$) $\cup$ {$c_{in}(t)$} (if more_knowledge($c_{po}(t)$, $c_{in}(t)$)='taxonomic')

***or*** M-children($c_{po}(t)$) = M-children($c_{po}(t)$) $\cup$ {$c_{in}(t)$} (if more_knowledge($c_{po}(t)$, $c_{in}(t)$)='mereorological')

donde

more_knowledge($c_{po}(t)$, $c_{in}(t)$) = 'taxonomic' iff exists $c'_{in}(t) \in PMHRD_{in}(t)$ such that [$c'_{in}(t) \in$ T-parents($c_{in}(t)$)] and updated_concept($c'_{in}(t)$, $c_{po}(t)$);

more_knowledge($c_{po}(t)$, $c_{in}(t)$) = 'mereorological' iff exits $c'_{in}(t) \in PMHRD_{in}(t)$ such that [$c'_{in}(t) \in$ M-parents($c_{in}(t)$)] and updated_concept($c'_{in}(t)$, $c_{po}(t)$)

Cuando buscamos la ubicación correcta del concepto $c_{in}(t)$ en la ontología PO(t), cada concepto $c_{po}(t)$ debe ser comprobado porque cada concepto tiene potencialmente una relación con $c_{in}(t)$. PUO es la definición encargada de descubrir las relaciones reales entre $c_{in}(t)$ y los

conceptos en PO(t). Si se encuentra una relación entre $c_{po}(t)$ y $c_{in}(t)$, ésta será añadida como hijo del anterior. Estas relaciones son establecidas cuando se encuentran conceptos padre de $c_{in}(t)$ en PO(t). Esto es indicado por la función updated_concept, puesto que los padres de $c_{in}(t)$ en PO(t) ya habrán sido incluidos en PO(t).

Dada una ontología personalizada PO(t), se dice que una ontología está libre de ambigüedades dado un concepto $c_{in}(t)$ perteneciente a una ontología entrante, si se dan las siguientes condiciones:

el término dado a este concepto tiene, en $O_{in}(t)$, el identificador del suministrador de la ontología como sufijo; y

PO(t) es una ontología personalizada actualizada; y

El conjunto de atributos específicos de $c_{in}(t)$ está formado por los atributos específicos de $c_{in}(t)$ en el PMHRD(t) correspondiente a PO(t).

Podemos formalizar lo anterior como sigue:

Definición 17: ontología libre de ambigüedades

Sea $PMHRD_{po}(t)$ un PMHRD(t) correspondiente a la ontología personalizada hasta el instante t, denotado por PO(t); sea $c_{in}(t)$ un concepto perteneciente al PMHRD(t) correspondiente a la ontología candidata entrante, denotada por $O_{in}(t)$; y sea $c^{p}_{in}(t)$ $c_{in}(t)$ en $PMHRD_{po}(t)$. Entonces, se define la *ontología libre de ambigüedades construida a partir de PO(t), $c_{in}(t)$ y $O_{in}(t)$*, denotada por *AFO(PO(t),$c_{in}(t)$, $O_{in}(t)$)*, como una modificación de PO(t) que cumple las siguientes condiciones:

(1) $NAME(c_{in}^{p}(t)) = NAME(c_{in}(t))$; y

(2) $SPE(c_{in}^{p}(t)) = SPE(c_{in}(t))$; y

(3) for all c(t) in $PMHRD_{po}(t)$, $PUO(PO(t),c(t), c_{in}(t), O_{in}(t))$

donde $PUO(PO(t),c(t), c_{in}(t), O_{in}(t))$ representa la ontología personalizada actualizada construida a partir de PO(t), c(t), $c_{in}(t)$ y $O_{in}(t)$,

Dada una ontología personalizada, denotada por PO(t), la podemos definir como ontología libre de conflictos semánticos con respecto a un concepto, denotado por $c_{in}(t)$, que pertenece a una ontología entrante si se cumple una de las condiciones siguientes:

Para cada concepto en PO(t), su conjunto de atributos específicos contiene el conjunto de atributos de $c_{in}(t)$ (Esto ocurre si todos estos conceptos son equivalentes a $c_{in}(t)$ desde el punto de vista de los atributos y de la estructura de la ontología);

PO(t) es una ontología libre de ambigüedades con respecto a $c_{in}(t)$ (Esto ocurre si existe algún concepto que pertenezca al PMHRD(t) correspondiente a PO(t) tal que los términos dados a ambos conceptos son el mismo y ambos conceptos no son equivalentes desde el punto de vista de los atributos o de la estructura);

Formalmente:

Definición 18: ontología libre de conflictos semánticos

Sea $PMHRD_{po}(t)$ un PMHRD(t) correspondiente a una ontología personalizada hasta el instante t, denotada por PO(t); sea $c_{in}(t)$ un concepto perteneciente al PMHRD(t), denotado por $PMHRD_{in}(t)$, correspondiente a una ontología candidata entrante, denotada por $O_{in}(t)$. Entonces, se define la *ontología libre de conflictos semánticos construida a partir de PO(t)*, $c_{in}(t)$ *y* $O_{in}(t)$, denotada por *SCFO(PO(t),* $c_{in}(t)$, $O_{in}(t)$*)*, como sigue:

Una modificación de PO(t) de forma que para todo c(t) en $PMHRD_{po}(t)$, $SPE(c(t)) = SPE(c(t)) \cup SPE(c_{in}(t))$ si $syn\_concepts(c(t), c_{in}(t))$[1]; o

Una modificación de PO(t) de forma que $SPE(c(t)) = SPE(c(t)) \cup SPE(c_{in}(t))$ si existe $c(t) \in PMHRD_{po}(t)$ tal que $[not\ (syn\_concepts(c(t), c_{in}(t)))]$ and $(NAME(c(t)) = NAME(c_{in}(t)))$; o

Una modificación de PO(t) de forma que $AFO(PO(t), c_{in}(t), O_{in}(t))$ si para todo $c'(t) \in PMHRD_{po}(t)$ tal que $[not\ (syn\_concepts(c(t), c_{in}(t)))]$ donde $AFO(PO(t), c_{in}(t), O_{in}(t))$ es la ontología libre de ambigüedades construida a partir de PO(t), $c_{in}(t)$ y $O_{in}(t)$.

Dada una ontología personalizada PO(t), se dice que es una ontología transformada obtenida a partir de PO(t) y de una ontología entrante $O_{in}(t)$ si PO(t) es una ontología libre de conflictos semánticos con respecto a cada concepto incluido en el PMHRD(t) correspondiente a $O_{in}(t)$. Podemos formalizar esta idea de la manera siguiente.

[1] Esta condición tendría mayor importancia si la A-equivalencia entre conceptos no hubiera sido definida únicamente como una comparación de nombres

Definición 19: ontología transformada

Sea PO(t) una ontología personalizada hasta el instante t y sea $PMHRD_{in}(t)$ el PMHRD(t) correspondiente a una ontología entrante, denotada por $O_{in}(t)$. Entonces, la *ontología transformada obtenida a partir de PO(t) y $O_{in}(t)$*, denotada por *$TO(PO(t),O_{in}(t))$*, se define como una modificación de PO(t) de forma que para todo c(t) en $PMHRD_{in}(t)$, se cumple $SCFO(PO(t), c(t),O_{in}(t))$ donde $SCFO(PO(t), c(t),O_{in}(t))$ representa la ontología libre de de conflictos semánticos construida a partir de PO(t), c(t) y $O_{in}(t)$).

Supongamos que disponemos de una ontología personalizada PO(t) y de un conjunto ordenado no vacío, cuyo cardinal sea al menos dos, que contenga las ontologías pertenecientes a IDO(t). Entonces, se dice que TO(t) es una ontología transformada y derivada de la integración obtenida a partir de PO(t), si TO(t) es una ontología transformada a partir de sí misma y de todas las ontologías de dicho conjunto ordenado. La siguiente definición formaliza esta idea:

Definición 20: ontología transformada derivada de la integración

Sea IDO(t) una ontología instanciada y derivada de la integración hasta el instante t; sea Onto_set(t) un conjunto ordenado no vacío conteniendo las ontologías pertenecientes a IDO(t) y cumpliendo que $Cardinal(Onto\_set(t)) \geq 2$ y que $Cardinal(PMHRD_i(t)) \geq Cardinal(PMHRD_j(t))$ sí y solo si $i<j$, donde $PMHRD_k(t)$ representa al PMHRD(t) correspondiente al k-ésimo elemento de Onto_set(t); y sea PO(t) una ontología personalizada hasta el instante t. Entonces, se define la *ontología transformada y derivada de la integración obtenida a partir de PO(t) y Onto_set(t)*, denotada por *TIDO(PO(t), Onto_set(t)),* como la modificación de PO(t) cumpliendo que para todo O(t) en Onto_set(t)\{PO(t)}, se cumple TO(PO(t), O(t)).

Finalmente consideraremos que TIDO(PO(t), Onto_set(t)) = PO(t) si Cardinal(Onto_set(t)) = 1, esto es, dada una ontología transformada y derivada de la integración, ésta no sufre modificación alguna si sólo hay uan ontología en Onto_set(t).

Identifiquemos cada elemento de esta definición en nuestro ejemplo IDO(t) es la ontología mostrada en la Figura 3.10, Onto_set es el conjunto de ontologías {$O_B(t)$, $O_D(t)$ , $O_A(t)$} que cumple la condición de ordenación requerida. PO(t) es $O_B(t)$, al ser $O_B(t)$ la ontología referencia. Entonces, la ontología obtenida al aplicar TIDO($O_B(t)$, {$O_B(t)$, $O_D(t)$, $O_A(t)$ }\{$O_B(t)$) es la ontología mostrada en la Figura 3.11. Este ejemplo se detalla en el siguiente capítulo.

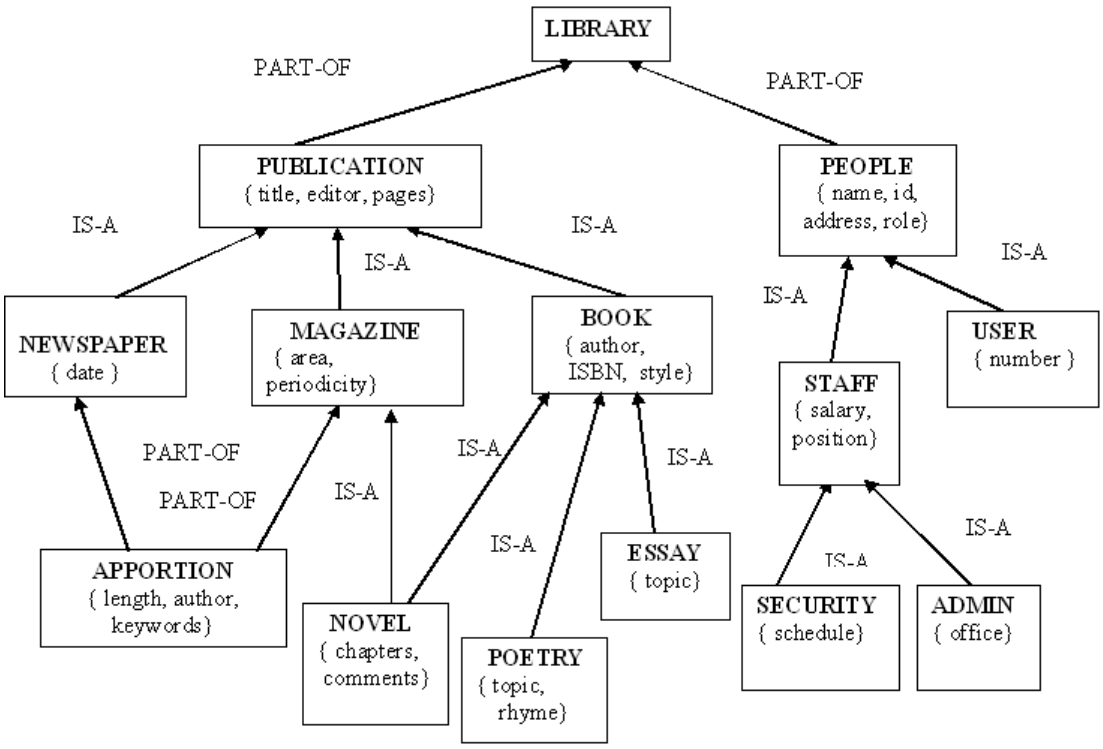


**Figura 3.11.La ontología transformada y derivada de la integración**

## 3.3 UNA VISIÓN ALGORÍTMICA DE LOS PROCESOS DE INTEGRACIÓN

Se ha formalizado un entorno completo para integrar ontologías. Sin embargo, la presentación de un enfoque algorítmico podría facilitar la comprensión del proceso. El proceso de integración se puede dividir en varios procesos, denominados selección, instanciación y transformación. Dichos procesos son descritos en los próximos apartados.

### 3.3.1 Algoritmo general de integración de ontologías

Sea $O_i(t)$ la i-ésima ontología a incorporar a $O_{int}(t)$; n = número de ontologías a integrar (tanto las ontologías ya existentes como las nuevas). Sea candidates(t) el conjunto de ontologías a integrar.

For i=1 to n

If (existe alguna ontología $O_j(t)$ incluida en candidates(t) tal que $O_i(t)$y $O_j(t)$

pertenezcan al mismo usuario) then (eliminar de candidates(t) la ontología

más antigua)

End-if

End-for

subset= Select_Ontologies(candidates(t))

i=1

While i ≤ Card(subset) do

Añadir $O_i(t)$ a $O_{int}(t)$ como hijo mereológico, de forma que su raíz tenga como nombre

"topic-according to-user i"

End-while

Ontology_Instantiation($O_{int}(t)$)

Ontology_ Transformation($O_{int}(t)$)

End

## 3.3.2 Select_Ontologies

Sea candidates(t) el conjunto de ontologías candidatas a integrar. Sea $compatible_i(t)$ el conjunto de ontologías $O_j(t)$ pertenecientes a candidates(t) que son compatibles a $O_i(t)$.

For i=1 to Card(candidates(t))

For i=j to Card(candidates(t))

compatiblei(t)=compatiblei(t) ∪ Oj (t)si compatible(Oi(t), Oj(t))

End-For

End-For

Devolver el mejor subconjunto según el criterio utilizado (p.ej., el subconjunto con mayor número de ontologías)

End

donde

compatible(x,y) es true sí y solo si (not(inconsistent(x,y) or equivalent(x,y)));

equivalent(x,y) es true sí y solo si para cada concepto perteneciente a x, existe otro en y tal que:

sus conjuntos de atributos son equivalentes; y

sus respectivos conceptos padres e hijos son equivalentes

*inconsistent(x,y)* es true sí y solo si existen al menos 2 conceptos, uno perteneciente a $O_i(t)$ y otro a $O_{int}(t)$, de forma que se cumple una de las condiciones siguientes:

Ambos tienen el mismo nombre, no tienen ningún atributo en común y sus respectivos conceptos padres/hijos, en caso de tenerlos, tienen atributos equivalentes.

Tienen los mismos atributos pero no hay otro concepto que sea padre de uno de ellos con los mismos atributos que uno de los padres del otro concepto. Lo mismo se puede decir para el caso de los conceptos hijo.

### 3.3.3 Ontological_ Instantiation

El siguiente paso es transformar la ontología integrada en una nueva ontología que pueda ser accedida por los usuarios de forma global, adaptando la terminología que aparece en la ontología derivada de la integración a la terminología de los usuarios. Este proceso implica la adaptación de la terminología de cada ontología incluida en la ontología integrada a la terminología usada por el usuario que solicitó la integración

Esto es posible cuando el usuario solicitante posea una ontología en el dominio que se integra. Si no, la referencia dependerá de las preferencias del usuario. Este proceso de adaptación terminológica implica la necesidad de un mecanismo de detección y gestión de términos sinónimos. En nuestro enfoque se detectan conceptos sinónimos analizando equivalencias estructurales y basadas en atributos. A pesar de la adaptación automática de la terminología, cuando el nombre de un concepto se modifica para adaptarlo a la terminología usada por un usuario, el término original asignado a aquel concepto se guarda como nombre alternativo. De esta forma podemos ofrecer la mejor información posible al usuario, teniendo éste la potestad de elegir el término que considere más apropiado para cada concepto.

Algoritmo

Sea $O_{int}(t)$ la ontología derivada de la integración; sea $O_i(t)$ el i-ésimo hijo mereológico de $O_{int}(t)$, i=1,...,n; n= número de hijos mereológicos de $O_{int}(t)$; y sea O(t) la ontología referencia.

For i=1 to n

Para cada concepto c(t) en $O_i(t)$ do

Si existe algún concepto c'(t) en O(t) tal que

equivalents_concepts(c(t),c'(t))

then name(c(t))= name(c'(t))

End-if

End-for

donde *equivalent_concepts(x,y)* es true sí y sólo si:

Ambos conceptos tienen conjuntos de conceptos equivalentes.

Ambos conceptos tienen sus respectivos conceptos padres/hijos equivalentes.

## 3.3.4 Ontological_Transformation

Una vez unificada la terminología, se debe proceder a fusionar las ontologías incluidas en la ontología derivada de la integración para generar una única ontología que será mostrada al usuario. Podemos definir el siguiente algoritmo para esta tarea:

Algoritmo

Sea $O_{int}(t)$ la ontología derivada de la integración; sea $O_i(t)$ el i-ésimo hijo mereológico de $O_{int}(t)$, i=1,...,n; y n= número de hijos mereológicos de $O_{int}(t)$.

For i=1 to n

Para cada concepto c(t) en $O_i(t)$ do

Si existe algún concepto c'(t) en $O_{int}(t)$ tal que

equivalent_concepts(c(t),c'(t)) or (c(t) y c'(t) tienen el mismo nombre)

then unir_atributos_y_relaciones(c(t), c'(t))

else enlazar c(t) a sus padres en $O_{int}(t)$

End-for

End-for

donde *equivalent_concepts(x,y)* es true sí y sólo si:

Los conjuntos de atributos de ambos conceptos son equivalentes

Los conceptos padre/hijo de ambos conceptos son equivalentes

## 3.4 COMPARACIÓN CON OTROS ENTORNOS PARA LA CONSTRUCCIÓN COOPERATIVA DE CONOCIMIENTO

En esta sección se compara el entorno de integración de ontologías propuesto en esta tesis con distintos sistemas para la construcción cooperativa de componentes de conocimiento.

Con respecto al sistema presentado en (Euzenat, 1996), habría que comentar que en este enfoque la comunidad debe usar la misma terminología. En nuestro entorno, ésto se soluciona por un mecanismo para gestionar sinonimia, que permite que cada usuario use su propia terminología. Consideramos esto necesario para mantener la eficiencia lingüística durante el proceso de adquisición de conocimiento.

El sistema presentado en García (1996), que es similar al de Euzenat, tiene varios inconvenientes. El principal es la necesidad de consenso para establecer un "puente" entre dos conceptos similares, lo cual ralentiza el proceso de colaboración. Por otro lado, este proceso implica que un usuario deba conocer el vocabulario de los otros usuarios y examinar sus ontologías. Por el contrario, en el enfoque sugerido en esta tesis, los usuarios no tienen que conocer la terminología empleada por otros usuarios. Además, el refinamiento de las ontologías privadas de los usuarios es posible como resultado del proceso de diálogo colaborativo.

Wiederhold (1994) propone un entorno similar al presentado en esta tesis, ya que se asocian terminologías originales a ontologías privadas. Sin embargo se difiere en la forma en que se detectan los conflictos. En este trabajo se asume que los términos nunca significan lo mismo a no ser que se diga explícitamente a través de una regla. Este conjunto de reglas debe ser gestionado por un conjunto de colaboradores que deben encargarse de conocer toda la

terminología asociada a los posibles subdominios. En nuestro enfoque, la gestión y detección de conceptos sinónimos es realizada por el sistema.

Otra diferencia con el enfoque de Wiederhold es que sólo forman parte del proceso de fusión las partes comunes de los dominios implicados en el proceso. Para ello se define un álgebra con operadores binarios (unión, intersección y diferencia). Además, en nuestro entorno se permite refinar las ontologías privadas como resultado del proceso cooperativo, aspecto que no es posible en el enfoque de Wiederhold, puesto que las ontologías resultado del proceso de fusión son consideradas un producto final.

En (Fridman-Noy and Musen, 1999), las ontologías fuente deben cubrir dominios similares o solapados para ser integrados. En nuestro caso, las ontologías deben cubrir el mismo dominio, aunque la semántica de ambos procesos es similar. Con respecto al proceso de integración propiamente dicho, SMART(Fridman-Noy and Musen, 1999; Musen and Fridman-Noy, 1999) fusiona conceptos cuyas etiquetas lingüísticas sean iguales o similares (por ejemplo Military-Unit y Modern-Military Unit). Sin embargo, no se tienen en cuenta propiedades internas y estructurales. Esto queda parcialmente resuelto en PROMPT (Fridman-Noy and Musen, 2000). En los sistemas basados en PROMPT, el usuario siempre toma la decisión acerca de qué conceptos son sinónimos (y por tanto fusionables) porque PROMPT fue diseñado para apoyar el proceso de integración, mientras que nuestro objetivo es realizar integración automática de ontologías. Este enfoque también usa el concepto de ontología de referencia, que también se usa como referencia en la resolución de conflictos.

FCA-Merge (Stumme and Maedche, 2001) parte de unos supuestos diferentes. En nuestro caso partimos de un conjunto de ontologías sobre el mismo dominio, mientras que FCA-Merge requiere de un conjunto adecuado de documentos en lenguaje natural para apoyar el proceso de razonamiento basado en instancias. La integración se basa en los contextos encontrados para ambas ontologías en el conjunto de documentos, mientras que en nuestro enfoque nos basamos en la estructura de las ontologías a integrar. Asimismo, la ontología no se genera de forma automática en FCA-Merge. Se realizan sugerencias basadas en el análisis de contextos y el usuario deberá tomar la decisión final.

Otro tema interesante a tratar es el de las inconsistencias. En (Tamma and Bench-Capon, 2001) se especifican dos tipos de inconsistencias: semánticas y estructurales. Las inconsistencias estructurales son debidas a diferencias en propiedades conceptuales, mientras que las semánticas son las causadas por la diferencia en semántica y nivel de granularidad de la representación. Para nosotros, la semántica ontológica incluye estructura, así que todas las inconsistencias encontradas son semánticas, aunque pueden ser extraídas desde un punto de vista estructural o de atributos.

Chimaera (McGuiness et al, 2000) tampoco realiza integración automática de ontologías, sino que también crea una lista de sugerencias de conceptos a integrar. Se usan heurísticas basadas principalmente en similaridad lingüística y en la existencia de enlaces taxonómicos, mientras que nuestro enfoque tiene en cuenta más factores a la hora de realizar la integración.

En APECKS (Tennison and Shadbolt, 1998) se muestran todas las ontologías al usuario, de forma que cualquier usuario puede consultar una ontología dada y, dependiendo de su familiaridad con el dominio en cuestión, puede no entender su contenido. Esta confusión puede ser causada por la complejidad e inconsistencia de un elevado número de conceptos que se le muestran. En nuestro enfoque, los usuarios sólo reciben la ontología construida por el sistema como resultado de la integración, lo cual ayuda a evitar confusiones.

En el sistema de Nakata (Nakata et al, 1998), los usuarios pueden modificar el conocimiento de forma no muy recomendable para los generadores de conocimiento colaborativo. En nuestro enfoque se resuelve este problema puesto que los usuarios finales sólo pueden consultar y no modificar la ontología integrada, y los expertos sólo pueden trabajar con sus ontologías privadas. Otra característica importante de este enfoque es que, a través del diálogo cooperativo proporcionado por dicho entorno, los usuarios pueden consultar el contenido de la ontología construida cooperativamente en un instante determinado para recoger la máxima información posible mientras construyen sus ontologías privadas. La gestión de términos sinónimos adapta la información requerida por el usuario en cuestión a su propia terminología. En otras palabras, cada agente tiene su particular visión, la cual evoluciona con el tiempo, de la ontología integrada.

Nuestra noción de ontologías candidatas es diferente al presentado en (Pinto and Martins, 2001). En este enfoque, las ontologías candidatas son ontologías que aún no toman parte del proceso de integración, mientras que para nosotros las ontologías candidatas sí toman parte del proceso de integración aun cuando algunas de ellas puedan ser descartadas. En ambos enfoques se descartan ontologías, aunque se usan criterios diferentes para ello.

La integración y el proceso de la transformación es, en parte, un proceso comparativo como el presentado en (Shaw and Gaines, 1989) pero con diferencias significativas. Allí, como resultado del proceso de comparación, se agrupan los conceptos en cuatro categorías, a saber, consenso, conflicto, correspondencia, y contraste. Entonces, los conflictos deben resolverse o marcarse como irresolubles. Éste es un proceso manual que requiere la participación activa de los expertos. Por otro lado, nuestro trabajo hace la comparación automática de sistemas conceptuales, ontologías en nuestro caso. En nuestro entorno podemos distinguir cuatro relaciones. El consenso es reflejado en nuestro entorno como equivalencia de conceptos; los conflictos irresolubles son nuestra fuente de inconsistencias; la correspondencia es la sinonimia; y finalmente, el contraste aparece cuando los expertos definen partes independientes del mismo dominio. Para nosotros, la única tarea del experto es crear el conocimiento, y el sistema se encarga de comparar este conocimiento con el conocimiento de otros expertos. Además, también difiere la manera en que también se categorizan los conceptos. En (Shaw and Gaines, 1989), la categorización se basa en el resultado cuantitativo del proceso de comparación mientras que nuestras decisiones se toman en base a factores cualitativos, como son la existencia de atributos o relaciones comunes.

Nuestro entorno también se puede usar para el desarrollo cooperativo de memorias corporativas, como se define en (Dieng et al, 1998). Para ello, cada departamento de la organización es considerado como un creador de conocimiento y facilita una ontología que refleja el conocimiento producido en ese departamento. De esta forma, el entorno propuesto permite la fusión de diferentes modelos ontológicos. Asimismo, se podría adaptar la terminología usada en la memoria corporativa a la terminología de cada departamento, sin tener que imponer ningún vocabulario común.

## 3.5 RESUMEN

En este capítulo se ha formalizado un entorno para integrar ontologías del dominio. Dicho entorno se compone de diferentes conjuntos de funciones, cada uno de los cuales tiene un propósito y objetivos diferentes. Por un lado, existe una necesidad de funciones aplicables a una ontología para obtener los diferentes componentes semánticos de la misma. Por otro lado, se requieren del mismo modo funciones para comparar las estructuras conceptuales contenidas en un conjunto de ontologías. Esta familia de funciones se puede dividir asimismo en varios grupos. El primer grupo permitiría comparar entidades ontológicas individuales que pertenezcan a diferentes ontologías (p.ej., conceptos o conjuntos de conceptos). El segundo grupo se construye sobre el anterior, de forma que se pueden realizar inferencias entre diferentes ontologías. Tales inferencias permiten decidir si ontologías distintas son redundantes o inconsistentes. Dos ontologías son redundantes cuando describen el dominio de la misma forma. Dos ontologías son inconsistentes cuando ambas descripciones representa puntos de vista incompatibles. Un tercer problema viene dado cuando partes de diferentes especificaciones de un dominio son equivalentes. Estamos hablando de sinonimia. El entorno presentado en este capítulo incluye mecanismos para resolver estos problemas inherentes a la cooperación.

La operación básica de este proceso es la integración de un conjunto predefinido de ontologías. En este sentido debemos aclarar que el término "integración de ontologías" debe ser entendido de manera diferente al término "inclusión de ontologías" usado en Ontolingua (Farquhar *et al*., 1997). La "inclusión de ontologías" se refiere a la reutilización de ontologías para crear nuevas, de forma que la tarea de creación de ontologías es menos costosa. El proceso de inclusión se realiza mediante la traducción del vocabulario para adaptar la terminología usada en las ontologías a la ontologías ya incluida. Esta traducción se refleja en los axiomas, de forma que aquéllos que pertenecen a una ontología pueden ser incluidos en otra. Sin embargo, creemos que la elección de la terminología más apropiada para la ontología final, que verá un usuario, es un proceso dinámico atendiendo a una serie de principios.

Para ser más preciso, esta elección depende de varios parámetros: (1) quién solicita la integración; (2) cualquier ontología presente en la ontología derivada de la integración tiene que

ser consistente con el resto de ontologías presentes en ella; (3) ninguna ontología presente en la ontología derivada de la integración puede ser redundante con respecto al resto de ontologías incluidas en ella; y (4) la cantidad de conocimiento contenida en cada ontología. Este último parámetro se evalúa siguiendo el principio de máximo contenido informativo, mientras que el segundo y tercer factor son resueltos usando los métodos propuestos para tratar los problemas inherentes a la cooperación.

El primer parámetro se puede dividir en dos partes, una dependiente del entorno y otra del usuario. La parte dependiente del entorno sólo se activa cuando el usuario no impone restricciones respecto a la terminología a utilizar, por lo que se aplicará el principio de máximo contenido informativo. La parte dependiente del usuario se encarga de decidir qué terminología usar. Actualmente, la única facilidad proporcionada por el sistema es el uso de la terminología del experto que solicitó la integración como terminología de referencia. Por lo tanto, sería deseable un incremento en las facilidades del cliente, tales como permitir al usuario seleccionar la terminología a utilizar.

En principio, se diseñó el entorno para satisfacer dos principios de integración:

(1) orientación al usuario

(2) máximo contenido informativo

El primero queda garantizado por el hecho de que el marco de integración usa una referencia semántica y terminológica para el proceso. Dicha referencia es el conocimiento suministrado por el usuario que solicita la integración. Además, el marco de integración utiliza las versiones más recientes de las ontologías creadas por los usuarios. El segundo principio queda garantizado por los diferentes criterios establecidos a lo largo del proceso.

El propio entorno para integrar ontologías puede ser dividido en diferentes fases, a saber, selección, instanciación, y transformación. El proceso de integración opera sobre un conjunto de ontologías fuente que son integradas, de forma que se produce una ontología transformada,

global. El proceso de selección se encarga de decidir qué ontologías serán finalmente integradas. Con este fin, se debe calcular para cada ontología su conjunto de ontologías compatibles, para posteriormente seleccionar en mejor subconjunto de ontologías compatibles[2]. A continuación, el proceso de instanciación armoniza la terminología utilizada por el conjunto de ontologías seleccionadas. Finalmente, se fusionan las ontologías, de forma que se genera la ontología transformada y final.

En este capítulo, el entorno de integración propuesto y formalizado ha sido comparado con diferentes entornos y sistemas para la construcción cooperativa de conocimiento, y se han comentado las principales diferencias con respecto a ellos. Además, las funciones propuestas a lo largo de este capítulo han sido ilustradas con ejemplos parciales. Sin embargo, sería deseable mostrar un ejemplo completo que incluya todos los pasos de integración. Esto será realizado en el siguiente capítulo.

[2] Se pueden aplicar diferentes criterios para este propósito

# CAPÍTULO 4

# UN EJEMPLO COMPLETO

# DE INTEGRACIÓN

En el capítulo anterior se ha presentado y formalizado un entorno para integrar ontologías. Además, se propuso una serie de ejemplos para ilustrar las definiciones incluidas en el capítulo. Sin embargo, todavía no se ha mostrado un ejemplo completo que describa exhaustivamente el funcionamiento del entorno de integración. El objetivo de este capítulo es cubrir esta necesidad. El ejemplo propuesto en este capítulo realiza la integración de un conjunto de ontologías que modelan un mismo dominio de conocimiento. Este capítulo se puede estructurar en diferentes partes. Primero se describirá el dominio de aplicación. A continuación se explicará detalladamente la secuencia completa de integración. Finalmente, habría que matizar que la intención de este capítulo no es la de validar el marco de integración sino ilustrar su funcionamiento. La validación será motivo de posterior tratamiento.

## 4.1 EL DOMINIO

El dominio escogido para ilustrar el funcionamiento del entorno de integración es una biblioteca. Este dominio puede ser considerado como bastante estable, por lo que se pueden construir ontologías que mantengan su vigencia durante bastante tiempo. Una biblioteca es un lugar donde la gente va en busca de información. Tal información está generalmente contenida en publicaciones, las cuales son editadas en diferentes formatos, entre los que podemos distinguir audio, vídeo y texto. En este dominio, los actores principales son las personas y las publicaciones. Existen distintos tipos de publicaciones y distintos tipos de personas en función del papel que desempeñan en la biblioteca. Algunas personas van en busca de información mientras que otras trabajan en la biblioteca y su función es asistir a los usuarios. Además, también existen objetos de diferente naturaleza en las bibliotecas. Por ejemplo, hay distintos tipos de equipamiento, habitaciones, etc en las bibliotecas. Todos estos elementos también deben ser incluidos en el modelo de conocimiento de una biblioteca.

Con todo esto, se construyeron cuatro ontologías en este dominio. Por motivos de simplicidad, sólo se mostrará una parte de tales ontologías. Este ejemplo está restringido a los

actores principales del dominio, es decir, personas y publicaciones. Estos cuatro modelos no se construyeron con el objetivo de construir un modelo de conocimiento completo del dominio, sino para ilustrar el modus operandi del entorno de la integración. La estructura de esta sección puede ser descrita como sigue. Primero, se describirán y mostrarán las cuatro ontologías fuente. A continuación se describirá paso a paso la integración de las cuatro ontologías.

## 4.2 ONTOLOGÍAS FUENTE

Las figuras 4.1-4.4 muestran las ontologías usadas en este ejemplo. Tales ontologías son idénticas a las usadas en el capítulo 3 para ilustrar algunos conceptos definidos allí. Las cuatro ontologías tienen un tamaño similar y han sido construidas por individuos con formación y experiencia similar. Por consiguiente, como era de esperar, las cuatro ontologías construidas tienen un grado significante de solapamiento, Sin embargo, contienen todos los elementos necesarios para ser consideradas apropiadas para este ejemplo.

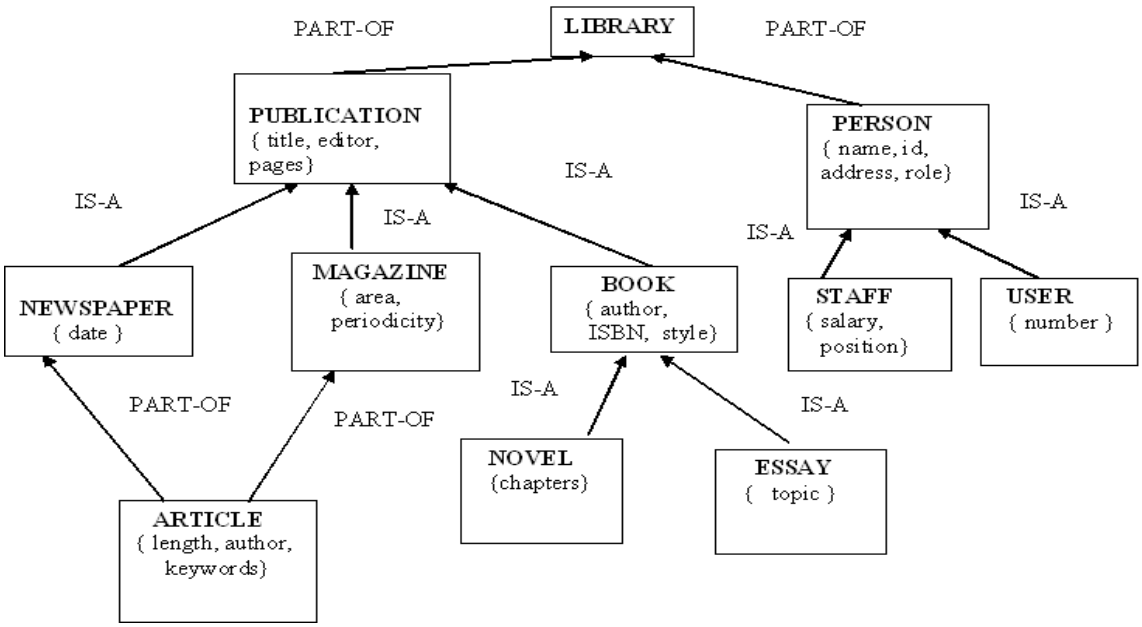


**Figura 4.1 Ontología A**

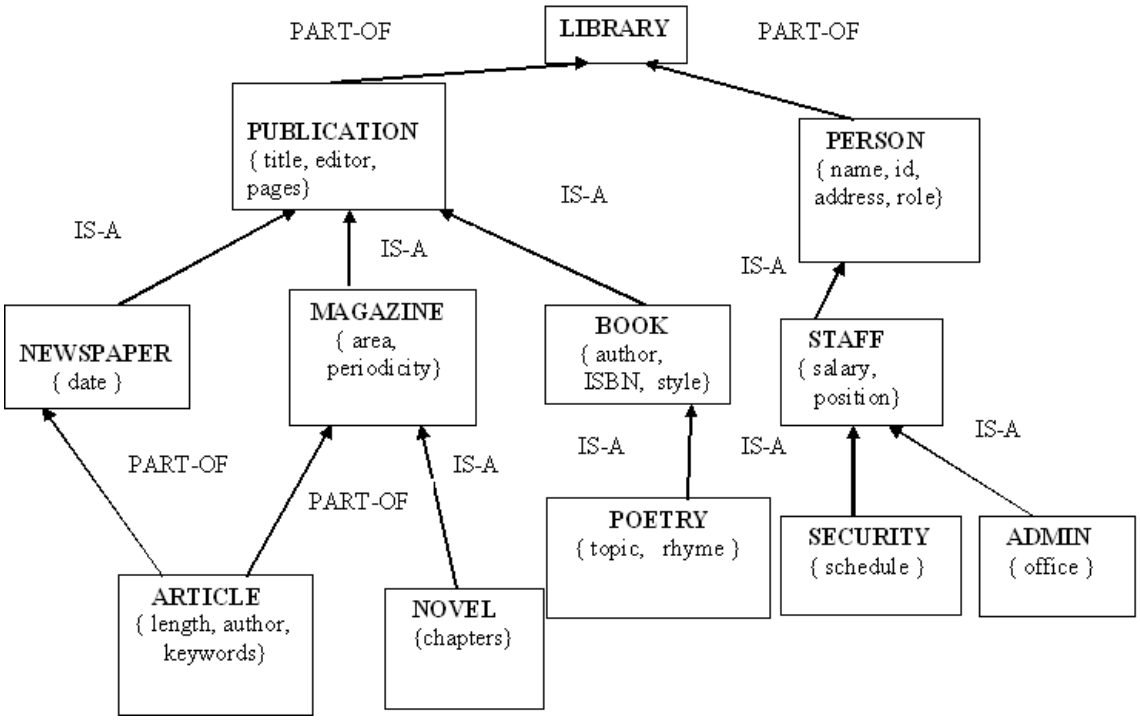


**Figura 4.2 Ontología B**

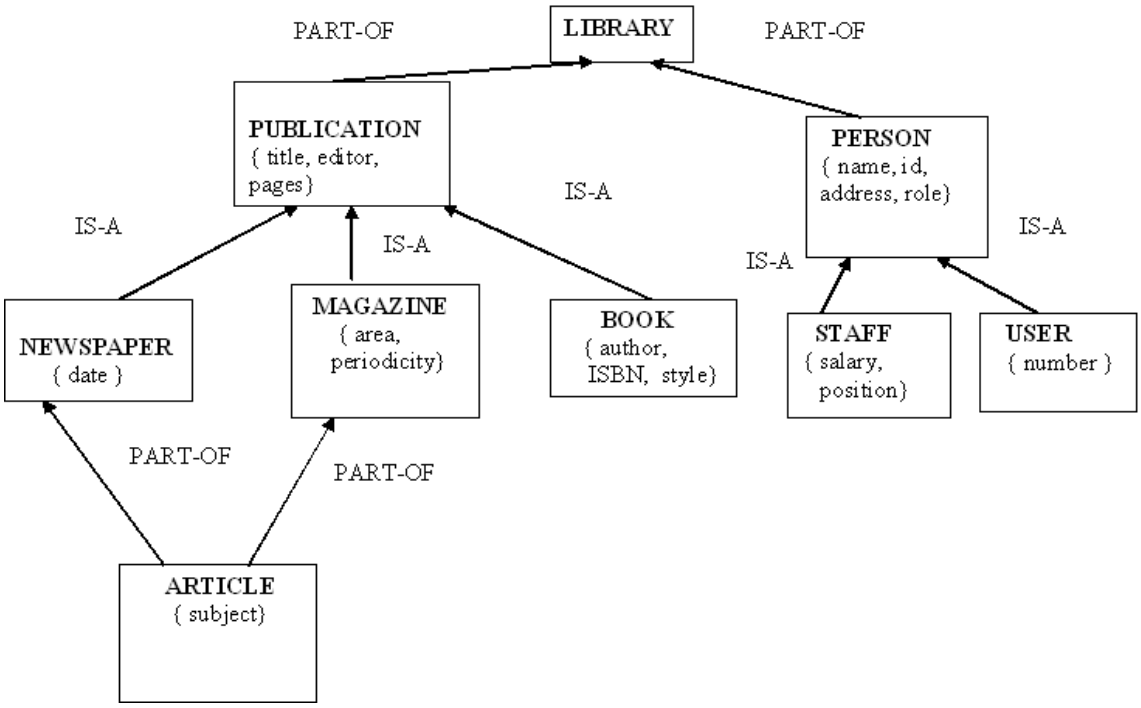


**Figura 4.3 Ontología C**

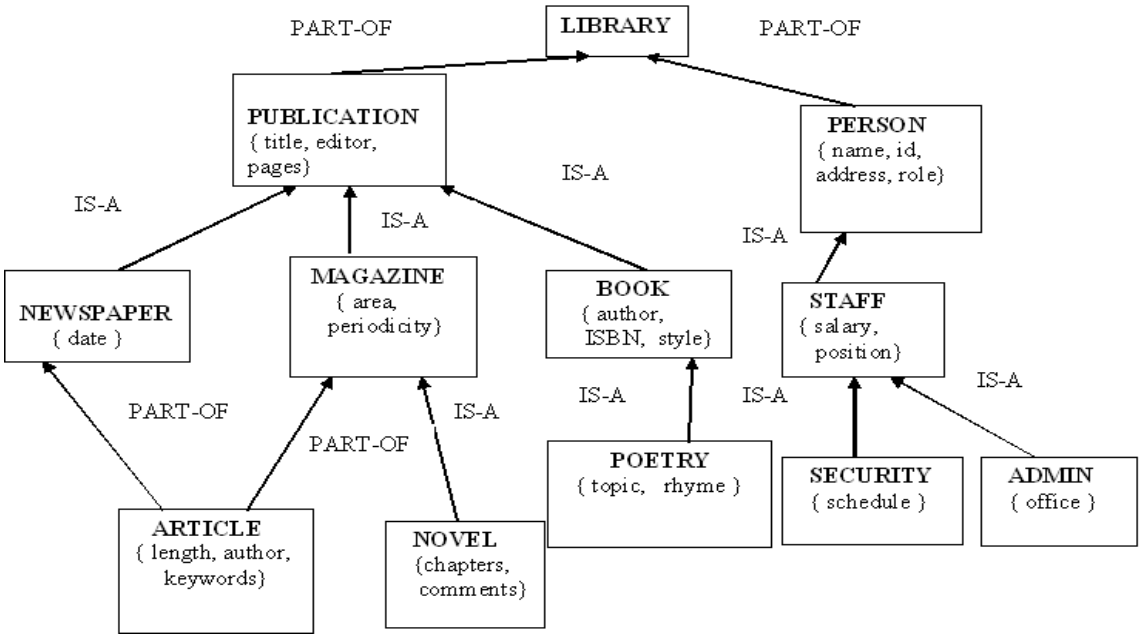


**Figura 4.4 Ontología D**

## 4.2.1 Integración

Esta sección representa la aplicación del marco de integración a las ontologías fuente mencionadas con anterioridad. De esta forma, el marco completo de integración va a ser aplicado a este conjunto de ontologías. Por consiguiente, podemos empezar por calcular el conjunto de ontologías fuente, denotado por candidates(t), como sigue:

Candidates(t)= { $O_A(t)$, $O_B(t)$, $O_C(t)$, and $O_D(t)$ }

Además, supondremos que los procesos de integración usan $O_B(t)$ como ontología referencia.

*4.2.1.1 Selección de Ontologías*

A estas alturas disponemos de un conjunto de ontologías y se necesita realizar un pre-procesamiento inicial para calcular los subconjuntos diferentes de ontologías compatibles. Para este propósito, se debe encontrar qué ontologías son incoherentes o equivalentes. En este caso, este proceso debe ser realizado cuatro veces, uno para cada ontología fuente. Por consiguiente, necesitamos aplicar aquí las definiciones 9, 10 y 11. A continuación se presentan los resultados obtenidos para cada ontología fuente $O_A(t)$, $O_B(t)$, $O_C(t)$, y $O_D(t)$.

**ONTOLOGíA $O_A(t)$**

$compatible_{OA(t)}(t)$ = { $O_A(t)$, $O_B(t)$, $O_D(t)$ } porque:

compatible_ontologies ($O_A(t)$, $O_A(t)$) = **true;** y

compatible_ontologies ($O_A(t)$, $O_B(t)$) = **true**; y

compatible_ontologies ($O_A(t)$, $O_C(t)$) = *false*; y

compatible_ontologies ($O_A(t)$, $O_D(t)$) = **true**.

$O_C(t)$ no es compatible con $O_A(t)$ porque los conceptos "$article_A$" y "$article_C$" son inconsistentes.

**ONTOLOGÍA $O_B(t)$ ;**

$compatible_{OB(t)}(t)$ = { $O_B(t)$, $O_A(t)$, $O_D(t)$ } porque:

compatible_ontologies ($O_B(t)$, $O_A(t)$) = **true**; y

compatible_ontologies ($O_B(t)$, $O_B(t)$) = **true**; y

compatible_ontologies ($O_B(t)$, $O_C(t)$) = *true*; y

compatible_ontologies ($O_B(t)$, $O_D(t)$) = **true**;

$O_C(t)$ es compatible con $O_B(t)$. Sin embargo, $O_C(t)$ no es compatible con $O_A(t)$ , que ya ha sido incluida en el conjunto de ontologías compatibles con $O_B(t)$.

**ONTOLOGÍA $O_C(t)$**

$compatible_{Oc(t)}(t)$ = { $O_C(t)$, $O_B(t)$ } porque:

compatible_ontologies ($O_C(t)$, $O_A(t)$) = *false*; y

compatible_ontologies ($O_C(t)$, $O_B(t)$) = **true**; y

compatible_ontologies ($O_C(t)$, $O_C(t)$) = **true**; y

compatible_ontologies ($O_C(t)$, $O_D(t)$) = *false*.

$O_C(t)$ no es compatible con $O_D(t)$ porque los conceptos "$article_D$" y "$article_C$" son incompatibles. El motivo de la incompatibilidad entre $O_C(t)$ y $O_A(t)$ fue mencionado anteriormente.

**ONTOLOGÍA $O_D(t)$**

$compatible_{OD(t)}(t) = \{ O_D(t), O_A(t) , O_B(t) \}$ porque:

compatible_ontologies ($O_D(t)$, $O_A(t)$) = **true**; y

compatible_ontologies ($O_D(t)$, $O_B(t)$) = **true**;

compatible_ontologies ($O_D(t)$, $O_C(t)$) = *false*; y

compatible_ontologies ($O_D(t)$, $O_D(t)$) = **true**.

La razón de incompatibilidad entre $O_D(t)$ y $O_C(t)$ fue mencionada con anterioridad.

A estas alturas se debe tomar una primera decisión, pues se debe seleccionar uno de los conjuntos recientemente calculados de ontologías compatibles para la integración. En este caso, tres de los cuatro conjuntos contienen las mismas ontologías, $O_A(t)$, $O_B(t)$, y $O_D(t)$. Para esta selección se aplicará el criterio que establece que el mejor subconjunto es el que contiene el mayor número de ontologías. Esta decisión conlleva la terminación de este paso, es decir, de la selección de las ontologías.

### *4.2.1.2 Inicialización de la Integración*

Las ontologías usadas para la integración son $O_A(t)$, $O_B(t)$, y $O_D(t)$. La siguiente operación a realizar es agregar estas tres ontologías como hijos mereológicos del concepto raíz de la ontología derivada de la integración, denotada por $O_{int}(t)$. El concepto raíz de esta ontología será un concepto llamado "library", que es el nombre del dominio. En particular, el concepto raíz de las tres ontologías fuente será un hijo mereológico de la raíz de la ontología derivada de la integración. La Figura 4.5 muestra la ontología resultante de este paso.

*4.2.1.3 Instanciación de Ontologías*

El paso siguiente es armonizar la terminología usada por las tres ontologías. En este caso, el vocabulario debe ser adaptado al de $O_B(t)$ porque esta ontología es la ontología referencia. Este proceso es descrito a continuación. Disponemos del conjunto de ontologías a integrar, compuesto por $O_A(t)$, $O_B(t)$, y $O_D(t)$, y debemos buscar conceptos sinónimos en estas ontologías. En caso de que uno de los conceptos sinónimos pertenezca a la ontología de la referencia, se adoptará el término usado en dicha ontología. En este ejemplo, el concepto $PERSON_A(t)$ es sinónimo a $PEOPLE_B(t)$, por lo que el término asociado a tal concepto en $O_A(t)$ debe ser cambiado por el término asociado al concepto sinónimo en la ontología referencia $O_B(t)$. Por consiguiente, $PERSON_A(t)$ se convertiría en $PEOPLE_A(t)$. Tal concepto incluiría como nombre alternativo su nombre anterior, es decir, PERSON. Sin embargo, esto no pasa con $PERSON_D(t)$ porque los respectivos conjuntos de conceptos hijos no son equivalentes. El segundo ejemplo de sinonimia se encuentra para los conceptos $ARTICLE_A$, $APPORTION_B$, y $ARTICLE_D$. Como en el caso anterior, el término usado en la ontología referencia será adoptado. La Figura 4.6 muestra la ontología derivada de la integración ya instanciada.

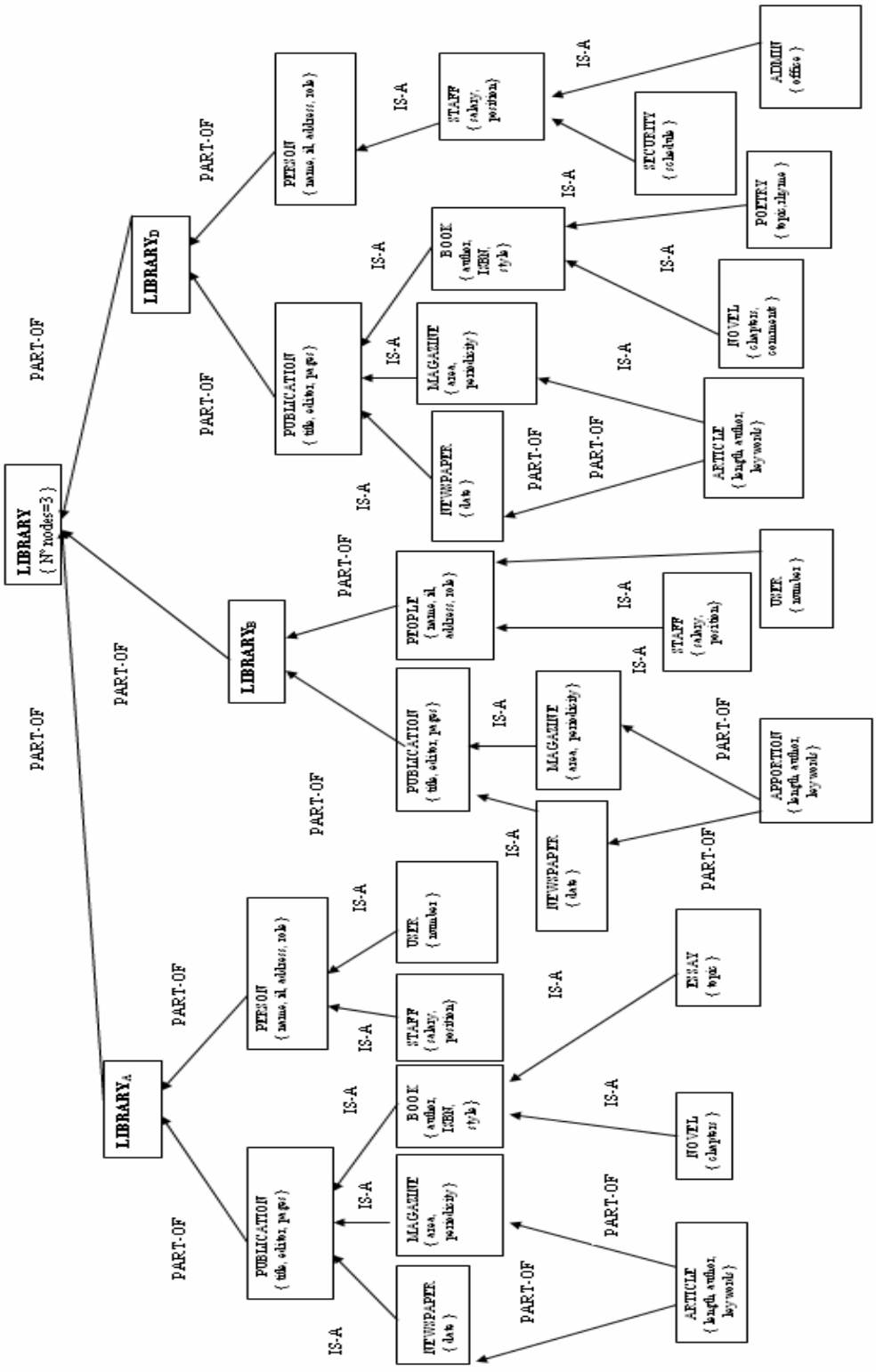


**Figura 4.5 Ontología derivada de la integración**

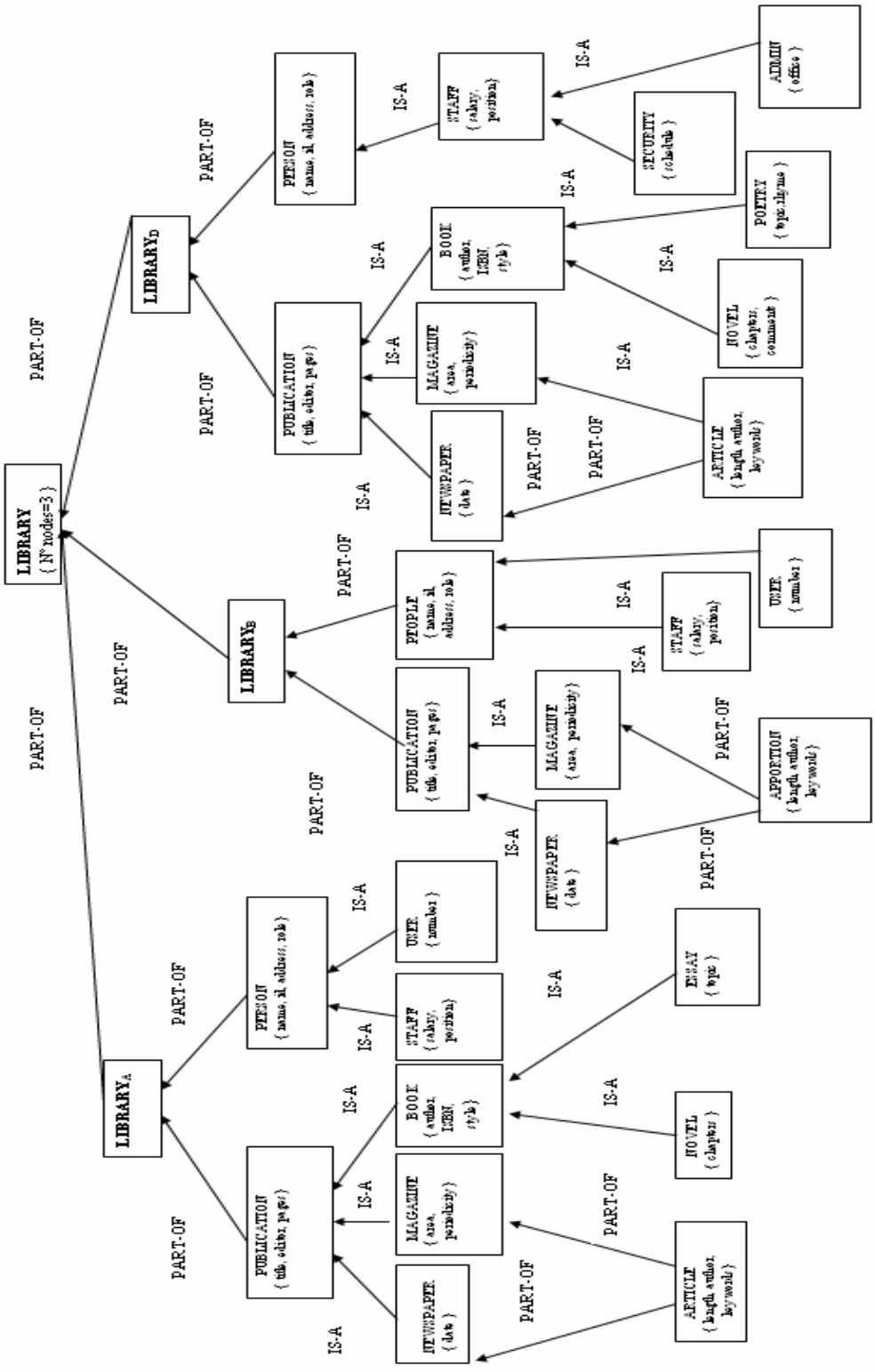


**Figura 4.6 Ontología derivada de la integración e instanciada**

*4.2.1.4 Transformación de Ontologías*

Llegados a este punto, las ontologías comparten la misma terminología, la cual se ha unificado usando la ontología referencia como la terminología estándar, y quedaría fusionar las tres ontologías para obtener la ontología global. Para tal propósito se aplicará la definición 14. Primero, se debe seleccionar una de las ontologías como el esqueleto para fusionar las restantes. Para ello, se ordenarán las tres ontologías según su número de conceptos (para cumplir el principio de máximo contenido de información). En este caso, $O_A(t)$ tiene 11 conceptos, $O_B(t)$ contiene 8, y $O_D(t)$ tiene 12 conceptos. Por consiguiente, se tomaría $O_D$ (t) como el esqueleto de proceso de fusión en caso de que no hubiera ontología referencia. Sin embargo, en este caso, al existir ontología referencia, $O_B(t)$, ésta será la seleccionada. La situación inicial puede describirse de la siguiente forma:

$PO(t)=O_B(t)$

Onto_set = { $O_B(t)$, $O_D(t)$, $O_A(t)$ }

Como la primera ontología incluida en onto_set es la seleccionada como el esqueleto para el proceso de fusión, el proceso de transformación continúa procesando la segunda ontología, en este caso, $O_D(t)$. De ahora en adelante, el proceso es iterativo, y en particular, se llevarán a cabo dos iteraciones, una por cada ontología contenida en onto_set.

ITERACIÓN 1: $O_D(t)$

Se invoca a la función SCFO (definición 18) para cada concepto contenido en $PHMRD_D(t)$.

$PHMRD_D(t)$ = {publication, newspaper, magazine, book, article, novel, poetry, person, staff, security, and admin}

El proceso comenzaría por $publication_D$. Al aplicar *SCFO(PO(t), c(t),$O_A$(t))*, el sistema busca en PO(t) conceptos sinónimos a c(t). Como resultado de esta búsqueda, no se encuentra en PO(t) ningún concepto sinónimo a c(t). Se podría pensar que $publication_{PO}$ es sinónimo a $publication_D$, pero esto no ocurre en realidad porque sus respectivos conjuntos de conceptos hijos taxonómicos no son equivalentes. Por lo tanto se ejecutará la segunda condición (ver definición 18) y se unificarán los atributos de ambos conceptos. Sin embargo, ambos conceptos tienen los mismos atributos específicos, por lo que no habrá modificación alguna. Finalmente, debido a que el único concepto padre de $publication_D$ es el concepto LIBRARY, no se establecerían nuevas relaciones.

Para c(t)=$newspaper_D$, el sistema encontraría que $newspaper_{PO}$ es un concepto sinónimo, por lo que se activaría la primera condición (ver definición 18). Para c(t) en {$newspaper_D$ , $magazine_D$, $article_D$} el proceso sería similar y la ontología permanecería inalterada. Sin embargo, la situación es diferente para c(t) en {$book_D$, $novel_D$, $poetry_D$, $person_D$, $staff_D$ , $security_D$, and $admin_D$}. Veamos qué sucede en este caso.

*SCFO(PO(t), book, $O_D$(t))*: no hay ningún concepto sinónimo ni con el mismo nombre que "book", por lo que se invoca a AFO.

*AFO(PO(t), book, $O_D$(t))*: se genera un nuevo concepto en PO(t), cuyo nombre es “book" y su conjunto de atributos específicos es {author, ISBN, style}. De la misma forma que antes, se lanza PUO, y PO(t) es modificada por PUO(PO(t), $publication_{po}$, $book_D$, $O_D$(t)), porque "book" es hijo taxonómico de "publication" en $O_D$, y $publication_{po}$ es una versión actualizada de $publication_D$. Por lo tanto, el nuevo concepto $book_{po}$ debe ser enlazado a $publication_{po}$ como hijo taxonómico. La Figura 4.7 muestra PO(t) tras realizar esta operación.

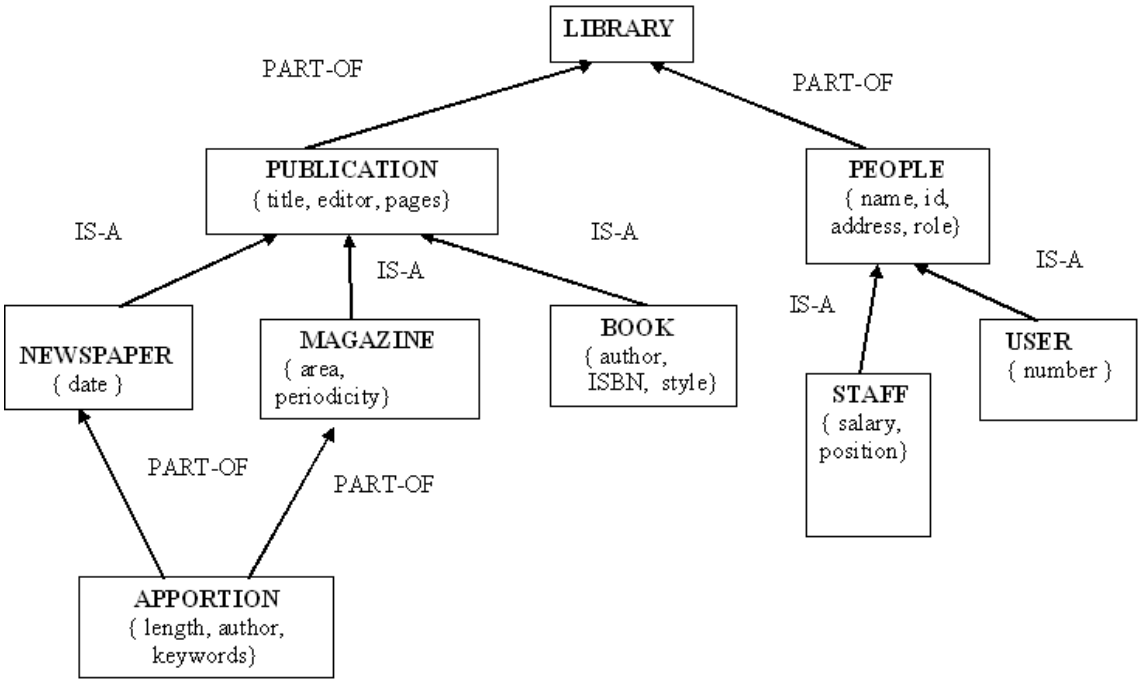


**Figura 4.7 PO(t) tras incluir $Book_D$**

*SCFO(PO(t), novel, $O_D(t)$)*: no existe ningún concepto sinónimo ni con el mismo nombre que el concepto "novel" por lo que se activará de nuevo AFO.

*AFO(PO(t), novel, $O_D(t)$)*: se genera un nuevo concepto en PO(t), cuyo nombre es "novel" y su conjunto de atributos específicos es {chapters, comments}. Como sucedió anteriormente, se hace una llamada a PUO. PO(t) es modificada por PUO(PO(t), $magazine_{po}$, $novel_D$, $O_D(t)$), porque "novel" es hijo taxonómico de "magazine" en $O_D$, y $magazine_{po}$ es una versión actualizada de $magazine_D$. Por lo tanto, el nuevo concepto $novel_{po}$ debe ser enlazado a $magazine_{po}$ como hijo taxonómico. La Figura 4.8 muestra PO(t) tras realizar esta operación.

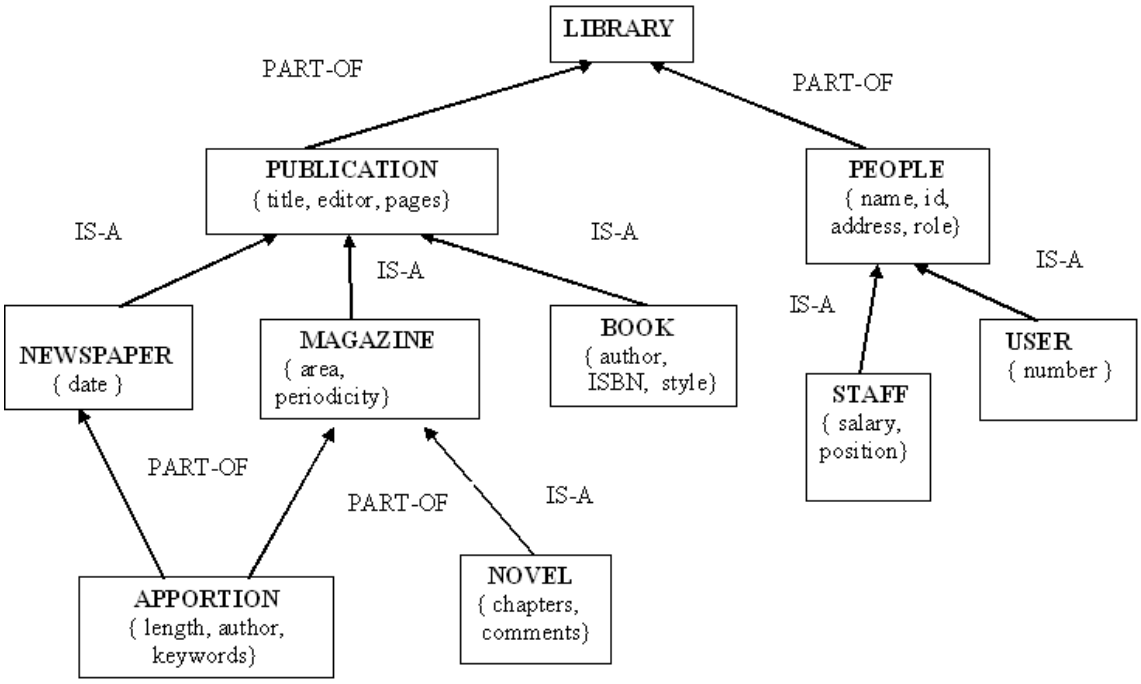


**Figura 4.8 PO(t) tras incluir $Novel_D$**

*SCFO(PO(t), poetry, $O_D(t)$)*: tampoco existe ningún concepto sinónimo ni con el mismo nombre que "poetry", por lo que debemos recurrir de nuevo a AFO.

*AFO(PO(t), poetry, $O_D(t)$)*: un nuevo concepto se genera en PO(t), cuyo nombre es "poetry" y su conjunto de atributos específicos es {topic, rhyme}. En este caso también debemos ejecutar PUO, y PO(t) es modificado por PUO(PO(t), $book_{po}$, $poetry_D$, $O_D(t)$), porque "poetry" es hijo taxonómico de "book" en $O_D$, y $book_{po}$ es una versión actualizada de $book_D$. Por lo tanto, el nuevo concepto $poetry_{po}$ debe ser enlazado a $book_{po}$ como hijo taxonómico. La Figura 4.9 muestra PO(t) tras realizar esta operación.

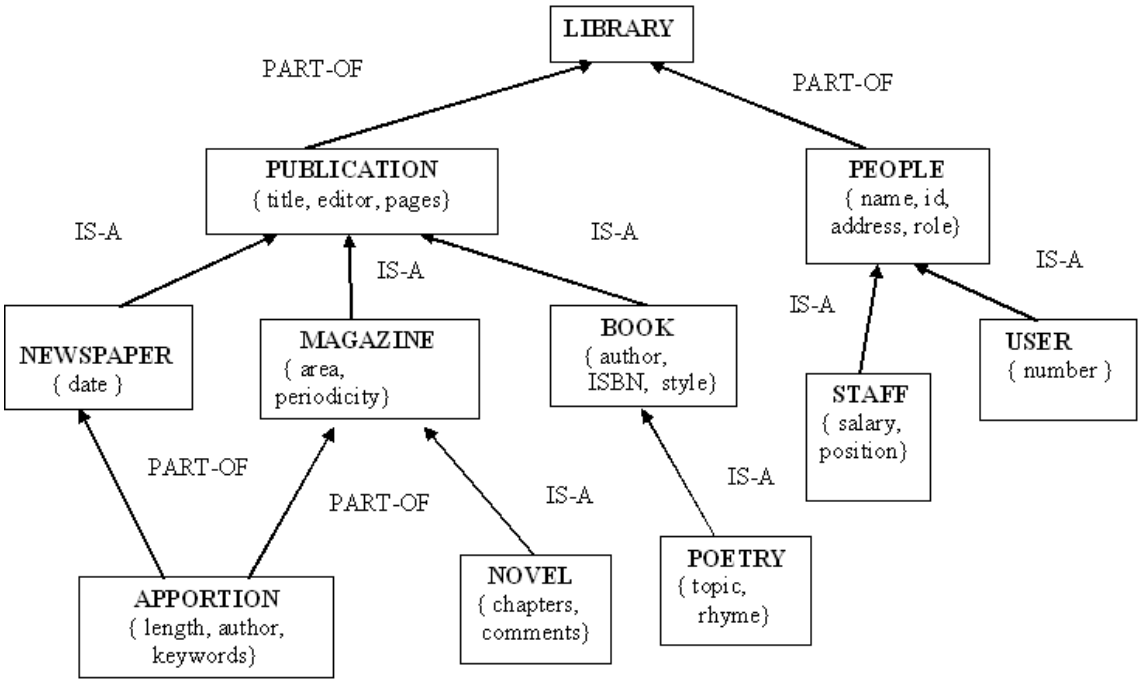


**Figura 4.9 PO(t) tras incluir $Poetry_D$**

*SCFO(PO(t), person, $O_D(t)$)*: no existe ningún concepto sinónimo, pero hay un concepto (people) con el mismo término como nombre alternativo, por lo que la segunda condición (ver definición 18) se activa y se unifican ambos conceptos. En este caso, los atributos y conceptos padre de $person_D$ ya aparecen en $people_{PO}$ por lo que PO(t) no cambia.

*SCFO(PO(t), staff, $O_D(t)$)*: no hay concepto sinónimo a "staff", pero sí un concepto con el mismo nombre, por lo que se activa la segunda condición de la definición 18 y se unifican ambos conceptos. En este caso, los atributos y conceptos padres de $staff_D$ ya aparecen en $staff_{PO}$, por lo que PO(t) no cambia.

*SCFO(PO(t), security, $O_D(t)$)*: no hay ningún concepto sinónimo a "security" ni ninguno con el mismo nombre, por lo que AFO es invocado.

*AFO(PO(t), security, $O_D(t)$)*: se crea un nuevo concepto en PO(t), cuyo nombre es "security" y su conjunto de atributos específicos es {schedule}. Se invoca de nuevo PUO, y PO(t) es modificado por PUO(PO(t), $staff_{po}$, $security_D$, $O_D(t)$), porque "security" es hijo taxonómico de "staff" en $O_D$, y $staff_{po}$ es una versión actualizada de $staff_D$. Por lo tanto, el nuevo concepto $security_{po}$ debe ser enlazado a $staff_{po}$ como su hijo taxonómico. La Figura 4.10 muestra PO(t) tras realizar esta operación.

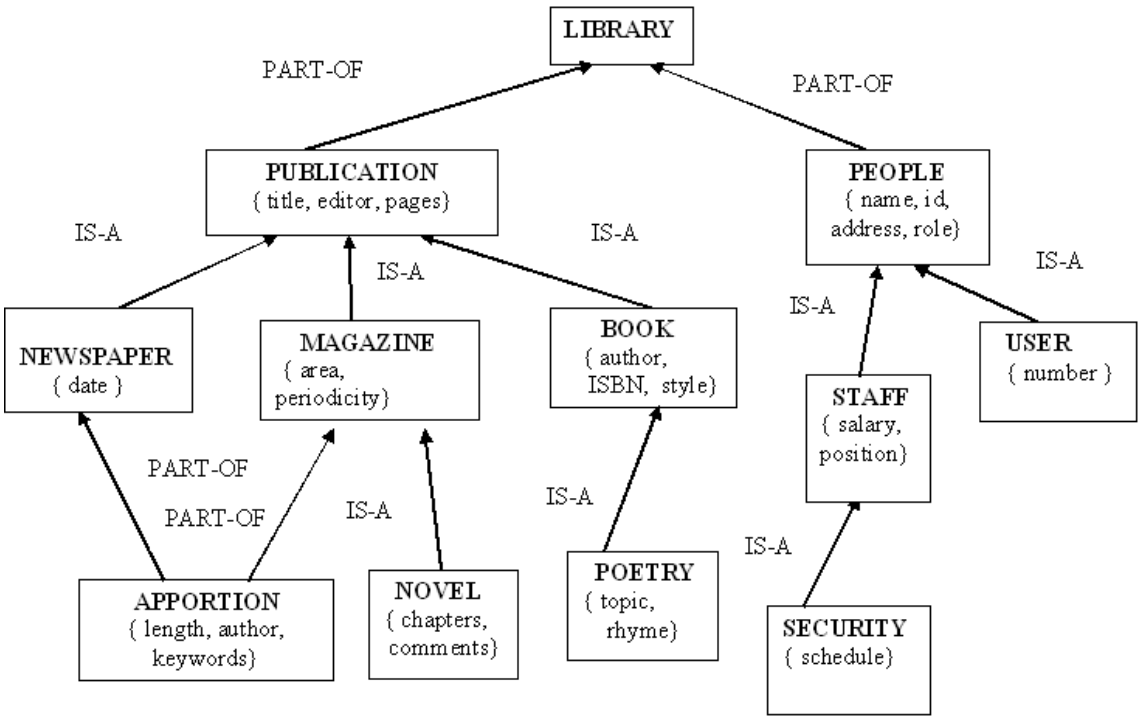


**Figura 4.10 PO(t) tras incluir $Security_D$**

*SCFO(PO(t), admin, $O_D(t)$)*: no hay ningún concepto sinónimo a "admin" ni ninguno con el mismo nombre, por lo que AFO es invocado.

*AFO(PO(t), admin,, $O_D(t)$)*: se crea un nuevo concepto en PO(t), cuyo nombre es "admin" y su conjunto de atributos es {office}. Nuevamente es necesario ejecutar PUO, y PO(t) es modificado por PUO(PO(t), $staff_{po}$, $admin_D$, $O_D(t)$), puesto que "admin" es hijo taxonómico de "staff" en $O_D$, y $staff_{po}$ es una versión actualizada de $staff_D$. Por lo tanto, el nuevo concepto

$admin_{po}$ debe ser enlazado a $staff_{po}$ como su hijo taxonómico. El resultado de este proceso es mostrado en la Figura 4.11.

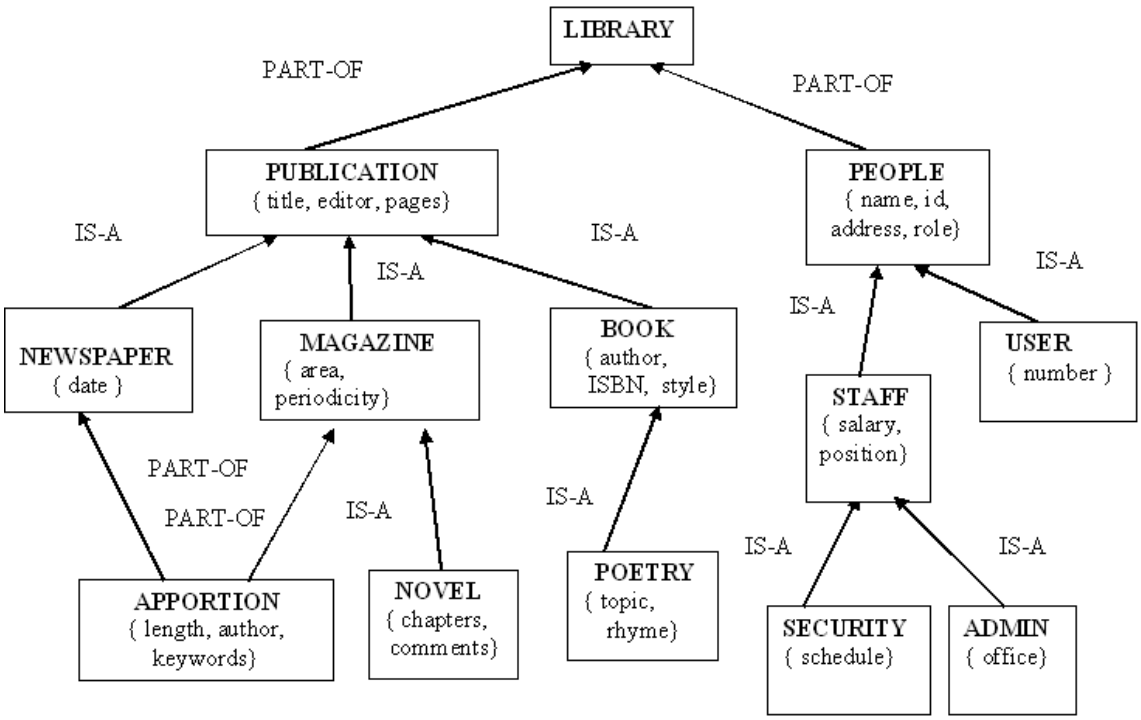


**Figura 4.11 PO(t) al incluir $Admin_D$**

En este momento, todos los conceptos que pertenecen a $O_D(t)$ han sido procesados e insertados en la ontología parcial final por lo que podemos decir que ha acabado la primera iteración. Restaría por ejecutar la segunda iteración.

ITERACION 2: $O_A(t)$

La función SCFO (definición 18) se ejecutaría para cada concepto en $PHMRD_A(t)$.

$PHMRD_A(t)$ = {publication, newspaper, magazine, book, article, novel, essay, person, staff, user}

Veamos cómo se aplica SCFO sobre cada concepto.

c(t) es $publication_A$.

Al aplicar *SCFO(PO(t), c(t),$O_A$(t)*, el sistema busca en PO(t) conceptos sinónimos a c(t). En este caso se encontraría el concepto $publication_{PO}$ como sinónimo. Por lo tanto, se activaría la primera condición de la definición 18 y la ontología no sufriría modificación alguna.

Para c(t) en {$newspaper_A$, $magazine_A$, $article_A$, $book_A$, $person_A$, $staff_A$, $user_A$}, el proceso sería similar y la ontología permanecería siendo la misma. Sin embargo, no existen conceptos sinónimos para $novel_A$ y $essay_A$. Veamos qué ocurre en estos dos casos.

*SCFO(PO(t), novel, $O_A$(t))*: no hay ningún concepto sinónimo ni con el mismo nombre que "novel", por lo que se activaría la segunda condición de la definición 18. Por lo tanto se unificarían los atributos y relaciones de ambos. El resultado de este proceso se muestra en la Figura 4.12.

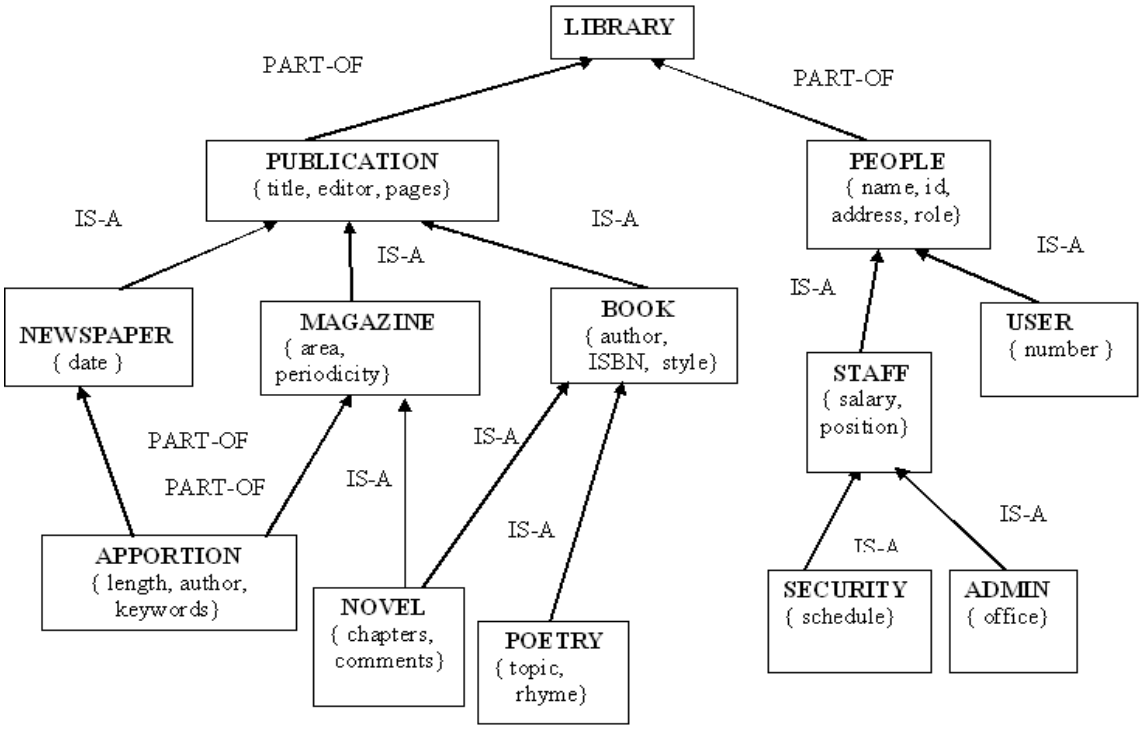


**Figura 4.12 PO(t) tras incluir novel$_A$**

*SCFO(PO(t), essay, $O_A$(t))*: no se encuentra ningún concepto sinónimo ni ningún concepto con el nombre "essay" por lo que se vuelve a invocar a la función AFO.

*AFO(PO(t), essay, $O_A$(t))*: se genera un nuevo concepto en PO(t), cuyo nombres es "essay" y su conjunto de atributos específicos es {topic}. PO(t) sería modificado por PUO(PO(t), book$_{po}$, essay$_A$, $O_A$(t)), al ser "essay" un hijo taxonómjco de "book" en $O_A$, y book$_{po}$ ser una versión actualizada de book$_A$. Por lo tanto, el nuevo concepto essay$_{po}$ debe enlazarse a book$_{po}$ como su hijo taxonómico. En la Figura 4.13 se muestra PO(t) tras esta modificación.

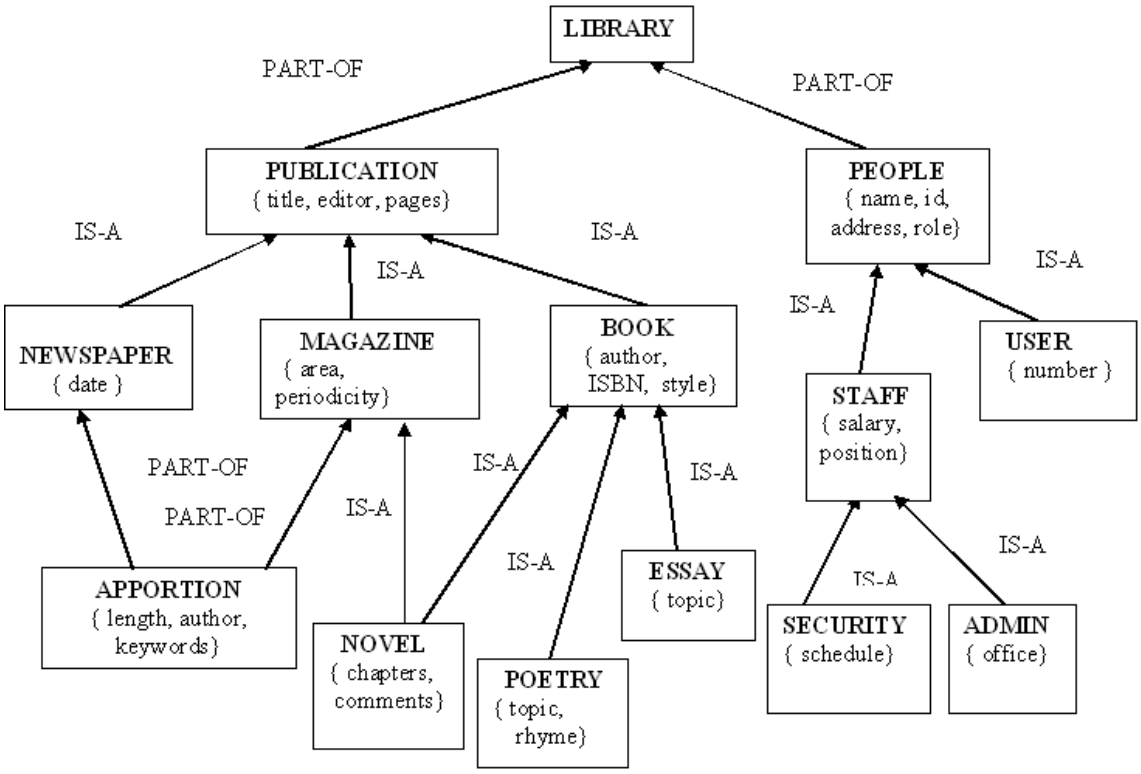


**Figura 4.13 Ontología derivada de la integración, transformada y final**

Después de procesar este concepto, todos los conceptos en $O_A(t)$ han sido procesados, por lo que la segunda iteración del algoritmo de transformación ha concluido. Como no quedan más ontologías para procesar, se puede decir que el proceso de la integración ha concluido. Por consiguiente, la Figura 4.13 también es la ontología final construida al integrar $O_A(t)$, $O_B(t)$, y $O_D(t)$.

## 4.3 DIFERENTES PUNTOS DE VISTA PARA EL EJEMPLO

En el apartado anterior se ha presentado el ejemplo desde el punto de vista definido por $O_B(t)$. Ahora, el proceso de integración será repetido usando las otras ontologías como referencia del proceso. Sin embargo, sólo los resultados finales del proceso de integración van a ser mostrados.

### 4.3.1 Usando $O_A(t)$ y $O_D(t)$ como Referencia

Al usar $O_A(t)$ u $O_D(t)$ como ontología de referencia se obtendrían los mismos resultados ya que comparten la misma terminología y sus conjuntos de ontologías compatibles son iguales. La única diferencia con respecto al resultado obtenido para la integración realizada usando $O_B(t)$ como referencia sería el nombre de los conceptos PEOPLE y APPORTION, que cambian a llamarse PERSON Y ARTICLE respectivamente, quedando los anteriores como nombres alternativos. La Figura 4.14 muestra el resultado de dichos procesos de integración.

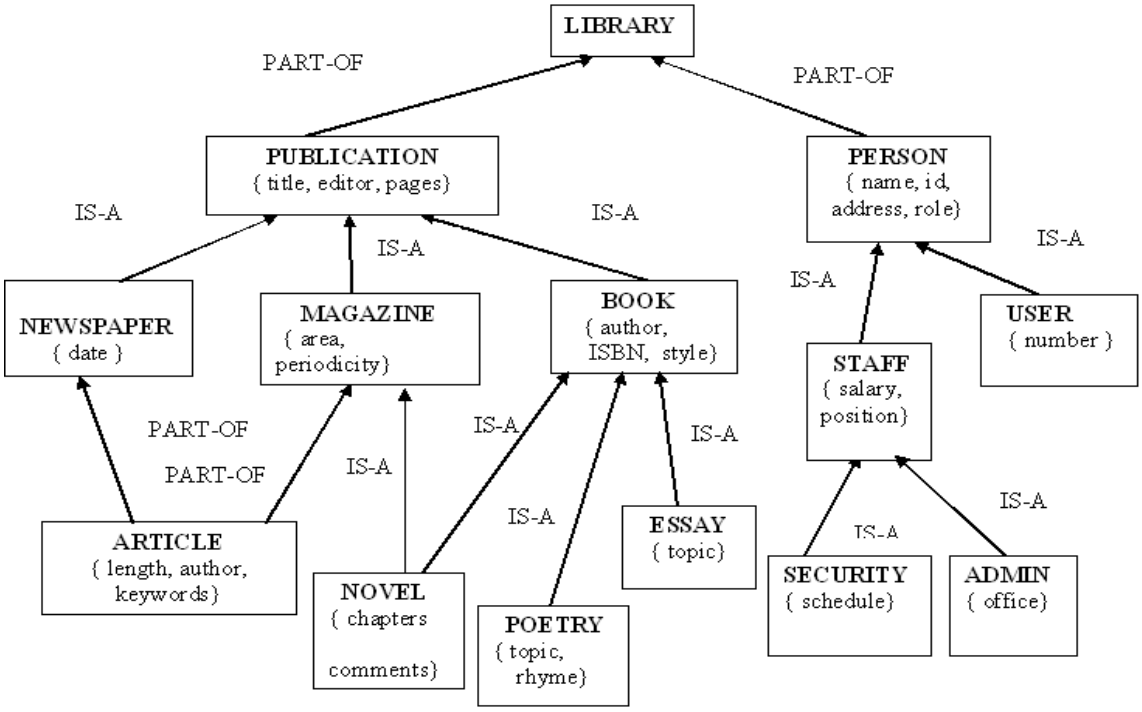


**Figura 4.14 Ontología global, final usando $O_A(t)$ y $O_D(t)$ como referencia**

## 4.3.2 Usando $O_C(t)$ como Referencia

En este caso, las ontologías que toman la parte en el proceso de integración serían $O_B(t)$ y $O_C(t)$. En este caso, $PEOPLE_B$ y $PERSON_C$ serían identificados como sinónimos y se usará el término PERSON en el concepto integrado porque $O_C(t)$ es la ontología referencia. El resultado final de tal proceso de la integración se muestra en la Figura 4.15.

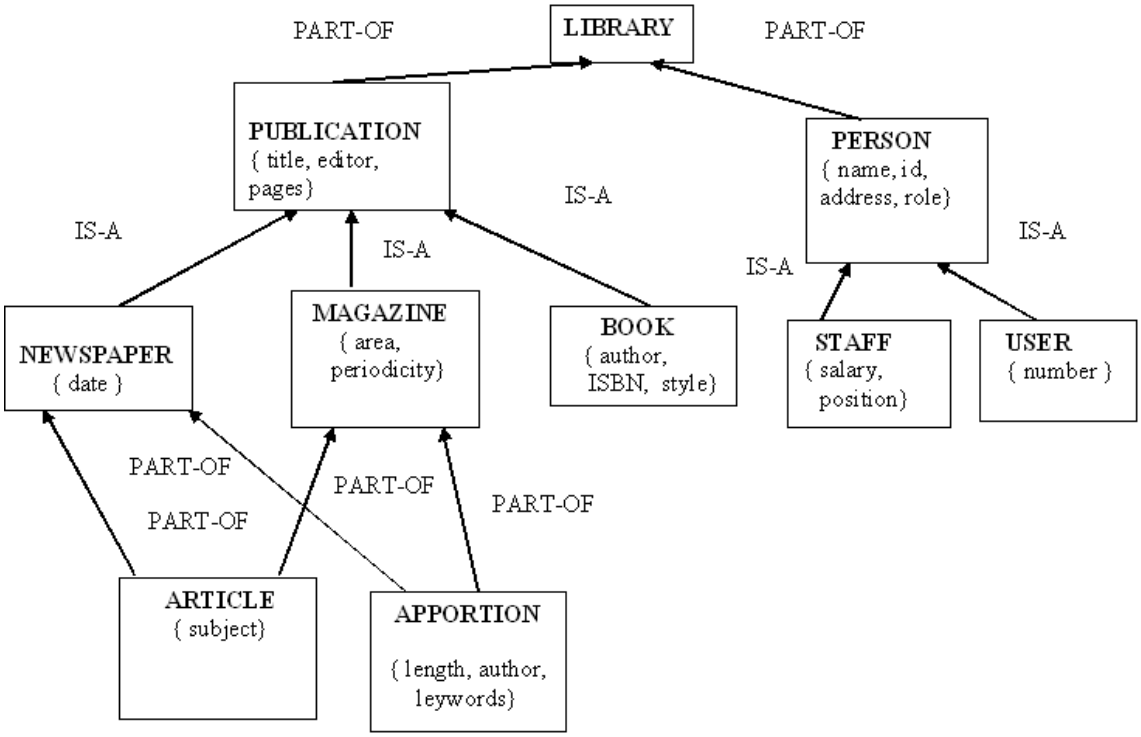


**Figura 4.15 Ontología final, global usando $O_C(t)$ como referencia**

## 4.4 RESUMEN

En este capítulo se ha mostrado un ejemplo práctico detallado que describe el modus operandi del entorno de integración. En tal ejemplo se muestran todas las cuestiones importantes de la integración, como el descubrimiento de conceptos sinónimos o de ontologías inconsistentes. Es más, el ejemplo constituye una aplicación detallada del procedimiento para seleccionar las ontologías a integrar. En el ejemplo se integraron cuatro ontologías. Tres de ellas eran compatibles para la integración, descartando la cuarta debido a la inconsistencia con respecto al resto de ontologías. Se han realizado cuatro procesos de la integración en este ejemplo, cada uno con una ontología diferente como referencia. Así, los resultados obtenidos permiten observar parte de las capacidades de personalización del entorno. Analizando los resultados de los distintos procesos de integración llevados a cabo en este capítulo, puede verse que el entorno realiza la integración de puntos de vista consistentes. En particular, un punto de vista puede ser al mismo tiempo consistente con respecto a dos puntos de vista incoherentes entre sí. Esto ocurre aquí con $O_B(t)$, que es consistente con respecto a $O_A(t)$ y $O_C(t)$, que son inconsistentes entre sí. Por consiguiente, estos procesos de integración guiados por referencia quedan influenciados por la subjetividad del conocimiento de los constructores de las ontologías privadas, en particular, por la referencia. En caso de que no se proporcione ninguna ontología de referencia, el proceso se guiará por un razonamiento democrático, es decir, el conocimiento especificado por más participantes de la integración será el más importante.

A través de este capítulo, se ha ilustrado el entorno de integración formalizado en el capítulo 3 con un ejemplo completo. Sin embargo, la validación de dicho entorno no ha sido realizada. El próximo capítulo se encargará de dicha validación.

# CAPÍTULO 5

# VALIDACIÓN DEL ENTORNO DE INTEGRACIÓN

En los capítulos anteriores se han especificado un modelo ontológico formal y un entorno formal para la integración de ontologías. En este capítulo nos proponemos validar el entorno ontológico global. En particular, este proceso de validación tiene dos apartados. Por un lado se mostrará el proceso de integración. Para ello, se analizarán y compararán los elementos ontológicos contenidos en ontologías individuales con los contenidos en ontologías obtenidas a través de procesos de integración. Por otro lado, se validará la calidad de las ontologías construidas mediante procesos de integración. Con esta finalidad se utilizarán algunos principios útiles para la construcción de "buenas" ontologías. Estos dos estudios constituirán una validación interna del sistema. Sin embargo, quedaría por demostrar la pertinencia del entorno para ayudar a resolver problemas. Esto se ilustrará mediante la aplicación del entorno para resolver distintos problemas. Se describirán las aplicaciones más importantes y se analizará el papel de la metodología de integración en ellas.

## 5.1 VALIDACIÓN DEL ENTORNO DE INTEGRACIÓN

El marco usado para validar el entorno de integración se basa en la medición del conocimiento ganado por los usuarios. La ganancia es medida mediante comparaciones de la ontología del usuario con la ontología transformada y derivada de la integración, y enumerando el número de conceptos, relaciones y atributos que aparecen en la ontología final que no aparecían en la ontología del usuario. Por consiguiente, se necesita un núcleo matemático para medir tal ganancia de conocimiento. En la próxima sección se propone un método para medir dichas ganancias de conocimiento.

### 5.1.1 Un Método para Medir Ganancias de Conocimiento

Esta sección explica cómo vamos a medir el conocimiento adquirido por los usuarios. Para ello se calculará una ganancia media ponderada en las tres categorías ontológicas usadas para realizar esta evaluación, a saber, conceptos, atributos y relaciones. El mismo procedimiento

ha sido utilizado para todas las categorías de conocimiento. La fórmula para dicho cálculo se basa en los siguientes elementos:

- Valor de una categoría de conocimiento en cada ontología de los expertos.

- Valor de una categoría de conocimiento en la ontología transformada, derivada de la integración.

Introduzcamos primero la notación utilizada en las siguientes fórmulas.

- **n** es el número de ontologías que toman parte en el proceso de integración.

- $|\mathbf{X_i}|$ representa el número de entidades de conocimiento (p.ej., conceptos (C), atributos (A) o relaciones (R)) en la ontología i. Así, el número de conceptos en la ontología i es denotado por $|C_i|$, el número de atributos por $|A_i|$, y el número de relaciones por $|R_i|$. Por lo tanto, $X \in \{A,C,R\}$, con i=1..n.

**La ganancia local ($\Delta_i$)** representa el conocimiento ganado por el experto i al usar nuestro entorno de integración. Esto es, el porcentaje de conocimiento incluido en la ontología transformada, derivada de la integración que no aparece en la ontología i. Se puede considerar la ganancia con respecto a la ontología i; $X \in \{A,C,R\}$; i=1..n, y $X_{int}$ representa la categoría de conocimiento correspondiente en la ontología transformada, derivada de la integración.

Para ello, la ganancia media ponderada se calcula usando la fórmula siguiente:

$$gain = \frac{\sum_{1}^{n}(|X_i| \times \Delta_i)}{\sum_{1}^{n}|X_i|} \qquad \Delta_i = \frac{(|X_{\text{int}}| - |X_i|)}{|X_i|}$$

Finalmente, una ganancia global de conocimiento puede ser asignada a cada proceso de integración de conocimiento. Para ello, se calculan los valores para cada categoría parcial de la siguiente forma:

$$global\_gain = \frac{|C_{\text{int}}| * gain_C + |R_{\text{int}}| * gain_R + |A_{\text{int}}| * gain_A}{|C_{\text{int}}| + |R_{\text{int}}| + |A_{\text{int}}|}$$

donde

$gain_i$ representa la ganancia de conocimiento para conceptos (C), relaciones (R), y atributos (A) respectivamente, i en {C,R,A}.

$|X_{int}|$ representa el número de conceptos, relaciones y atributos en la ontología final integrada, con X en {C,R,A}.

Expliquemos las razones para usar estas fórmulas. La ganancia local calcula el porcentaje de conocimiento añadido por la ontología transformada, derivada de la integración, lo cual es una manera apropiada para obtener resultados relativos. Se declaró anteriormente que nos interesa obtener valores relativos en lugar de absolutos debido a que ontologías que corresponden a dominios diferentes poseen tamaños diferentes. De esta forma, el cómputo de la ganancia de conocimiento absoluta produciría, en nuestra opinión, resultados no muy fiables.

La ganancia de conocimiento se obtiene para las tres categorías de conocimiento mencionadas calculando la media ponderada de las ganancias locales. Las ganancias locales se ponderan para considerar la cantidad diferente de conocimiento contenida en cada ontología. Este aspecto debe tenerse en cuenta para calcular los resultados de un dominio específico con la finalidad de obtener resultados realistas. De hecho, lo que estamos midiendo y comparando, de alguna forma, es la cantidad de axiomas estructurales contenidos en las ontologías fuente y la ontología final, pero ponderando las distintas categorías de conocimiento.

## 5.1.2 Ejemplo

Presentemos un ejemplo. Supongamos que tenemos dos ontologías, llamadas $O_1$ y $O_2$. $O_1$ tiene 108 conceptos, 120 relaciones y 176 atributos. Por otra parte, $O_2$ tiene 65 conceptos, 86 relaciones y 111 atributos. Al integrar dichas ontologías se obtiene una nueva ontología que contiene 123 conceptos, 156 relaciones, y 215 atributos. La Tabla 5.1 muestra la descripción de tanto las ontologías fuente como la integrada.

| Ontología | Conceptos | Relaciones | Atributos |
|---|---|---|---|
| $O_1$ | 108 | 120 | 176 |
| $O_2$ | 65 | 86 | 111 |
| Integrada | 123 | 156 | 215 |

**Tabla 5.1 Descripción de las ontologías**

Ahora podemos proceder a calcular la ganancia de conocimiento para cada categoría:

**CONCEPTOS**

Calculemos la ganancia local para $O_1$ y $O_2$, denotadas por $\Delta_1$ y $\Delta_2$ respectivamente:

$$\Delta_1 = \frac{135-108}{108} * 100 = 25\% \qquad \Delta_2 = \frac{135-65}{65} * 100 = 107\%$$

La ganancia de conocimiento para los conceptos se puede calcular de la forma siguiente:

$$gain_c = \frac{108 * 25 + 65 * 107}{108 + 65} = 55.8\%$$

Una media no ponderada hubiera producido un resultado del 66% de ganancia, pero este valor no sería muy realista porque influirían más las ontologías pequeñas que las grandes.

**RELACIONES**

Calculemos ahora la ganancia local de relaciones para $O_1$ y $O_2$, denotadas por $\Delta_1$ y $\Delta_2$ respectivamente:

$$\Delta_1 = \frac{156-120}{156} * 100 = 23\% \qquad \Delta_2 = \frac{156-86}{156} * 100 = 45\%$$

Podemos calcular la ganancia de conocimiento de relaciones de la siguiente forma:

$$gain_R = \frac{120 * 23 + 86 * 45}{120 + 86} = 32\%$$

**ATRIBUTOS**

Calculemos la ganancia local de atributos para $O_1$ y $O_2$, esto es, $\Delta_1$ y $\Delta_2$ respectivamente:

$$\Delta_1 = \frac{215-176}{215} * 100 = 18\% \qquad \Delta_2 = \frac{215-111}{215} * 100 = 48\%$$

Podemos calcular la ganancia de conocimiento para atributos de la siguiente forma:

$$gain_A = \frac{176 * 18 + 111 * 48}{176 + 111} = 29\%$$

Finalmente, la ganancia global del proceso de integración se calcularía como sigue:

$$global\_gain = \frac{123 * 56 + 156 * 32 + 215 * 29}{123 + 156 + 215} = 36\%$$

## 5.1.3 Experimentos de Integración

El entorno de integración ha sido evaluado por individuos con diferente experiencia, pero principalmente por estudiantes de pre y postgrado. El sistema ha sido usado durante un año por estudiantes de último curso y por estudiantes de doctorado de diferentes disciplinas. El rango de dominios en los cuales se ha aplicado varía desde tecnologías de la información (p.ej., videoconferencia) hasta medicina (p.ej., Leucemia).

### *5.1.3.1 Experimento 1*

El trabajo consistía en construir diferentes ontologías en distintos dominios que fueron asignados a los estudiantes. Dicho conocimiento fue adquirido mediante entrevistas con expertos. A cada estudiante se le asignó un dominio, por lo que debía entrevistar a un experto en dicho dominio y construir la correspondiente ontología. Por lo tanto, cada estudiante actuó como ingeniero de conocimiento. Una vez construidas las ontologías, los estudiantes que tenían asignados el mismo tema tenían que usar el entorno de integración para integrar los resultados obtenidos individualmente para comprobar la utilidad de los procesos de integración. De alguna forma, cuando los estudiantes usaban el sistema actuaban como casi-expertos, puesto que sus ontologías representaban conocimiento adquirido de expertos.

Cada dominio requería una diferente carga de trabajo para generar la ontología, de forma que el tamaño de las ontologías obtenidas por cada estudiante dependía del dominio en cuestión. Esto hizo que el tamaño de las ontologías no fuera homogéneo entre dominios. Por ello, se pensó que los resultados debían ser analizados y comparados en términos relativos y no absolutos. Así, los valores mostrados como resultado de la evolución de la metodología de integración son porcentuales. Se han seleccionado tres dominios para mostrar sus resultados: a) videoconferencia sobre redes multicast; b) circuitos integrados; y c) clima. A continuación procederemos a introducir dichos dominios.

- Videoconferencia sobre redes multicast: Este dominio cubre la transmisión de información para videoconferencia usando redes IP multicast en vez de las tradicionales redes IP unicast. Dos estudiantes se encargaron de entrevistar expertos en este dominio y construir las correspondientes ontologías.

- Circuitos integrados: Tres estudiantes fueron encargados de adquirir conocimiento de expertos sobre el diseño y la implementación de circuitos integrados. Dichos expertos proporcionaron información sobre las características de circuitos integrados, la tecnología usada, etc.

- Clima: Este dominio fue asignado a dos estudiantes, que estudiaron los distintos elementos que determinan el clima de una región, así como los diferentes fenómenos atmosféricos como tormentas, lluvias tropicales, etc. Como resultado se obtuvieron dos ontologías que se integraron usando nuestro entorno de integración.

Se ha aplicado la metodología para medir la ganancia de conocimiento a estos tres dominios, y los resultados se presentan en las Tablas 5.2-5.3. Como se puede apreciar, el uso del entorno de integración es beneficioso para los usuarios porque la ontología transformada, derivada de la integración tiene más cantidad de conocimiento que las distintas ontologías individuales. Una alta ganancia de conocimiento indica que diferentes expertos han especificado diferentes partes del dominio, mientras que una baja ganancia indica que los expertos han descrito el dominio de forma similar. Podemos extraer algunas conclusiones de estos resultados. La mayor tasa de acuerdo entre los expertos, y por lo tanto, la menor ganancia de conocimiento ocurre en la categoría de los conceptos. Esto se debe a diferentes razones. Por un lado, no es sorprendente que diferentes expertos encuentren similares entidades conceptuales en un dominio dado, aunque, por otro lado, el mecanismo de detección de sinónimos puede ayudar en esta tarea.

En nuestros resultados, la mayor ganancia de conocimiento aparece en la categoría de atributos. Esta disensión puede ser facilitada por el propio entorno, puesto que los atributos carecen de estructura interna, por lo que dos atributos son considerados equivalentes cuando

tienen el mismo nombre. Por otro lado, es más difícil ponerse de acuerdo sobre los atributos que sobre los conceptos, puesto que cada experto puede definir un concepto de forma perspectivas diferentes, aunque compatibles e incluso equivalentes; esto es, pueden compartir la definición de los elementos básicos de un concepto, pero cada experto añade diferentes detalles al mismo concepto. Los resultados relativos a las relaciones son similares a los obtenidos para los conceptos, porque las relaciones se dan entre conceptos, por lo que si los expertos encuentran un conjunto similar de entidades conceptuales en un dominio, es factible que el conjunto de relaciones también sea similar.

La naturaleza de los dominios puede ser otra razón para explicar los resultados, puesto que los dos primeros dominios pertenecen al área tecnológica, mientras que el tercero no. Además, no existe consenso en la comunidad científica sobre el tercer dominio, esto es, el cambio climático.

| Dominio | Conceptos | Atributos | Relaciones |
|---|---|---|---|
| Videoconferencia | 14.1% | 50% | 25% |
| Circuitos integrados | 19.8% | 63.6% | 33% |
| Clima | 55.8% | 36.3% | 54% |

**Tabla 5.2 Ganancia de conocimiento para cada dominio**

| Dominio | Conceptos | Atributos | Relaciones | Ganancia global |
|---|---|---|---|---|
| **Videoconferencia** | **131** | **195** | **136** | **32%** |
| **Circuitos integrados** | **55** | **118** | **41** | **46%** |
| **Clima** | **108** | **108** | **116** | **48%** |

**Tabla 5.3 Ganancia de conocimiento global para cada dominio**

*5.1.3.2 Experimento 2*

La situación inicial era un conjunto de siete expertos en el dominio de Hardware. Se pidió a los expertos que construyeran una ontología privada sobre este dominio. Las ontologías generadas fueron integradas para obtener una ontología común. En ese momento la ontología común fue enviada a cada experto, y se les pidió que refinaran sus ontologías privadas para realizar una segunda integración. La única influencia externa a la que estaban sometidos los expertos era el acceso a la ontología integrada. Con el objeto de mantener este experimento manejable, se pidió a los expertos que limitaran el tamaño de sus ontologías a unos 30 conceptos. Las características de las ontologías construidas por cada experto en cada paso se muestran en las Tablas 5.4-5.5.

| Experto | Relaciones | | Conceptos |
|---|---|---|---|
| | Taxonomía | Mereología | |
| E1 | 24 | 17 | 43 |
| E2 | 27 | 4 | 30 |
| E3 | 23 | 6 | 28 |
| E4 | 22 | 5 | 26 |
| E5 | 20 | 7 | 29 |
| E6 | 24 | 4 | 28 |
| E7 | 49 | 16 | 68 |

**Tabla 5.4. Ontologías para el primer paso de integración**

| Experto | Relaciones | | Conceptos |
|---|---|---|---|
| | Taxonomía | Mereología | |
| E1 | 25 | 25 | 52 |
| E2 | 36 | 8 | 45 |
| E3 | 23 | 6 | 28 |
| E4 | 40 | 8 | 44 |
| E5 | 25 | 10 | 37 |
| E6 | 39 | 6 | 44 |
| E7 | 51 | 16 | 70 |

**Tabla 5.5. Ontologías para el segundo paso de integración**

Este experimento tuvo dos objetivos principales: (a) validar el entorno de integración; y (b) analizar los efectos de los procesos de integración sobre el refinamiento de ontologías para alcanzar el consenso. Para lograr el primer objetivo se midió la ganancia de conocimiento en cada iteración. La Tabla 5.6 muestra los resultados obtenidos para cada experimento para

conceptos y relaciones. Podemos ver que la ganancia de conocimiento decrece conforme realizamos más iteraciones. Esto se debe a que, tras el primer paso de integración, los constructores de las ontologías refinaron sus ontologías privadas usando el conocimiento contenido en los resultados del primer proceso de integración. Por consiguiente, como las ontologías del segundo paso tienen un mayor grado de similaridad tras los refinamientos, la ganancia de conocimiento decrece.

| Iteración | Conceptos | Relaciones | Ganancia global |
|---|---|---|---|
| Iteración 1 | 403% (181) | 528% (225) | 472% |
| Iteración 2 | 216% (160) | 472% (274) | 377% |

**Tabla 5.6 Ganancias medias de conocimiento para las dos iteraciones**

Por lo que respecta al segundo objetivo, se calculó el impacto de cada ontología individual sobre la ontología integrada. Llamamos impacto de una ontología a la cantidad de elementos de conocimiento proporcionados por una ontología a la ontología integrada. Los valores de impacto para cada ontología en cada iteración se muestran en la Tabla 5.7.

| Ontología | Impacto 1 | Impacto 2 |
|---|---|---|
| O1 | - | - |
| O2 | 39% | 31% |
| O3 | 27% | 17% |
| O4 | 19% | 18% |
| O5 | 11% | 7% |
| O6 | 11% | 13% |
| O7 | 18% | 13% |
| media | 21% | 16% |

**Tabla 5.7 Valores de impacto**

Mirando a ambos conjuntos de ontologías (Tabla 5.4 y Tabla 5.5), las ontologías de la segunda iteración tienen mayor tamaño que las pertenecientes a la primera. Sin embargo, el impacto medio es menor en la segunda iteración. Esto se debe al mayor solapamiento entre las respectivas ontologías. El solapamiento entre ontologías muestra lo similares que son las ontologías. Por lo tanto, cuando buscamos el consenso, dicho solapamiento debe ir aumentando. Los expertos refinan sus ontologías usando su propio conocimiento y el conocimiento contenido en la ontología integrada, que puede ser vista como el conocimiento aceptado por la comunidad puesto que la ontología integrada sólo contiene conocimiento consistente.

Finalmente, debería decirse que el entorno no ha sido evaluado en dominios de aplicación grandes debido a la problemática inherente a evaluar herramientas y entornos de adquisición de conocimiento (Shadbolt, O'Hara and Crow, 1999). Sin embargo, somos conscientes de la pertinencia de estudiar el rendimiento del marco de integración con ontologías de gran tamaño (sobre 100-200 conceptos).

## 5.2 VALIDACIÓN DEL PROCESO DE CONSTRUCCIÓN DE ONTOLOGÍAS

Para el proceso de creación de ontologías hemos intentado seguir los principios para el desarrollo de ontologías mencionados en (Gómez-Pérez and Benjamins, 1999), tales como completitud, coherencia, mínimo compromiso ontológico o estandarización de nombres. Se ha adoptado una visión de completitud similar a la de (Gruber, 1995), donde completitud significa que la definición de un concepto se expresa en términos de condiciones necesarias y suficientes, obteniendo así una definición completa de cada término. Por coherencia entendemos consistencia, la cual es garantizada por nuestro modelo. El compromiso ontológico mínimo implica un acuerdo para usar el vocabulario compartido de forma consistente, aspecto que es un requerimiento de este enfoque. La estandarización de nombres es difícil de conseguir puesto que cada usuario genera su ontología independientemente del resto, aunque los usuarios tienen la posibilidad de obtener la integración de las ontologías de otros usuarios; como resultado de este proceso pueden conocer la terminología usada por otros usuarios y seleccionar el término más

apropiado para cada concepto o atributo. Por tanto, podemos decir que la estandarización de nombres se consigue cuando los usuarios acuerdan el vocabulario a utilizar.

El proceso de construcción de ontologías descrito en las secciones anteriores ha sido validado en cuatro dominios. Los cuatro dominios seleccionados son muy distintos entre sí, pues varían desde dominios científicos (p.ej. diagnóstico médico) hasta dominios de ficción (p.ej., vampiros). Para ser preciso, los dominios a los que hacemos referencia son: "Enfermedades y plagas en tomates" (D1), "Diagnóstico de enfermedades causadas por protozoos" (D2), y "Drácula y los vampiros", "Diagnóstico de roturas en retroproyectores" (D4). Para cada dominio compararemos algunos datos ontológicos obtenidos al construir dos agentes cooperativamente las correspondientes ontologías, tales como el número de conceptos, el número total de atributos, la profundidad máxima, el número total de relaciones taxonómicos y mereológicos, etc.

A continuación describiremos los dominios anteriormente mencionados:

- D1: Las enfermedades y plagas en los tomates son factores que tienen gran influencia en la calidad y cantidad de unidades cultivadas. Por lo tanto, es crucial disponer de vías para prevenir y solucionar estos problemas. El conocimiento del dominio usado para crear las ontologías correspondientes fue extraído de entrevistas con expertos. Se ha hecho un estudio reducido, centrado en los males y plagas más importantes que pueden padecer los tomates.

- D2: Este dominio se centra en las enfermedades más comunes causadas por protozoos en humanos y animales. Para ello, se tomó en cuenta el proceso de diagnóstico que seguiría un experto, cuyo objetivo es asignar el tratamiento correcto para curar a los pacientes, bien sea animal o humano.

- D3: Este dominio de aplicación trata el origen y la presencia de vampiros reales (tal y como lo entienden los expertos) en la historia de la Humanidad, sus diferencias con vampiros ficticios, y pruebas analizadas desde diferentes puntos de vista científicos. En

resumen, se tratan todos los aspectos que nos llevan a representar mentalmente el concepto de vampiro humano. La construcción de la correspondiente ontología no se basó en entrevistas con expertos, sino en algunas cintas disponibles sobre el tema. Esto es también una restricción para la ontología, porque estos vídeos fueron la única fuente de información utilizada.

- D4: El problema a resolver aquí es la interpretación de síntomas de rotura en retroproyectores. Se estudiaron los siguientes fallos: problemas mecánicos en el proyector, perturbaciones de sonido, encuadre imperfecto, películas imperfectas.

Al comienzo de esta sección se mencionó un conjunto de principios útiles para construir ontologías. Para evaluar la calidad de una ontología una vez ya construida, hemos seguido un enfoque similar al definido en (Crow and Shadbolt, 1999). Ahí se define una manera para evaluar la calidad de una ontología. Una buena ontología debe cumplir los siguientes requisitos: precisión, anchura, consistencia, completitud y utilidad. Las ontologías usadas para la validación del proceso de construcción cumplen dichas propiedades. En particular, la construcción de las ontologías usadas para esta evaluación se puede caracterizar como sigue.

El conocimiento usado para construir las ontologías se obtuvo directamente a partir de expertos, o bien directamente mediante entrevistas o indirectamente a través de vídeos. Las palabras de los expertos fueron contrastadas con el conocimiento presente en las respectivas ontologías, por lo que se asegura que cada pieza de conocimiento es todo lo precisa como sea posible. Por lo que respecta a la anchura de la ontología, como se puede ver en la Tabla 5.8, la organización ontológica presenta una cantidad de enlaces taxonómicos/mereológicos entre conceptos que asegura que la estructura de la ontología integrada no es exclusivamente en profundidad. El modelo ontológico formal usado se basa en algunas propiedades que aseguran la consistencia interna de las ontologías construidas. Por lo tanto, las ontologías usadas en esta validación son consistentes. Debido al origen del conocimiento, el conocimiento contenido en las ontologías es completo según los expertos que participaron en su construcción, por lo que se cumple otro requerimiento. Para nosotros, la utilidad de una ontología depende del uso deseado para las ontologías. Así, la ontología sobre “Drácula y los vampiros” puede ser muy útil con un

propósito educacional, al ser un tema interesante de estudio. Para un estudiante es posiblemente más fácil y atractivo aprender con esta ontología que con otra relacionada con protozoos debido a la terminología científica usada en dicha ontología. Las ontologías aquí presentadas se pueden aplicar de diferentes formas, como con fines educativos/formativos.

### 5.2.1 Resultados Ontológicos

La siguiente tabla muestra las propiedades de las cuatro ontologías construidas para cada dominio.

| Dominio | NOC | MD | NOA | NOMR | NOTR | MNOTC | MNOMC |
|---|---|---|---|---|---|---|---|
| D1 | 28 | 5 | 64 | 11 | 17 | 6 | 5 |
| D2 | 20 | 4 | 37 | 6 | 14 | 4 | 3 |
| D3 | 38 | 6 | 65 | 7 | 45 | 7 | 2 |
| D4 | 42 | 6 | 174 | 4 | 38 | 8 | 2 |

**Tabla 5.8. Resultados ontológicos**

donde

*NOC*: Número de conceptos

*MD*: Profundidad máxima

*NOA*: Número de atributos

*NOMR*: Número de relaciones mereológicas

*NOTR*: Número de relaciones taxonómicas

*MNOTC*: Número máximo de hijos taxonómicos

*MNOMC*: Número máximo de hijos mereológicos

La siguiente figura (Figura 5.1) muestra una representación gráfica de la información contenida en la tabla anterior.

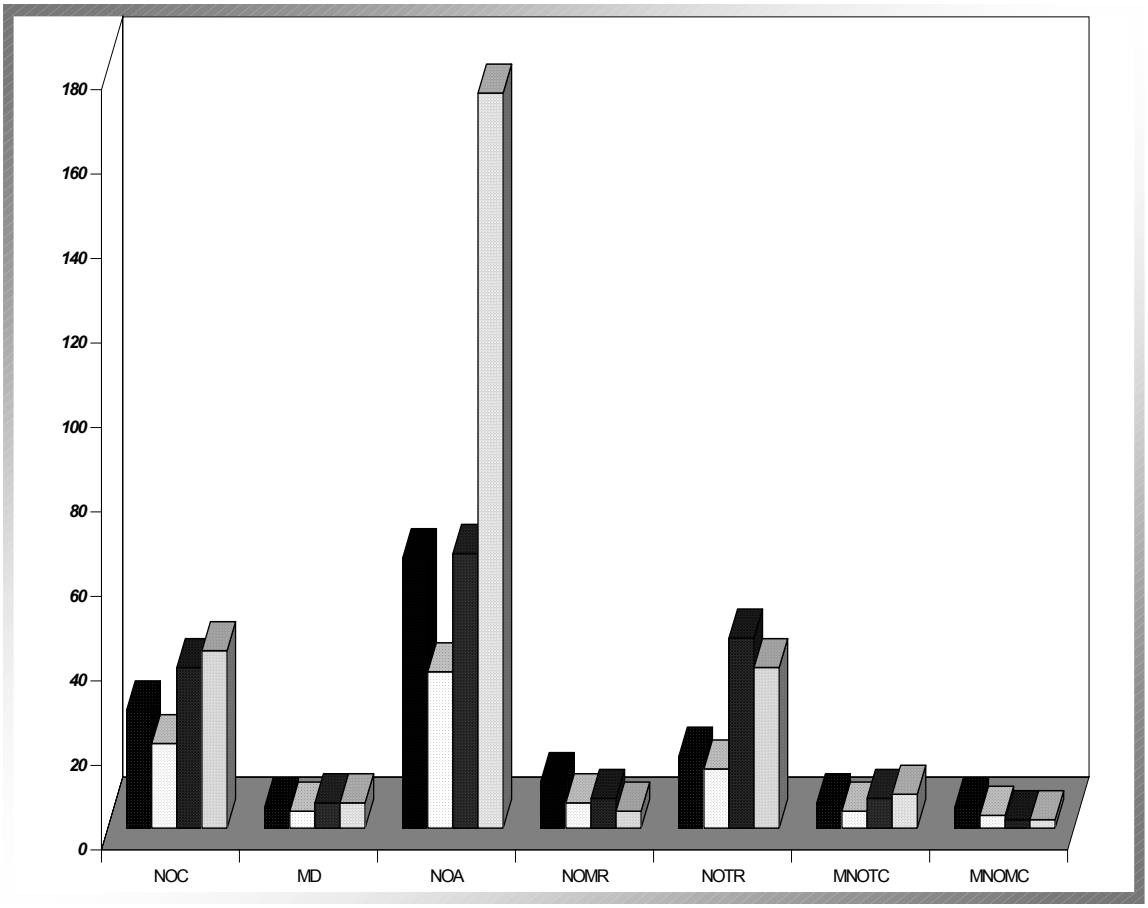


**Figura 5.1. Representación gráfica de los resultados ontológicos**

Se pueden hacer algunas consideraciones sobre las ontologías estudiadas y sus respectivos dominios. Si observamos los resultados, podemos ver que los parámetros medidos en D1 y D2 tienen un propiedad común. Los valores para D1 son siempre mayores que para D2, mientras

que los valores para D3 y D4 son algunas veces mayores y otras menores, por lo que siguen un comportamiento diferente. Esto podría ser provocado por el hecho de que D1 y D2 son dominios relativamente cercanos en términos de lo que se pretende obtener con estas ontologías. En particular, ambos dominios son relativos a diagnósticos de enfermedades en tomates o provocados por protozoos. Por lo tanto, hay algunas similaridades en las ontologías resultantes de ambos procesos. La carga temporal necesaria para la construcción de cada ontología fue similar, de forma que la profundidad del estudio para cada dominio es también similar (ver Tabla 5.5).

El tercer dominio (D3) es, como ya hemos dicho, un dominio ficticio. Ahí no hay ninguna posibilidad de comparación fiable con el resto de dominios en términos de concepto o atributos. En otras palabras, la diferencia de contexto con respecto a los otros dominios lo convierte en un caso de estudio particular. De esta forma, intentar comparar este dominio con los otros resultaría en comparar un número muy alto de relaciones taxonómicas (45) con los de D1 y D2, que presentan un número bastante menor (17 y 14 respectivamente).

Podemos explicar este fenómeno como sigue. El dominio de los vampiros, al ser parcialmente ficticio, genera una gran cantidad de conceptos, que se pueden dividir en históricos y ficticios. El hecho de que la fuente de información haya sido diferente en este dominio (cintas de vídeo) con respecto a los otros (expertos) puede ser otra causa para esta diferencia. Los expertos que interactúan directamente con los ingenieros de conocimiento van directamente al grano mientras que la información contenida en una cinta de vídeo es menos precisa y puede generar un mayor número de conceptos. En el contexto de ontologías construidas a partir de expertos, podría sorprender la diferencia en cuanto a número de atributos entre D1, D2 y D4, porque D4 tiene un número de atributos más alto que D1 y D2: D4 tiene 174 mientras que D1 y D2 tienen 64 y 37 respectivamente. Una explicación para esto podría ser el número de entrevistas con los expertos. Para modelar D1 y D2 se realizaron tres entrevistas con expertos, mientras que para D4 se realizaron ocho, lo cual posibilita que el experto detalle más los conceptos tratados.

Otra observación sobre los datos es que D3 y D4 tienen más conceptos que D1 y D2. D3 y D4 se centran claramente en la clasificación de conceptos del dominio. Este aspecto puede ser detectado por el número de relaciones taxonómicas con respecto a las mereológicas. Por consiguiente, las tareas de clasificación suelen producir más relaciones taxonómicas que mereológicas debido a la especialización que se realiza en base a los atributos de los conceptos. Este hecho puede ser visto como una ventaja con respecto al proceso mereológico.

Para terminar con los comentarios acerca de los resultados obtenidos, no debemos olvidar que estas ontologías fueron construidas por estudiantes de último año de la Facultad de Informática de la Universidad de Murcia en un período de tiempo relativamente corto, y no se pretendía que las ontologías fueran perfectas; de hecho posiblemente podrían ser mejoradas a través de reingeniería ontológica (Gómez-Pérez, 1999).

## 5.3 APLICACIONES

En este apartado se describen las diferentes aplicaciones del entorno de integración. En particular se señalan dos aplicaciones principales. La primera corresponde a la construcción cooperativa de ontologías en biología molecular. Por otro lado, el modelo ontológico y el entorno de integración presentado aquí se han usado como tecnología básica de representación y como motor de inferencia para construir un sistema para facilitar la gestión de conocimiento en Unidades de Cuidados Intensivos.

### 5.3.1 CONSTRUCCIÓN DE ONTOLOGÍAS EN BIOLOGÍA MOLECULAR

#### *5.3.1.1 Motivación*

Actualmente existen tres tipos de proyectos relacionados con representación de conocimiento en bioinformática: (1) bases de datos relacionales; (2) taxonomías sin o con poco

conocimiento axiomático; y (3) sistemas que intentan simular el razonamiento humano en dominios muy específicos (Fridman-Noy and Hafner, 2000). La mayoría de ellos se basan en el concepto de ontología, que es una especificación de una conceptualización diseñada para ser reutilizada en diversas aplicaciones e implementaciones (Gruber, 1993).

Un ejemplo del primer grupo es el proyecto Gene Ontology (GO), una base de datos independiente que promueve la unificación terminológica en bases de datos biológicas (GO, 2000). La ontología desarrollada en (Karp, 2000) sería un ejemplo del tipo de proyectos encuadrados en el segundo grupo. La ontología se usa para proporcionar una codificación semántica estructurada de las funciones biológicas. Se han desarrollado lenguajes y variantes de XML para representar conocimiento biológico [(Gilmour, 2000);(Guerrini and Jackson, 2000)] Dentro del tercer grupo de proyectos podemos destacar el proyecto GENSIM (Karp, 1993) y el sistema presentado en (Andrade and Valencia, 1998).

Asimismo, el diseño, desarrollo y uso de ontologías en bioinformática está creciendo rápidamente, y está ganando cada vez más relevancia. Esto se puede observar por el creciente número de proyectos de investigación que hacen uso de ontologías con diversas finalidades (ver por ejemplo [(Baker et al, 1999); (Giudicelli and Lefranc, 1999); (Westbrook and Bourne, 2000)].

Además, la creación de métodos para definir y mantener modelos compartidos de dominios biológicos ha sido identificada como una tarea crítica en (Altman et al, 2001). Sin embargo, estos esfuerzos han llevado al desarrollo de bases de datos, ontologías y sistemas independientes. Una tarea interesante que todavía no ha sido atendida suficientemente es la integración de varias ontologías biológicas, lo cual podría ayudar a conseguir el objetivo de armonizar diferentes terminologías y aprovechar las ventajas de compartir y reutilizar conocimiento ontológico.

Analizando todos los factores anteriores, llegamos a la conclusión de que la construcción cooperativa de ontologías en biología molecular es un área adecuada para aplicar nuestro entorno de integración.

### *5.3.1.2 Descripción*

El objetivo de esta aplicación del entorno de integración para la construcción cooperativa de ontologías en el dominio de la biología molecular fue demostrar que nuestro enfoque de integración ayuda a los expertos a construir ontologías de forma más cómoda. Dos expertos construyeron dos ontologías en dos dominios biológicos a través de procesos de integración. El primer dominio fue la célula, y el segundo fue la proteína. A continuación describimos ambos procesos de construcción ontológica.

#### *ONTOLOGÍA DE LA CÉLULA*

Se pidió a dos expertos que construyeran una ontología parcial de tamaño reducido en el dominio de la célula. El hecho de solicitar ontologías de tamaño reducido fue debido a dos motivos. Por un lado, los expertos se centrarían en la parte más significativa (bajo su punto de vista) del dominio, de forma que se podrían comprobar las capacidades para detectar inconsistencias y equivalencias del entorno. Por otro lado, se podría realizar un análisis terminológico sobre la parte esencial del dominio. De esta forma, una vez construidas las dos ontologías se procedió a su integración usando nuestro entorno. Podemos apreciar en la Figura 5.2 que la ontología construida por el primer usuario sólo contiene relaciones mereológicas. Esta ontología posee 11 conceptos y 10 relaciones mereológicas. Por lo que respecta a la ontología construida por el segundo experto (ver Figura 5.3), ésta tiene 14 conceptos, 9 relaciones mereológicas y 4 relaciones taxonómicas. En estas figuras se han omitido los atributos, pero se comprobó que se definieran los atributos de forma consistente. Esto se debe al hecho que cuando el proceso de integración encuentra conceptos con el mismo nombre perteneciendo a ontologías distintas y sin atributos en común, entonces ambas ontologías sería marcadas como inconsistentes atendiendo a la definición especificada en el capítulo 3.

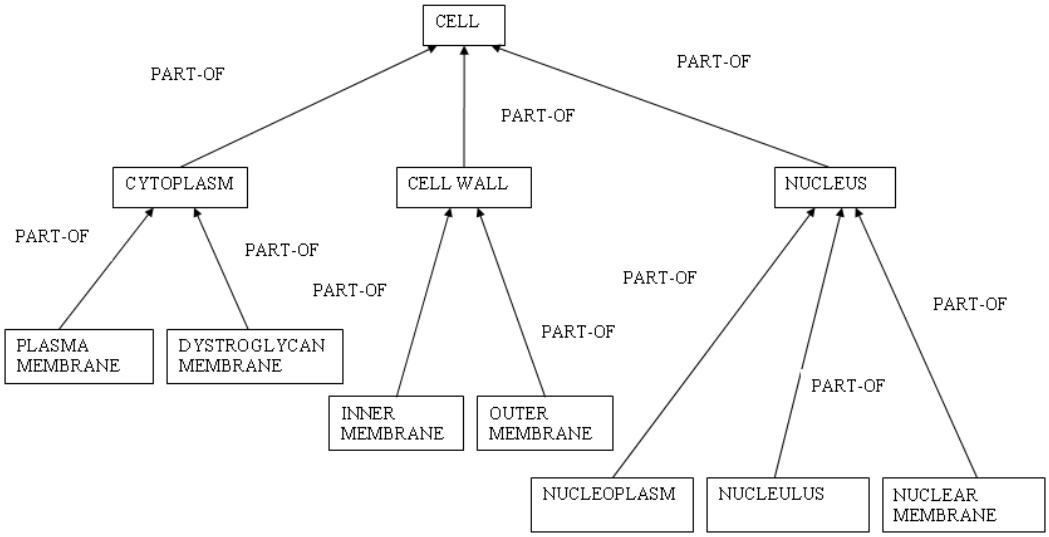


**Figura 5.2 Ontología del primer experto**

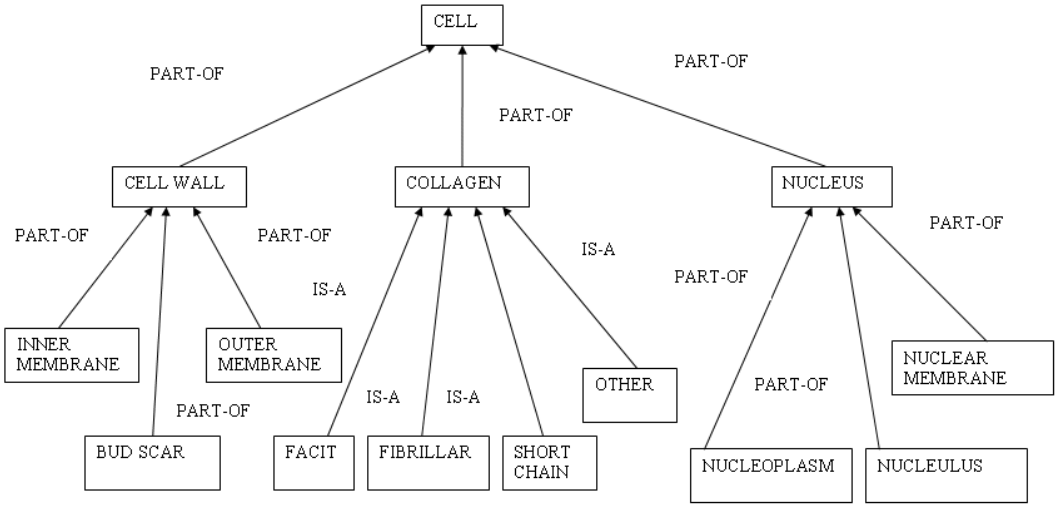


**Figura 5.3 Ontología del segundo experto**

Describamos cómo se integran las ontologías anteriores con nuestro entorno de integración. En primer lugar, las ontologías fuente deben ser comparadas para asegurarnos de que no son ni equivalentes ni inconsistentes. En este caso solo tenemos dos ontologías para integrar, y puesto que no son ni inconsistentes ni equivalentes, es decir, puesto que son compatibles para la integración, el conjunto de ontologías a integrar estará compuesto por estas dos ontologías. Por otro lado, supongamos que el proceso de integración fue solicitado por un individuo diferente de los expertos que construyeron las ontologías. En este caso, la ontología del segundo experto será la ontología referencia puesto que posee más conceptos (14) que la del primer experto (11). El primer paso del proceso de integración se muestra en la Figura 5.4.

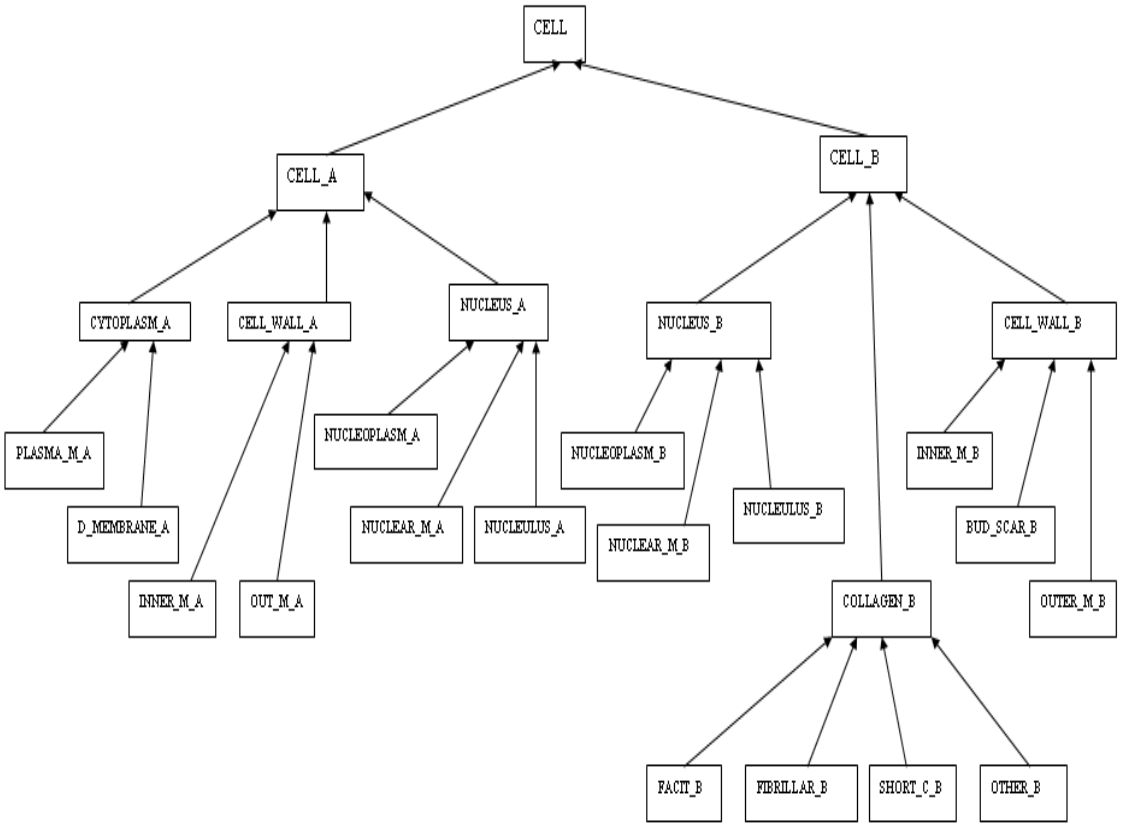


**Figura 5.4 Ontología derivada de la integración de la célula**

El esqueleto inicial de la ontología transformada será la ontología del segundo experto y a ella deberemos añadir el conocimiento incluido en la ontología del primer experto. Para ello se seguirá un proceso iterativo de forma que en cada iteración se incorpora un nuevo concepto a la ontología global. Al inicio, la ontología global tiene los siguientes conceptos: {Cell, Cytoplasm, Plasma membrane, Dystroglycan membrane, Cell wall, Inner membrane, Outer membrane, Nucleus, Nucleoplasm, Nuclear membrane}.

Esta ontología ya contiene conceptos llamados "Cell", "Cell wall", "Inner membrane", "Outer membrane", "Nucleus", "Nucleoplasm" y "Nuclear membrane", de forma que se debe comprobar que estos conceptos, pertenecientes a la ontología del experto 1, no presentan inconsistencias con respecto a los de mismo nombre ya incluidos en la ontología global. Puesto que hemos asumido que conceptos con el mismo nombre tienen al menos un atributo común, no existe inconsistencia desde el punto de vista de los atributos. Desde un punto de vista estructural, las relaciones en las cuales aparecen estos conceptos comunes en la ontología del experto 1 también aparecen en la ontología global, por lo que tampoco existiría inconsistencia desde el punto de vista estructural. De esta forma se pueden unificar los conceptos comunes.

Ahora, se debe insertar en la ontología global los conceptos de la ontología del experto 1 que no hayan sido todavía incluidos en la global, esto es, "Cytoplasm", "Plasma membrane" y "Dystroglycan membrane". Para ello, se debe encontrar la ubicación adecuada en la ontología. Esto se consigue al encontrar en la ontología global la situación de sus conceptos padre (en la ontología del experto 1) en la ontología global, y establecer las relaciones correspondientes. Por ejemplo, en el caso de "Cytoplasm", su único concepto padre (mereológico) es "Cell", de forma que lo insertaremos en la ontología global como hijo mereológico de "Cell". Actuaremos de forma similar para "Plasma membrane" y "Dystroglycan membrane". Finalmente, la FIgura 5.5 muestra el resultado de este proceso de integración.

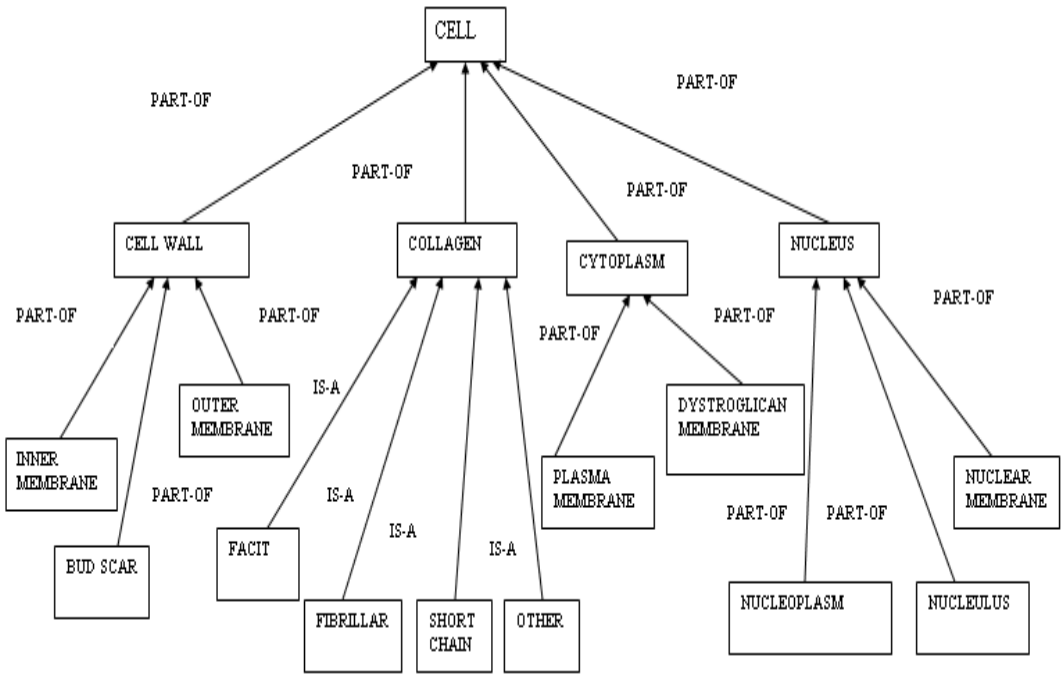


**Figura 5.5 La ontología transformada y final de la célula**

La Tabla 5.9 refleja el número de conceptos y relaciones de cada una de las ontologías implicadas en este proceso de integración.

| **Ontología** | **Conceptos** | **Relaciones** |
|---|---|---|
| **Experto 1** | *11* | *9* |
| **Experto 2** | *14* | *13* |
| **Global** | *17* | *17* |

**Tabla 5.9. Valores para cada categoría ontológica**

Ahora podemos aplicar la metodología descrita al inicio de este capítulo para medir la ganancia de conocimiento asociada a este proceso de integración. Los resultados se muestran en la Tabla 5.10.

| Ontología | Categoría Ontológica | $\Delta_i$ | Ganancia de Conocimiento | Ganancia Global |
|---|---|---|---|---|
| **Experto 1** | *CONCEPTOS* | *6/11= 54'5%* | *9/25= 36%* | |
| **Experto 2** | | *3/14 = 21'4%* | | |
| **Experto 1** | *RELACIONES* | *8/9= 88'8%* | *12/22= 54'5%* | *45%* |
| **Experto 2** | | *4/13=30'7%* | | |

**Tabla 5.10. Ganancia de conocimiento asociada a este experimento**

Resumiendo este proceso de integración, las dos ontologías no eran inconsistentes ni equivalentes. Se encontraron equivalencias entre conceptos de las dos ontologías. Concretamente los siguientes pares de conceptos fueron encontrados equivalentes ($Nucleus_A$, $Nucleus_B$), ($Nucleulus_A$, $Nucleulus_B$), ($Nucleoplasm_A$, $Nucleoplasm_B$), y (Nuclear $membrane_A$, Nuclear $membrane_B$). Se podría pensar que las organizaciones mereológicas establecidas en ambas ontologías a partir de los conceptos "Cell Wall" son equivalentes. Sin embargo, $Cell\ Wall_A$ y $Cell\ Wall_B$ no son equivalentes porque sus respectivos conjuntos de hijos mereológicos no son equivalentes.

*ONTOLOGÍA DE LA PROTEÍNA*

Se usaron dos expertos para este experimento. Los conceptos y las relaciones mereológicas y taxonómicas de la ontología del experto 1 son mostradas en la Figura 5.6. Dicha ontología tiene 18 conceptos, 14 relaciones taxonómicas y 3 relaciones mereológicas. Por otra parte, la ontología del experto 2 (ver Figura 5.7) posee 14 conceptos, 7 relaciones mereológicas, 6 relaciones taxonómicas y 3 topológicas. A su vez, cada concepto tiene asociado un conjunto de atributos, que no se muestran en las figuras pero que se recogen en la Tabla 5.11. En la Tabla 5.11 se distinguen dos tipos de atributos, a saber, específicos y heredados. Los atributos específicos aparecen con un estilo normal, mientras que los heredados se subrayan. Vamos a describir el proceso de integración de ambas ontologías. En primer lugar se comparan las ontologías a integrar para, como sucedió en el caso anterior, ver si son equivalentes o

inconsistentes. Tal y como en el caso de la célula, dichas ontologías son compatibles para la integración, y volveremos a suponer que la integración fue solicitada por un individuo externo al proceso.

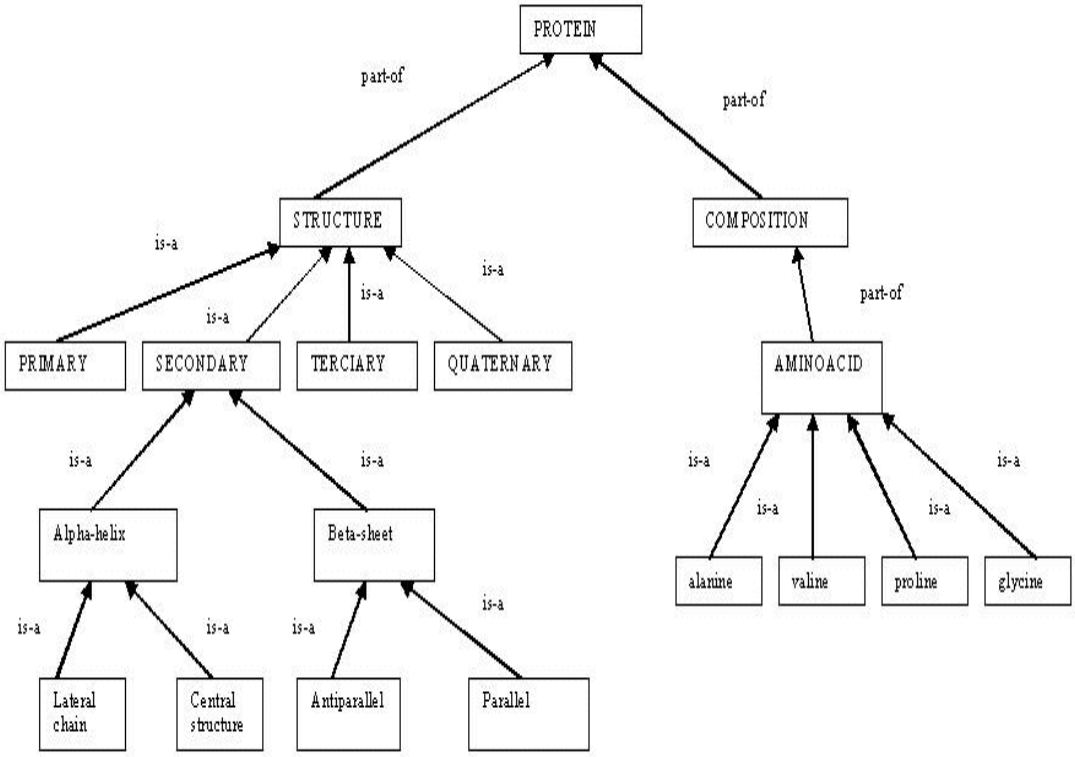


**Figura 5.6. Ontología sobre proteínas del primer experto**

Por lo tanto, en este caso se tomará como referencia la ontología del primer experto, puesto que tiene más conceptos que la del segundo experto. Así, el esqueleto inicial de la ontología global será la ontología de referencia, y el conocimiento ontológico incluido en tal ontología será incorporado a ella, por lo que faltaría incorporar el conocimiento de la segunda ontología. Para ello, al igual que en el caso anterior se sigue un proceso iterativo de forma que se procesa un concepto de la ontología del segundo experto cada vez.

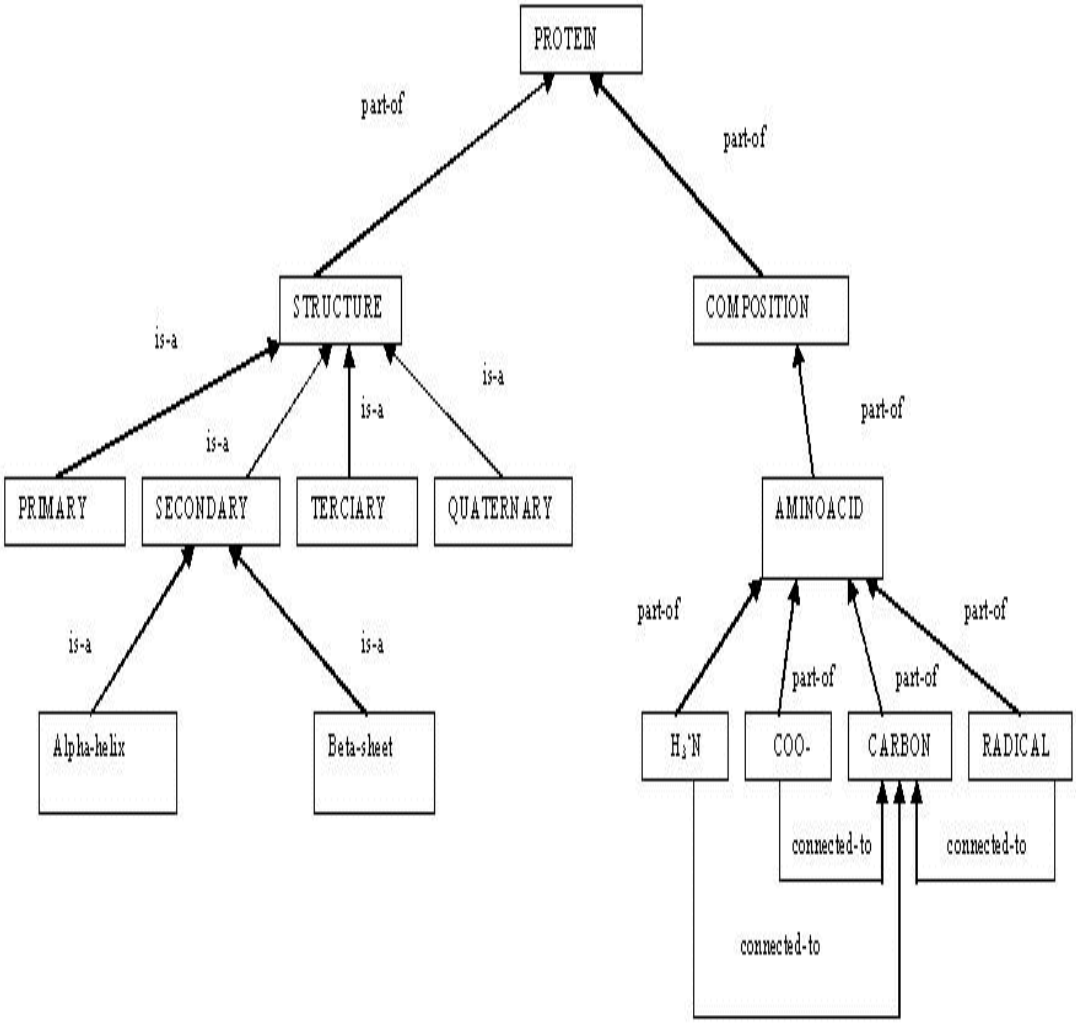


**Figura 5.7. La ontología de la proteína construida por el segundo experto**

| **EXPERTO 1** | | **EXPERTO 2** | |
|---|---|---|---|
| **Concepto** | **Atributos** | **Concepto** | **Atributos** |
| *Protein* | Name, function | *Protein* | Name |
| *Structure* | Structure_Type | *Structure* | Structure_Type |
| *Primary* | Order, Structure_Type | *Primary* | Order, Structure_Type |
| *Secondary* | Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type | *Secondary* | Phi, psi, N-terminus charge, C-terminus charge, Structure_Type |
| *Terciary* | Spatial form, Structure_Type | *Terciary* | Spatial form, Structure_Type |
| *Quaternary* | Number chains, Structure_Type | *Quaternary* | Number chains, Structure_Type |
| *Alpha-helix* | Ccap, Ncap, geometry, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type | *Alpha-helix* | Ccap, Ncap, Phi, psi, N-terminus charge, C-terminus charge, Structure_Type |
| *Beta-sheet* | Orientation, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type | *Beta-sheet* | Stability, orientation, Phi, psi, N-terminus charge, C-terminus charge, Structure_Type |
| *Lateral chain* | Ccap, Ncap, geometry, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type | *Composition* | Distribution |
| *Central structure* | Ccap, Ncap, geometry, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type | *Aminoacid* | name, composition |
| *Parallel* | Stability, orientation, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type | Carbon | Atomic mass |
| *Antiparallel* | Stability, orientation, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type | COO- | Molecular mass, purity |
| *Composition* | Distribution | $H_3^+N$ | Molecular mass, purity |
| *Aminoacid* | Property, function, name | Radical | Name, composition |
| *Alanine* | GCx, Property, function, name, composition | | |
| *Valine* | GUx, Property, function, name, composition | | |
| *Proline* | CCx, Property, function, name, composition | | |
| *glycine* | GGx, Property, function, name, composition | | |

**Tabla 5.11 Atributos de las dos ontologías**

En la Figura 5.8 podemos ver la ontología que resulta de la integración de las dos ontologías construidas por los expertos.

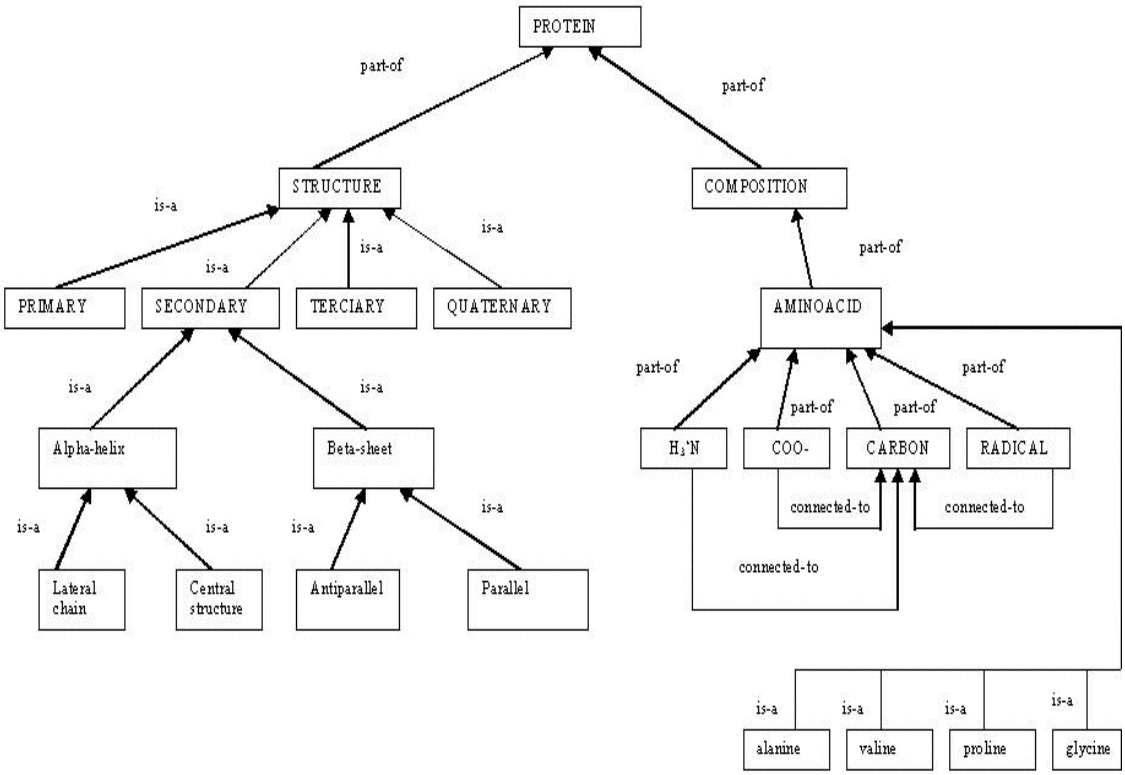


**Figura 5.8. La ontología final de la proteína**

La Tabla 5.12 muestra el número de conceptos y relaciones de las dos ontologías fuente y la global. La misma forma para medir ganancias de conocimiento se ha aplicado a las ontologías correspondientes. Los resultados de este análisis aparecen en la Tabla 5.13. Se pueden extraer algunas conclusiones de estos resultados. Se obtiene un resultado bastante interesante en la categoría de los conceptos, puesto que una ganancia del 37'5% es un valor relativamente alto para estas dos ontologías pequeñas, puesto que expertos diferentes suelen capturar similares entidades conceptuales en un dominio. Suele ocurrir que ésta sea la categoría con menos ganancia de conocimiento cuando dos expertos representan el mismo dominio desde un mismo

punto de vista. Sin embargo, en este caso ambas conceptualizaciones son realizadas desde puntos de vista parcialmente distintos.

| **Experto** | **Conceptos** | **Relaciones** | **Atributos** |
|---|---|---|---|
| **Experto 1** | *18* | *17* | *21* |
| **Experto 2** | *14* | *16* | *23* |
| **Global** | *22* | *24* | *29* |

**Tabla 5.12. Elementos ontológicos para cada ontología**

Analizando ambas ontologías, el primer experto es especialista en tipos de aminoácidos, mientras que el segundo experto se dedica más a describir la estructura interna de los aminoácidos. Por lo tanto, al integrar tal conocimiento, ambos expertos se darán cuenta de que su conocimiento individual aumenta. Algunas veces, aunque no en este ejemplo, la ganancia de conocimiento para conceptos tiene un menor valor debido a la capacidad de detección de sinónimos del entorno de integración. Los resultados relativos a las relaciones son similares al obtenido para conceptos puesto que las relaciones se dan entre conceptos por lo que, si los expertos encuentran conjuntos similares de conceptos, es probable que los conjuntos de relaciones sean también similares.

Los atributos son la categoría para la que se obtiene menor ganancia de conocimiento. Sin embargo, se supone que la definición de atributos es una tarea heterogénea, de forma que la mayor disensión, y por tanto la mayor ganancia, debería haber aparecido en esta categoría. En este ejemplo, el ratio de atributos por concepto es 29/22=1.3, lo cual es bastante bajo, por lo que podemos decir que cada concepto queda definido por un único atributo específico. Así, las posibilidades de divergencia es menor debido a que los desacuerdos cuando se definen atributos ocurren para los atributos menos significativos, ya que los expertos suelen coincidir en los atributos más relevantes.

| Experto | Categoría Ontológica | $\Delta_i$ | Ganancia de Conocimiento | Ganancia Global |
|---|---|---|---|---|
| **Experto 1** | CONCEPTOS | *4/18= 22'2%* | *12/32=37'5%* | *37'8%* |
| **Experto 2** | | *8/14= 57'1%* | | |
| **Experto 1** | RELACIONES | *7/17=41'1%* | *15/33=45'4%* | |
| **Experto 2** | | *8/16=50%* | | |
| **Experto 1** | ATRIBUTOS | *8/21=38'1%* | *14/44=31'8%* | |
| **Experto 2** | | *6/23=26%* | | |

**Tabla 5.13 Ganancia de conocimiento para el dominio de la proteína**

En este momento podemos realizar algunas afirmaciones sobre axiomas estructurales. Los axiomas estructurales son aquellos axiomas derivados de la propia estructura de la ontología, esto es, relaciones entre conceptos, los propios conceptos de la ontología y la existencia de atributos conceptuales. Por consiguiente, podemos obtener el número de axiomas estructurales contando el número de conceptos, relaciones y atributos contenidos en una ontología. En este caso, la ontología del primer experto contiene 56 axiomas estructurales, mientras que la del segundo tiene 53 y la ontología final tiene 75. Aplicando el cálculo de ganancia a los axiomas estructurales obtendríamos un resultado de 37'8%.

## 5.3.2 GESTIÓN DE CONOCIMIENTO EN UNIDADES DE CUIDADOS INTENSIVOS

### *5.3.2.1 Motivación*

La importancia de la Gestión de Conocimiento está creciendo actualmente en la Sociedad de la Información, y los dominios médicos no son una excepción. La motivación para esta aplicación es un proyecto de investigación multidisciplinar en curso entre nuestro grupo de investigación y un hospital para modelar eventos clínicos en Unidades de Cuidados Intensivos (UCIs). El objetivo final es la implantación de un sistema de gestión de conocimiento en dicha

unidad. En (Yu-N and Abidi, 1999), se considera que la gestión de conocimiento en entornos clínicos es importante debido a las características de los entornos clínicos y de la propia gestión de conocimiento. Desde una perspectiva de gestión clínica de UCIs, se pretende tener los procesos de toma de decisiones, adquisición y gestión de conocimiento en bases de conocimiento. De esta forma, tales procesos serán más eficientes en cuanto a tiempo y recursos empleados. Por otro lado, si la gestión clínica de UCIs se realiza a través de sistemas inteligentes orientados a gestión clínica, entonces, diferentes variables de gestión clínica (p.ej., médicos, equipamiento, enfermeras, etc) interactuarán con variable puramente clínicas. En esta sección se describe un sistema que hace uso del modelo ontológico y del entorno de integración para facilitar la gestión de conocimiento en Unidades de Cuidados Intensivos.

#### *5.3.2.2 Descripción del problema*

Un problema importante en la gestión de pacientes es la evolución temporal de la enfermedad (Shahar, 1997), de forma que no se pueden representar ni analizar datos clínicos sin tener en cuenta la dimensión temporal (Shahar, 2000). En este trabajo se considera dicho aspecto temporal puesto que nuestra meta es dar soporte automático a la decisión para cuidado de pacientes en períodos sustanciales de tiempo. Por consiguiente, podemos decir que el tiempo es especialmente importante cuando se usan datos clínicos temporales en aplicaciones para ayudar a la toma de decisiones, como son la determinación de un diagnóstico o la monitorización de la evolución de un paciente (Shahar, 1999). Los sistemas convencionales de monitorización de pacientes obtienen señales fisiológicas y las procesan para extraer información fiable. Cuando cierto parámetro alcanza cierto valor umbral se dispara una alarma informando de la condición anormal. Sin embargo, estos sistemas tienen importantes problemas por resolver como la baja especificidad de las alarmas y la activación de alarmas superfluas.

En Unidades de Cuidados Intensivos, donde es crítico actuar rápidamente, sería deseable contar con sistemas que aplicaran técnicas de gestión de alarmas para resolver alguno de los problemas mencionados con anterioridad. Actualmente existe un creciente interés en desarrollar sistemas de monitorización capaces de producir salidas significativas (smart alarms). Una forma de hacer esto es mediante la mejora de la calidad de la información suministrada por las alarmas

haciendo uso de la evolución temporal del paciente. Una alarma se puede definir como un aviso de un peligro que se avecina, y los sistemas tradicionales de monitorización generan una alarma cuando el valor de una variable es más alto que un valor pre-determinado.

Sería interesante reutilizar y compartir el conocimiento adquirido en una UCI con otras UCIs, y las ontologías son una tecnología apropiada para dicha tarea. En los dominios médicos (p.e.j, (Shahar et al, 1998)), las acciones y los efectos no son necesariamente instantáneas, sino que tienen una extensión temporal, esto es, se puede esperar que las acciones se desarrollen durante un intervalo o período de tiempo (Shahar and Musen, 1996). Esta característica es cubierta por el modelo ontológico midiendo la temporalidad de manera difusa. En ese sentido, otro objetivo de nuestro modelo ontológico era cubrir la dimensión temporal de las acciones (alarmas en nuestro caso).

Se han usado ontologías para representar conocimiento en dominios clínicos con diferentes propósitos. En esta sección mencionaremos algunos ejemplos. En (Schulz and Hahn, 2001) se propone un entorno ontológico para el dominio anatómico. Una ontología también ha sido usada en el sistema NeoGanesh (Dojat et al, 1997) y en el Déjà Vu (Dojat et al, 1998) para modelar el mundo en dos tipos de entidades: (1) entidades atemporales, usadas para modelar el sistema observado, y (2) entidades temporales, usadas para modelar la evolución. YAQ (Uckun et al, 1993) es otro ejemplo de ontología aplicada al razonamiento basado en modelo en dominios fisiológicos. RÉSUMÉ (Shahar and Musen, 1996) es otro sistema que usa ontologías en dominios clínicos. Las ontologías se usan en RÉSUMÉ de varias formas:

a) Ontología de parámetros-propiedades para representar el conocimiento específico del dominio (entidades, propiedades y relaciones).

b) Ontología de eventos, que incluye los eventos relevantes.

c) Ontología de contextos para representar todos los contextos potencialmente relevantes.

El modelo ontológico se usa desde dos puntos de vista diferentes. Por un lado, se usa para modelar la UCI, con su personal, equipamiento, pacientes, etc, esto es, desde un punto de vista organizacional. Por otro lado, se usa para modelar la evolución clínica de los pacientes de una UCI. Nos hemos centrado en una única UCI, aunque este enfoque puede ser fácilmente extendido a múltiples UCIs que podrían compartir conocimiento sobre hipótesis diagnósticas.

#### *5.3.2.3 Descripción del entorno*

En una UCI podemos encontrar distintos elementos tales como pacientes, médicos, enfermeras o equipamiento clínico (sistemas de alarmas, camas, etc). El propósito de esta aplicación es ayudar a los médicos a seguir la evolución de sus pacientes por medio de la integración del conocimiento suministrado por ellos y por los sistemas de alarmas inteligentes. La rapidez de respuesta es un factor crítico, por lo que tener un sistema que aplique gestión inteligente de alarmas es muy útil para evitar alarmas superfluas y redundantes, que suponen el principal problema de sistemas actuales de monitorización de pacientes. Sería también deseable disponer de alarmas etiquetadas, donde cada etiqueta describe estados patológicos del paciente.

**Gestión Clínica de Eventos**

En este contexto podemos distinguir cuatro entidades principales, que no son totalmente independientes entre sí: paciente, hipótesis diagnóstica, personal médico y evento. Tanto la evolución de los pacientes como las hipótesis diagnósticas se definen a través de un conjunto de eventos. Un evento tiene lugar en un contexto específico, el cual queda determinado por el paciente y su patología. Esto implica que deba existir alguna relación entre el evento y el resto de elementos de su contexto. La cuantificación de estas relaciones dependerá del tipo de evento. En caso de los eventos de los pacientes, se cuantificarán por medio de número naturales, mientras que en el caso de eventos de hipótesis usaremos número difusos. La razón para esta distinción es que el estado de un paciente cambia en un momento determinado, mientras que cuando especificamos una hipótesis diagnóstica, la relación temporal entre dos eventos no puede ser especificada de forma tan precisa como para pacientes porque pueden existir diferencias entre diferentes pacientes. Por lo tanto, se necesita una manera para cuantificar estas

relaciones. Más concretamente, se han usado tres tipos de relaciones difusas: (a) aproximadamente x unidades de tiempo después; (b) entre x e y unidades de tiempo después; y (c) exactamente x unidades de tiempo después.

Existen eventos cuya existencia determina un diagnóstico o que limita los posibles diagnósticos. También puede haber eventos que no pueden ocurrir si el paciente sufre una patología en concreto. Estos aspectos se resumen en los siguientes criterios:

- Criterio 1. Surge del requerimiento de la existencia de una serie de eventos para una patología en particular. Establece que el conjunto mínimo de eventos que han de ocurrir en la evolución del paciente, de forma que la correspondiente hipótesis diagnóstica sea aceptable.

- Criterio 2. Establece qué eventos no pueden aparecer en la evolución de un paciente que ha sido diagnosticado una enfermedad concreta. Supongamos que el paciente P padece la enfermedad C, cuya hipótesis diagnóstica implica que los eventos A y B no pueden aparecer en el contexto de dicha hipótesis. Entonces, si A y B se dan en la evolución del paciente P, entonces podemos suponer que C no es un diagnóstico válido para P.

- Criterio 3. Establece la posibilidad de admitir series de eventos cuya existencia no es obligatoria para la existencia de una dolencia. Esto es, aun cuando estos eventos menores no aparezcan en la evolución del paciente, el diagnóstico para el paciente seguiría siendo válido.

Los conjuntos de eventos que definen hipótesis diagnósticas no suelen ser temporalmente lineales sino que suelen contener ciclos, que son conjuntos ordenados de eventos que pueden ocurrir más de una vez al paciente. Por lo tanto, se puede modelar este tipo especial de eventos clínicos de forma especial. Para ello, podemos decir que una hipótesis diagnóstica está compuesta de eventos individuales y ciclos. Por otro lado, la evolución del paciente sólo contendrá eventos individuales y se debe detectar la ocurrencia de instancia de ciclos de la

hipótesis. Para ello se sigue el siguiente algoritmo para comprobar si la evolución de un paciente se ajusta al patrón indicado pos una hipótesis diagnóstica:

1. Si la hipótesis diagnóstica contiene ciclos, hay que detectar tales ciclos en la evolución del paciente.

2. Si se encuentra algún ciclo, reemplazar los eventos individuales del ciclo por los correspondientes ciclos en la evolución del paciente.

3. Comprobaciones a realizar:

3.1 Comprobar si se cumplen los criterios relacionados con eventos.

3.2 Comprobar la consistencia temporal entre la hipótesis diagnóstica y la evolución temporal del paciente.

4. Si se cumplen los criterios anteriores, entonces se acepta que el paciente evoluciona de acuerdo con la hipótesis diagnóstica.

Además, el proceso de inferencia asignará un valor a la hipótesis. Este valor será una medida de la exactitud del proceso de diagnóstico, de forma que cuanto más se aproxime la evolución del paciente a la hipótesis más alto será dicho valor. Por lo tanto, no se aceptará la hipótesis a menos que su valor asociado sea mayor que un valor umbral predeterminado. Este valor será establecido manualmente por el usuario, por lo que el grado de exigencia impuesto al sistema dependerá de dicho valor.

### Otras Características

Uno de los problemas que tienen los sistemas tradicionales de monitorización de pacientes es la existencia de falsas alarmas, esto es, alarmas que se activan pero que no tienen influencia real en la evolución del paciente. Cuando se comprueba la corrección de una hipótesis diagnóstica, se etiquetan las alarmas como positivas o negativas, y no se tienen en cuenta las falsas alarmas para comprobar si el paciente cumple una determinada hipótesis diagnóstica o no. Un problema surge cuando una alarma ha sido etiquetada como falsa pero no se trata de una falsa alarma sino de un olvido por parte del médico que especificó la hipótesis. Como consecuencia de realizar este proceso, si las dos ontologías, la que representa la hipótesis diagnóstica y la que representa la evolución del paciente, son consistentes, el sistema será capaz de descubrir conceptos, si hay alguno, de la ontología del paciente que no pertenece a la hipótesis diagnóstica. Esto podría suceder por diferentes motivos, como una falsa alarma o una alarma que falta. En este caso, los médicos podrían haber olvidado definir tal evento en la hipótesis.

El marco de trabajo ha sido diseñado de forma que se solicite a los expertos (médicos) si la nueva alarma debe ser parte de la hipótesis diagnóstica. En caso de respuesta afirmativa se deberán añadir restricciones temporales al nuevo evento mediante el establecimiento de relaciones con su contexto. El sistema ayudaría a tomar decisiones sobre diagnósticos inciertos de pacientes. Se podría pensar que el paciente tiene el diagnóstico propuesto por el médico, pero con frecuencia los diagnósticos en UCIs son inciertos. Entonces podemos intentar contrastar la evolución del paciente con diferentes hipótesis. Se activa el proceso de múltiples testeos cuando un paciente selecciona un grupo de hipótesis a probar. Esta petición resultaría en un conjunto de posibles hipótesis diagnósticas cuyo patrón temporal es satisfecho por la evolución temporal del paciente seleccionado. Se le daría entonces a cada hipótesis candidata un valor según las diferentes comprobaciones realizadas, y se escogería la(s) mejor(es) candidatas. Si el valor de la hipótesis escogida es mayor que el valor umbral preestablecido, entonces se aceptará la hipótesis para el paciente.

*5.3.2.4 Un Modelo Ontológico de una Unidad de Cuidados Intensivos*

El objetivo de esta sección es describir la estructura de una Unidad de Cuidados Intensivos de forma ontológica. Una UCI puede ser vista como el dominio a modelar y podemos extraer de este dominio los conceptos que pertenecen a dicho dominio, como persona o equipamiento. La Figura 5.9 muestra la representación gráfica de una UCI de acuerdo a nuestro modelo ontológico.

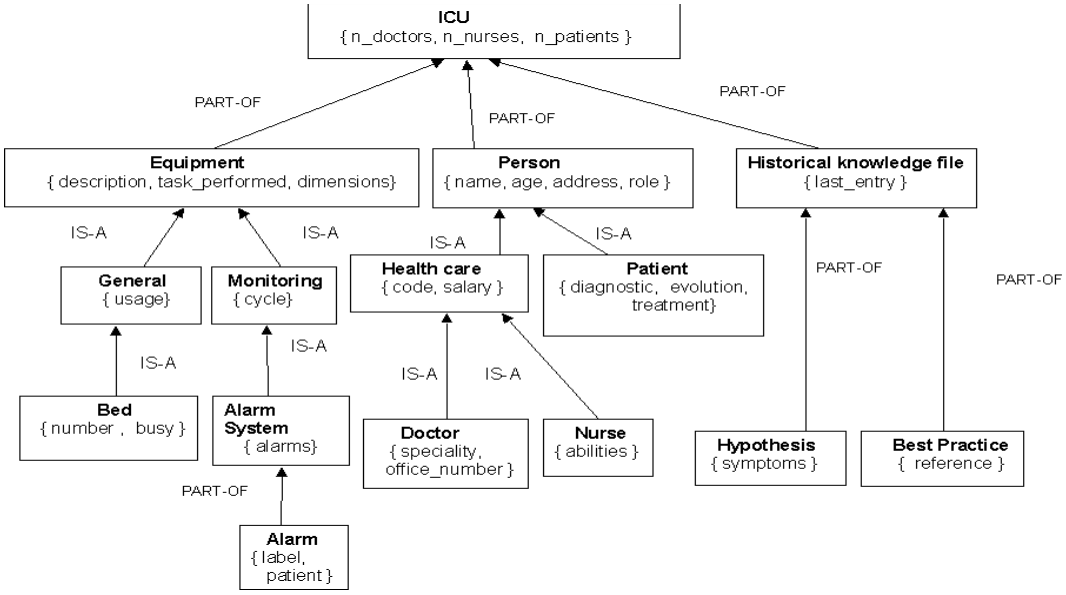


**Figura 5.9. Modelo ontológico parcial de una UCI**

Según la Figura 5.9, se puede ver una UCI como compuesta de tres partes distintas, "Equipment" (equipamiento), "Person" (personas de la UCI), y "historical knowledge file" (archivos históricos almacenados en la UCI, tales como secuencias de síntomas de hipótesis de diagnóstico o manuales de buenas prácticas). Por lo que respecta al equipamiento de la UCI, se puede categorizar de acuerdo a la tarea que realice. Podemos encontrar equipamiento de propósito general ("General") como una cama y equipamiento de monitorización ("Monitoring") como pueden ser sistemas de alarmas. Finalmente, deberíamos mencionar la taxonomía de personas ("Person") de la ontología de la UCI. Según el atributo "role", una

persona puede ser un paciente (“Patient”), o un miembro del personal sanitario (“Health Care Personnel”). Si seguimos especializando este concepto identificaríamos dos nuevos conceptos, médico (“Doctor”), y enfermera (“Nurse”).

Si ahora prestamos atención al concepto “Patient”, podemos ver que tiene un atributo llamado “evolution”. Para este atributo se puede construir una ontología de forma que se especifique la evolución temporal de dicho paciente. La ontología correspondiente a la evolución del paciente se muestra en la Figura 5.10, donde la evolución aparece compuesta por una serie de observaciones. La taxonomía de observaciones está compuesta por todas las posibles situaciones que pueden ser detectadas por el sistema. Entre estas observaciones habrá relaciones temporales, que son representadas en la figura por flechas etiquetadas con “AFTER”.

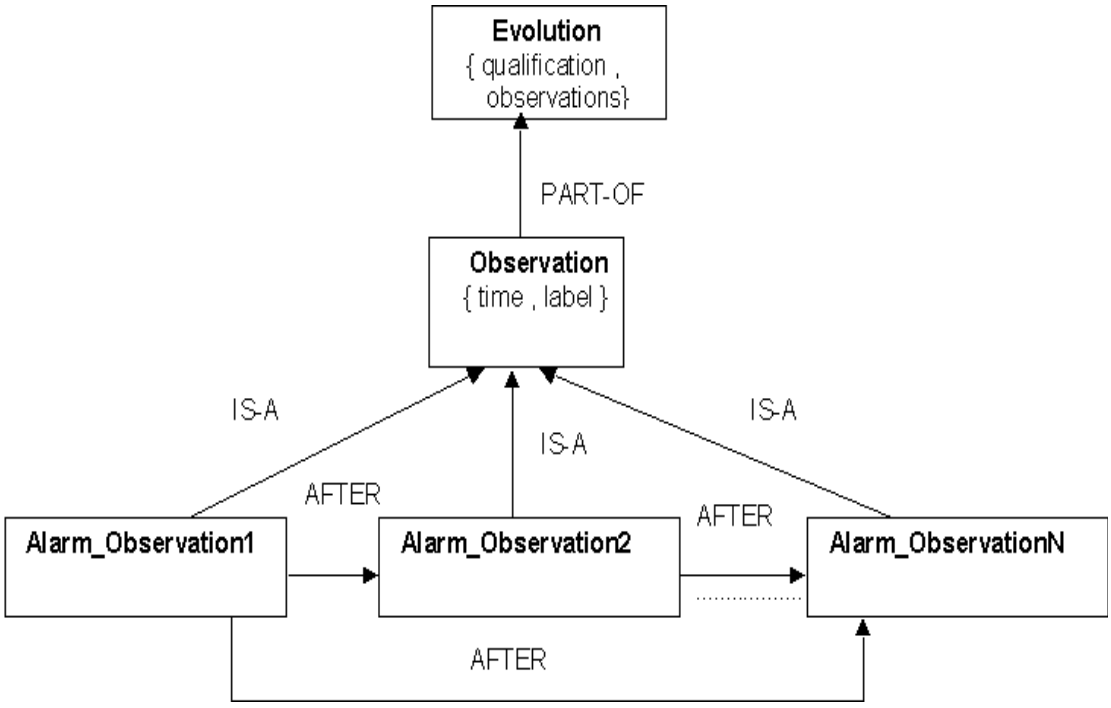


**Figura 5.10. Ontología de la evolución de un paciente**

Hemos presentado en esta sección un modelo ontológico simplificado de una UCI. Podemos extender nuestro enfoque a un entorno con más de una UCI, esto es, interconectar varias UCIs. La Figura 5.11 perfila cómo quedaría la ontología con más de una UCI. El nodo "Global_ICU" sería el nodo raíz de la ontología, mientras que cada UCI interconectada sería un hijo mereológico del concepto raíz. Cada ICU_i representaría una ontología como la especificada en esta sección. Esta interconexión facilitaría la compartición de conocimiento entre las distintas UCIs puesto que el conocimiento, en términos de hipótesis diagnósticas y tratamientos de cada UCI, podría ser accedido desde otras UCIs. Todo este conocimiento se almacena como ontologías, que son una herramienta poderosa que facilita la compartición y reutilización del conocimiento. Por lo tanto, si los médicos de una UCI incorporaran una nueva ontología (p.ej., una nueva hipótesis diagnóstica) en el sistema, el resto de UCIs podría beneficiarse de este conocimiento puesto que todas las ontologías serían compartidas. Por ejemplo, las UCIs de los países menos desarrollados podrían beneficiarse de la investigación médica realizada en los países más desarrollados de forma eficiente y poco costosa.

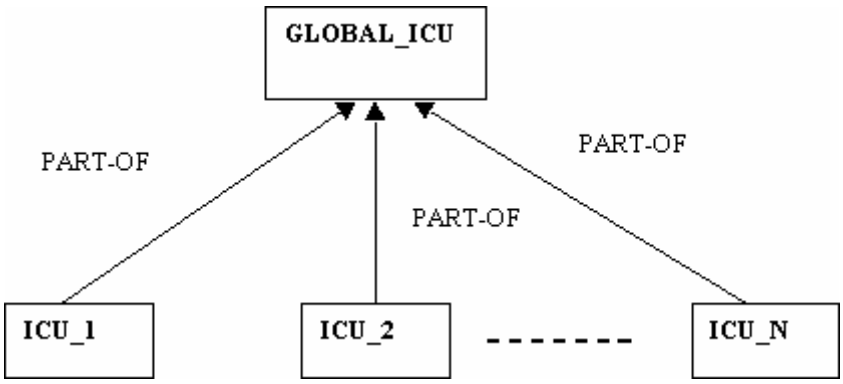


**Figura 5.11. Ontología para más de una UCI**

*5.3.2.5 Una extensión temporal al modelo ontológico*

El uso de una relación temporal para medir retardos entre las ocurrencias de eventos hace necesaria una extensión del modelo ontológico usado hasta el momento para incluir una nueva relación, la relación temporal. Por lo tanto se deberá añadir nuevas funciones al entorno presentado en los capítulos 2 y 3. Algunas de las funciones que presentamos a continuación son refinamientos de las especificadas en los capítulos 2 y 3.

**Función TEMHRD(t).** El dominio jerárquico, temporal, múltiple y restringido hasta el instante t, denotado por *TEMHRD(t),* se define como el conjunto de conceptos tal que están interasociados en múltiples jerarquías por medio de relaciones temporales. Formalmente, se define TEMHRD(t) como el conjunto de conceptos que cumple que:

[Card(TEMHRD(t)) ≥ 2] ⇔ [for all $c_i(t) \in$ TEMHRD(t) exists $c_j(t) \in$ TEMHRD(t), $c_i(t) \neq c_j(t)$, s. t. AFTER($c_i(t)$, $c_j(t)$,d(t)) v AFTER($c_j(t)$, $c_i(t)$,d(t))], donde AFTER(a, b, d(t)) representa que 'a ocurre después b, con una diferencia temporal d(t)', donde d(t) es un número difuso que representa el retardos entre los conceptos de la relación.

**Función TE-parents/TE-children**. Sea $c_i(t)$ un concepto perteneciente a un TEMHRD(t) no vacío. Podemos definir el *conjunto de padres temporales de $c_i(t)$ hasta el instante t*, denotado por *TE-parents ($c_i(t)$)* como el conjunto {$c_j(t) \in$ TEMHRD(t) s. t. AFTER($c_i(t)$, $c_j(t)$,d(t))}, donde d(t) es un número borroso que representa el retardo entre los conceptos de la relación. La function inversa para conceptos hijo es la siguiente:

El conjunto de *conceptos hijos temporales de $c_i(t)$ hasta el instante t*, denotado por *TE-children($c_i(t)$),* se define como el conjunto {$c_k(t) \in$ TEMHRD(t) s. t. AFTER($c_k(t)$, $c_i(t)$,d(t))}, donde d(t) es un número borroso que representa el retardo entre los conceptos de la relación.

**Función PMHRD(t)**. Sea TMHRD(t) un dominio jerárquico, múltiple, taxonómico, restringido hasta el instante t, sea $M^2$HRD(t) un dominio jerárquico, múltiple, mereológico, restringido hasta el instante t y sea TEMHRD(t) un dominio jerárquico, múltiple, temporal, restringido hasta el instante t. El *dominio jerárquico, múltiple, parcial, restringido hasta el instante t*, denotado por *PMHRD(t),* se define como el conjunto de conceptos resultado de la unión TMHRD(t) $\cup$ $M^2$HRD(t) $\cup$ TEMHRD(t).

**Conceptos temporalmente inconsistentes.** Sean $PMHRD_i(t)$ y $PMHRD_j(t)$ dos PMHRD(t); y sean let c(t) y c'(t) dos conceptos tales que c(t) $\in$ $PMHRD_i(t)$ y c'(t) $\in$ $PMHRD_j(t)$. Se dice que c(t) y c'(t) son *temporalmente inconsistentes*, denotado por TIC(c(t),c'(t)), sí y sólo si:

exists c''(t) $\in$ $PMHRD_i(t)$, exists c'''(t) $\in$ $PMHRD_j(t)$ s.t. (NAME(c(t))=NAME(c'(t))) $\wedge$ (NAME(c''(t))=NAME(c'''(t))) $\wedge$ [ (c(t) $\in$ TE-parents(c''(t)) $\wedge$ c'''(t) $\in$ TE-parents(c'(t))) $\vee$ (c(t) $\in$ TE-children(c''(t)) $\wedge$ c'''(t) $\in$ TE-children(c'(t))) ]

**Ontologías temporalmente inconsistentes.** Sean $PMHRD_i(t)$ y $PMHRD_j(t)$ dos PMHRD(t) correspondientes a las ontologías $O_i(t)$ y $O_j(t)$, respectivamente, consideradas en el instante t. Se dice que $O_i(t)$ y $O_j(t)$ son dos *ontologías temporalmente inconsistentes*, denotado por *TIO* $(O_i(t), O_j(t))$, sí y sólo si:

[exists c(t) in $PMHRD_i(t)$, exists c'(t) in $PMHRD_j(t)$ s. t. TIC(c(t), c'(t))]

Se utiliza la siguiente función para decidir si dos conjuntos de conceptos son temporalmente compatibles:

**Conjuntos temporalmente compatibles.** Sean $C_i$ y $C_j$ dos conjuntos de conceptos. Se dice que $C_i$ y $C_j$ son *conjuntos temporalmente compatibles*, denotado por *TCS($C_i$, $C_j$)*, sí y sólo si:

[for all c(t) in $PMHRD_i$(t), not exists c'(t) in $PMHRD_j$(t) s. t. TIC(c(t), c'(t))]

**Conceptos temporalmente equivalentes.** Podría ser interesante conocer si dos conceptos son temporalmente equivalentes o no. Formalmente:

Sean $PMHRD_i$(t) y $PMHRD_j$(t) dos PMHRD(t); y sean c(t) y c'(t) dos conceptos tales que c(t) $\in$ $PMHRD_i$(t) y c'(t) $\in$ $PMHRD_j$(t). Se dice que c(t) y c'(t) son *temporalmente equivalentes* , denotado por *TEC(c(t),c'(t))*, sí y sólo si:

not (TIC(c(t),c'(t))) $\wedge$ EA_equivalency(TE-parents(c(t), TE-parents(c'(t)) $\wedge$

EA_equivalency(TE-children(c(t), TE-children(c'(t))

**Conceptos incompatibles.** Dos conceptos distintos son incompatibles si existe alguna inconsistencia basada en atributos, basada en la estructura, o temporal entre ellos. Esta idea se puede expresar formalmente de la siguiente manera:

Sean $PMHRD_i$(t) y $PMHRD_j$(t) dos PMHRD(t); y sean c(t) y c'(t) dos conceptos tales que c(t) $\in$ $PMHRD_i$(t) y c'(t) $\in$ $PMHRD_j$(t). Se dice que c(t) y c'(t) son *conceptos incompatibles*, denotado por *incompatible_concepts(c(t),c'(t))*, sí y sólo si:

A_inconsistency(c(t),c'(t)) or OS_inconsistency(c(t), c'(t)) or TIC(c(t), c'(t))

**Ontologías equivalentes.** Dos ontologías son equivalentes si para todos los conceptos de cada una de ellas existe un concepto equivalente (desde el punto de vista de los atributos, de la estructura y temporal) en la otra ontología. Formalmente:

Sean $PMHRD_i(t)$ y $PMHRD_j(t)$ dos PMHRD(t) correspondientes a las ontologías $O_i(t)$ y $O_j(t)$, respectivamente, consideradas en el instante t. Se dice que $O_i(t)$ y $O_j(t)$ son *ontologías equivalentes*, denotado por *equivalent_ontologies* $(O_i(t), O_j(t))$, sí y sólo si:

[for all c(t) in $PMHRD_i(t)$, exists c'(t) in $PMHRD_j(t)$ s. t. syn_concepts(c(t), c'(t)) and TEC(c(t), c'(t))]

and

[for all c(t) in $PMHRD_j(t)$, exists c'(t) in $PMHRD_i(t)$ s. t syn_concepts(c(t), c'(t)) and TEC(c(t), c'(t))]

**Ontologías compatibles.** Dos ontologías son compatibles si no son equivalentes ni inconsistentes. Formalmente:

Sean $O_i(t)$ y $O_j(t)$ dos ontologías. Se dice que $O_i(t)$ y $O_j(t)$ son *ontologías compatibles*, denotado por *compatible_ontologies* $(O_i(t),O_j(t))$, sí y sólo si:

not (equivalent_ontologies($O_i(t)$,$O_j(t)$)) and not (inconsistent_ontologies ($O_i(t)$,$O_j(t)$)) and not (TIO($O_i(t)$,$O_j(t)$)).

### *5.3.2.6 Representación Ontológica de Ciclos*

Un ciclo puede ser visto como una subontología cuyo concepto raíz es el nombre del ciclo y cada evento individual es un hijo mereológico del concepto raíz. Sin embargo, debemos introducir algunos supuestos y propiedades relativas a estos ciclos.

- Un ciclo es un nodo hoja de la ontología que especifica una hipótesis diagnóstica.
- Los eventos incluidos en un ciclo no pueden aparecer en el resto de la hipótesis diagnóstica.
- Los eventos incluidos en un ciclo deben ser independientes de los eventos que no pertenecen al mismo ciclo.

Un ciclo puede definirse más formalmente como sigue: Un ciclo Cy(t) es un conjunto de conceptos (eventos) $\{c_0(t), c_1(t),\ldots\ldots, c_n(t)\}$, que cumplen las siguientes propiedades:

- Los eventos incluidos en el ciclo no pueden aparecer en el resto de la hipótesis diagnóstica.
- Sólo pueden existir relaciones temporales entre $c_i(t)$ y $c_j(t)$ $i>0, j>0, i\neq j$.
- $c_0(t)$ es la raíz del ciclo y es el padre mereológico del resto de eventos del ciclo.
- Un ciclo es un nodo hoja de la ontología que especifica una hipótesis diagnóstica.

Dado que un ciclo es visto como una ontología, debemos definir funciones que calculen la equivalencia y la consistencia entre ciclos.

*Cycle_Equivalency*: Sean $cycle_1(t)$ y $cycle_2(t)$ dos ciclos. Entonces, se dice que son equivalentes, denotado por *cycle_equivalency ($cycle_1(t)$, $cycle_2(t)$)* si equivalent_ontologies ($cycle_1(t)$, $cycle_2(t)$).

*Cycle_Consistency*: Sean $cycle_1(t)$ y $cycle_2(t)$ dos ciclos. Se dice que son consistentes, denotado por *cycle_consistency ($cycle_1(t)$, $cycle_2(t)$)* si not (inconsistent_ontologies ($cycle_1(t)$, $cycle_2(t)$)).

Por consiguiente, cuando se comprueba si dos ontologías son consistentes o equivalentes, al tratar un concepto que sea un ciclo se deberá aplicar la correspondiente función para ciclos.

### *5.3.2.7 Consistencia basada en Ontologías*

A continuación se describe el algoritmo para calcular la consistencia entre la evolución temporal del paciente y la hipótesis diagnóstica, que es empleado por el entorno de integración y por la extensión temporal presentada en este capítulo. En primer lugar, introduciremos un ejemplo que nos servirá para ilustrar el proceso. Supongamos que la hipótesis diagnóstica viene dada por la Tabla 5.14. La hipótesis se compone de un conjunto de eventos individuales $X_0$, $X_1$, $X_2$, y $X_3$, y de dos ciclos, $Cycle_1$ y $Cycle_2$. Además, la hipótesis diagnóstica se formula de forma que los miembros (eventos) del conjunto $\{X_1, Cycle_1\}$ deben aparecer obligatoriamente en la evolución de los pacientes para la patología en cuestión.

| Evento | Tiempo |
|---|---|
| $X_0$ | 0 minutes |
| $X_1$ | Approx 5 minutes after $X_0$ |
| $X_2$ | Approx 3 minutos after $X_0$ |
| $X_3$ | Approx 10 minutes after $X_1$ |
| $Cycle_1$ | Approx 2 minutes after $X_1$ |
| $Cycle_2$ | Approx 3 minutes alter $X_2$ |

**Tabla 5.14. La hipótesis diagnóstica**

| Evento | Tiempo |
|---|---|
| $X_0$ | 0 minutes |
| $X_2$ | 2 minutes |
| $X_1$ | 4 minutes 20 seconds |
| $E_1$ | 5 minutes 50 seconds |
| $E_3$ | 7 minutes 25 seconds |
| $E_2$ | 7 minutes 37 seconds |
| $E_4$ | 8 minutes |
| $X_3$ | 14 minutes 45 seconds |
| $E_1$ | 15 minutes 13 seconds |
| $E_2$ | 16 minutes 7 seconds |

**Tabla 5.15. La evolución del paciente**

Cada ciclo se define de la siguiente forma. $Cycle_1$ se compone de los eventos $E_1$ y $E_2$, de forma que $E_2$ ocurre después de $E_1$. Por su parte, $Cycle_2$ se compone de los eventos $E_3$ y $E_4$, ocurriendo $E_4$ después de $E_3$. No existe ningún retardo cuantificado entre $E_1$ y $E_2$, ni entre $E_3$ y $E_4$. Supongamos que hay un paciente conectado a nuestro sistema mediante un sistema de alarmas inteligentes. Por otro lado, la evolución temporal del paciente se puede definir como la secuencia de eventos mostrada en la Tabla 5.15.

Paso 1: Buscar la existencia de ciclos en la hipótesis diagnóstica. En este paso se chequea la ontología que representa la hipótesis diagnóstica para conocer si contiene ciclos o no. En caso de que no existan ciclos en la hipótesis diagnóstica, entonces iremos al paso 4 de este algoritmo. La entrada de este paso es la ontología que representa la hipótesis diagnóstica, y la salida es un valor booleano que indica si el algoritmo debe pasar al paso 2 ó al 4.

En nuestro ejemplo, este paso nos sirve para conocer si la hipótesis contiene ciclos. Como se puede observar en la Figura 5.15, la hipótesis contiene dos ciclos, $Cycle_1$ y $Cycle_2$, por lo que pasaremos al paso 2 del algoritmo.

Paso 2: Identificación de ciclos. En este paso, buscaremos en la evolución temporal del paciente las ocurrencias de los ciclos incluidos en la hipótesis. Para ello, se identifican individualmente los eventos de los ciclos en la ontología del paciente. Una vez que se han encontrado todos los eventos individuales del ciclo se comprueba la consistencia temporal interna de este ciclo. Si el ciclo no es consistente, entonces las dos ontologías se marcan como inconsistentes, por lo que el algoritmo terminaría. Este paso tiene como entradas la ontología que representa la evolución temporal del paciente y la hipótesis diagnóstica. La salida es un conjunto de ciclos incluidos en la hipótesis e identificados en la evolución temporal del paciente.

Al analizar el ejemplo mostrado en la Tabla 5.15 que representa la evolución temporal del paciente en cuestión, se encuentran los eventos individuales de ambos ciclos. $Cycle_1$ ocurre en dos ocasiones durante el período de tiempo cubierto por este ejemplo, mientras que $Cycle_2$ sólo lo hace una vez. En primer lugar se identificarían todas las ocurrencias de $Cycle_1$. Para ello se buscan sus eventos individuales. Encontramos una ocurrencia de $E_1$ en el instante 5 minutos 50 segundos, y una de $E_2$ en 7 minutos 37 segundos, por lo que se cumple la restricción temporal asociada al $Cycle_1$, esto es, que $E_2$ ocurre después de $E_1$. La segunda ocurrencia de $E_1$ se da en 15 minutos 13 segundos, y la segunda de $E_2$ en 16 minutos 7 segundos. Por otro lado, también se da $Cycle_2$ en la evolución del paciente. Sus eventos individuales los podemos encontrar en 7 minutos 25 segundos ($E_3$) y 8 minutos ($E_4$).

Paso 3: Inclusión de ciclos. El sistema de alarmas no es capaz de detectar ciclos, sino de avisar sobre eventos individuales, que pueden pertenecer o no a un ciclo. El paso anterior se encarga de detectar la existencia de los ciclos en la evolución del paciente. Los eventos individuales identificados como parte de ciclos de la hipótesis son reemplazados por los ciclos en la evolución del paciente. Cada evento individual tiene asignado un tiempo de ocurrencia, de forma que el ciclo que reemplaza al conjunto de eventos tiene que tener asignado un tiempo de ocurrencia. En particular, se le asignará el tiempo de ocurrencia del primer evento incluido en el ciclo. Cuando se identifica el mismo ciclo más de una vez en la evolución del paciente, el ciclo aparecerá sólo una vez, y se eliminará de la evolución del paciente el resto de ocurrencias. Este paso tiene como entradas las mismas que el paso anterior, además de la salida del paso 2. Su

salida es una nueva ontología que representa la evolución temporal del paciente en la cual ya se han incluido los ciclos.

Ejemplo: En nuestro ejemplo, la ontología que representa la evolución del paciente sería transformada en la que aparece en la Figura 5.12.

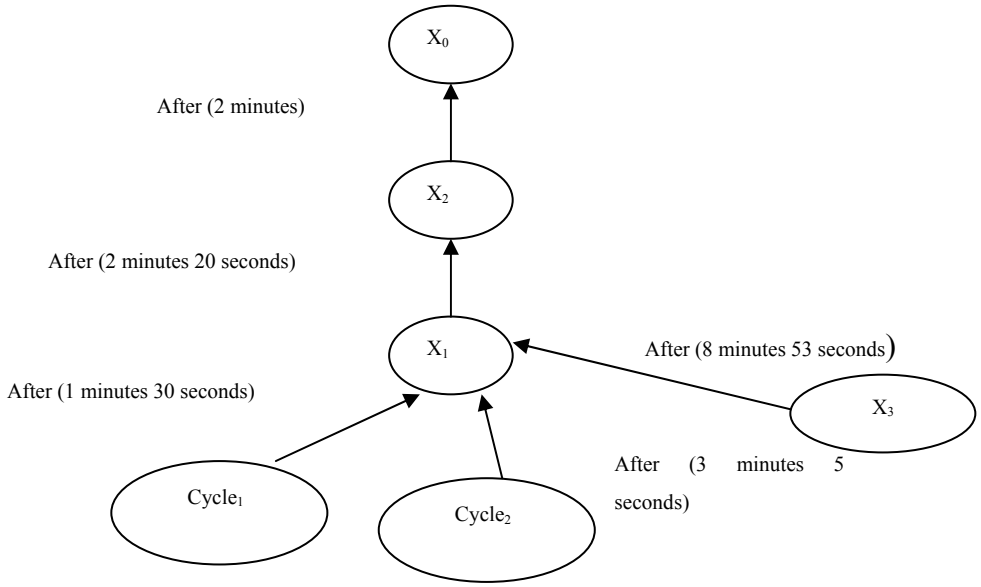


**Figura 5.12. La ontología una vez incluidos los ciclos**

Paso 4: Chequear la consistencia entre la evolución del paciente y la hipótesis diagnóstica. Para ello, se usa el mismo procedimiento empleado en el método sin ciclos. En este paso, un ciclo será considerado como un nodo simple puesto que la consistencia sólo tiene en cuenta las relaciones temporales y los tiempos de ocurrencia de cada evento/ciclo. Entonces, tendremos dos ontologías que representan la evolución del paciente y la hipótesis diagnóstica respectivamente. Asimismo, se comprobará la consistencia de ambas ontologías haciendo uso de la función consistent_ontologies.

Ejemplo: En este paso se debe comprobar la consistencia entre la hipótesis diagnóstica y la evolución del paciente. Para ello, podemos ver en las Tablas 5.16-5.17 las cuantificaciones temporales de los eventos pertenecientes a las correspondientes cadenas temporales. Ambas ontologías son consistentes desde un punto de vista temporal puesto que el tiempo de

ocurrencia de cada evento del paciente pertenece al intervalo de tiempo admisible para tal evento en la hipótesis. Por lo tanto, de momento, el paciente puede ser diagnosticado la enfermedad especificada por la hipótesis.

Paso 5: Finalmente, se deben comprobar los criterios relacionados con los eventos. En este ejemplo, se ha de aplicar a la evolución del paciente el criterio de que $X_1$ y $Cycle_1$ son eventos que deben ocurrir al paciente.

| Hipótesis | Tiempo borroso de ocurrencia |
|---|---|
| $X_0$ | 0 minutes |
| $X_2$ | Approx 3 minutes |
| $X_1$ | Approx. 5 minutes |
| $Cycle_2$ | Approx 6 minutes |
| $Cycle_1$ | Approx 7 minutes |
| $X_3$ | Approx 15 minutes |

**Tabla 5.16. La hipótesis**

| Evolución del paciente | Tiempo exacto de ocurrencia |
|---|---|
| $X_0$ | 0 minutes |
| $X_2$ | 2 minutes |
| $X_1$ | 4 minutes, 20 seconds |
| $Cycle_1$ | 5 minutes, 50 seconds |
| $Cycle_2$ | 7 minutes, 25 seconds |
| $X_3$ | 13 minutes, 15 seconds |

**Tabla 5.17. La evolución del paciente**

## 5.4 RESUMEN

En este capítulo se ha evaluado el entorno de integración de conocimiento siguiendo el siguiente proceso de validación. En primer lugar, se validó el entorno de integración mediante la medición y el análisis del conocimiento contenido en las ontologías privadas y en la integrada. A continuación, se compararon los resultados obtenidos en diferentes dominios. Esta tarea requiere definir una serie de parámetros ontológicos calculables, que sirven para cuantificar el conocimiento asociado a cada ontología.

En primer lugar se realizaron dos experimentos independientes. En el primero se escogieron tres dominios de aplicación. Se solicitó a un conjunto de estudiantes de pre y postgrado que entrevistaran a diferentes expertos, y construyeran ontologías del dominio a partir de tales entrevistas, haciendo uso del marco de integración propuesto. El objetivo era analizar los beneficios proporcionados por los procesos de integración de acuerdo al conocimiento ganado por los constructores de ontologías. Con el objeto de medir tales ganancias de conocimiento relativas se han establecido una serie de indicadores que están relacionados con la cantidad de entidades de conocimiento especificadas para cada categoría de conocimiento, esto es, conceptos, atributos y relaciones, y el tamaño de las ontologías.

En primer lugar, se calculó una ganancia local, esto es, el conocimiento ganado por una categoría de conocimiento para cada ontología. A continuación se extendió dicha medida de ganancia a un conjunto de ontologías, realizando una medición por cada categoría de conocimiento. Finalmente, se definió una medida de ganancia global para un conjunto de ontologías y un conjunto de categorías de conocimiento. De los resultados presentados en esta sección se concluyó que los procesos de integración son beneficiosos para los trabajadores de conocimiento debido a que las ontologías obtenidas mediante procesos de integración contienen más conocimiento que las ontologías individuales. El valor de ganancia global de conocimiento puede indicar diferentes cosas. En caso de una ganancia de conocimiento alta, estaríamos ante el caso en el que hay poco solapamiento entre las ontologías a integrar. Este bajo solapamiento se puede deber a que (1) los expertos hayan especificado partes diferentes del dominio, o (2) el entorno de integración no haya sido capaz de detectar la existencia de conceptos sinónimos.

Cuando tenemos una ganancia de conocimiento baja, esto implica que las ontologías se solapan significativamente, bien por las propias ontologías o por la detección de conceptos sinónimos. En ambos casos significaría un elevado grado de acuerdo entre los expertos.

También se ha realizado un análisis por categorías de conocimiento. De los experimentos realizados, se extrae que es más fácil estar de acuerdo sobre los conceptos que sobre los atributos. Los resultados sobre relaciones son más cercanos a los obtenidos para conceptos. Este hecho no es sorprendente puesto que, en nuestro modelo ontológico, las relaciones se dan entre conceptos. También se puede realizar un análisis según la naturaleza de los dominios de aplicación, puesto que se ha observado que se obtuvieron resultados semejantes en dominios similares.

Para el segundo experimento se solicitó a un grupo de expertos que construyeran ontologías individuales sobre el dominio de los ordenadores, y se realizaron dos procesos de integración. Las ontologías usadas en el segundo proceso de integración fueron versiones refinadas de las usadas en la primera iteración. El propósito de este experimento era analizar los efectos de los procesos de integración en los procesos de refinamiento de ontologías y cómo pueden ayudar para alcanzar el consenso entre los expertos.

El segundo objetivo de este capítulo era evaluar la calidad de las ontologías construidas. Con tal propósito se seleccionaron cuatro dominios heterogéneos, y se construyeron ontologías de forma cooperativa gracias a la participación de expertos. Después de concluir el proceso de construcción, se evaluó la calidad de las ontologías. Para ello se usaron los criterios de calidad mencionados en (Crow and Shadbolt, 1999), a saber, precisión, anchura, consistencia, completitud, y utilidad. Se mostró que las ontologías construidas usando el modelo ontológico y el entorno de integración cumplen todos estos criterios, de forma que se puede decir que el marco propuesto en esta tesis genera ontologías que cumplen tales criterios de calidad.

Para concluir la validación propiamente dicha, podemos extraer varias conclusiones de estos experimentos. En primer lugar, la interacción con los expertos facilita la focalización de contenidos. Esta interacción permite entender mejor el dominio y evita la aparición de atributos

o relaciones taxonómicas espúreas. En segundo lugar, los buenos resultados obtenidos en esta validación constituyen una justificación parcial de la consecución de los objetivos iniciales.

La segunda parte de la validación se corresponde con la descripción de distintas aplicaciones del entorno de integración. Se han presentado dos aplicaciones en esta parte del capítulo. El primero corresponde a la construcción cooperativa de ontologías en biología molecular, que demuestra cómo se ha usado el entorno de integración para construir dos ontologías parciales diferentes, una relativa a la célula y otra a la proteína. Estas ontologías fueron construidas por expertos con el objetivo de validar la utilidad de los procesos de integración en procesos de construcción de ontologías. La segunda aplicación del modelo ontológico y del entorno de integración fue una parte de un entorno para facilitar la gestión de conocimiento en Unidades de Cuidados Intensivos (UCIs).

# CAPÍTULO 6

# UNA APLICACIÓN SOFTWARE PARA FACILITAR LA INTEGRACIÓN Y GESTIÓN DE CONOCIMIENTO

# Capítulo 6. Una Aplicación Software para Facilitar la Integración y Gestión de Conocimiento

El objetivo general de esta aplicación era desarrollar un sistema que pudiera servir como entorno para la creación co-operativa de ontologías a través de procesos de integración. Además, se perseguía una meta más ambiciosa, que cada usuario se beneficiara de las aportaciones previas que otros usuarios habían realizado. En otras palabras, el sistema también fue diseñado para facilitar la compartición de conocimiento. El punto de partida de la herramienta es un conjunto de usuarios que trabajan en un intranet/internet y que pretenden construir ontologías de forma cooperativa. Sin embargo, esta cooperación es totalmente transparente para cada usuario, porque no saben cuando otros usuarios están usando ontologías creadas por usuarios diferentes. Es más, ningún usuario puede modificar ontologías construidas por otros usuarios, y ningún usuario tiene permitido el acceso directo a las ontologías de otros usuarios. El acceso al conocimiento de otros usuarios es realizado a través de los procesos de integración.

La herramienta software puede ser usada por aquellas personas de la comunidad de Adquisición de Conocimiento interesadas en representar el conocimiento por medio de las ontologías y por aquéllos que deseen compartir su conocimiento ontológico con el resto de usuarios. En el sistema, pueden distinguirse tres tipos de usuarios:

- Usuario normal
- Usuario experto
- Administrador

Un usuario normal es visto aquí como un consultor de conocimiento, una persona curiosa que quiere poner al día su conocimiento sobre un cierto tema en un momento particular. Por otro lado, un usuario experto es un constructor de conocimiento, encargado de construir

ontologías que serán consultadas por el resto de usuarios (normales o expertos) del sistema. Un tercer tipo de usuario es necesario para tomar las decisiones relativas a quién es un experto y quién es un usuario normal. Esta decisión se tomará en base a evaluar la información proporcionada por el usuario al registrarse en el sistema. El administrador del sistema posee el poder ejecutivo para esta decisión, ya que está al cargo de mantener y manejar el sistema.

## 6.1 EL SISTEMA CLIENTE SERVIDOR

El mecanismo para combinar el conocimiento es la integración de conocimiento a través de la integración de ontologías, porque se representa el conocimiento en esta tesis por medio de esta poderosa tecnología de representación de conocimiento. Por consiguiente, es necesario un sistema cuya arquitectura permita a los usuarios compartir el conocimiento de una manera transparente. Se ha desarrollado una infraestructura cliente-servidor de forma que cada usuario corresponde a un cliente y el motor de construcción de ontologías lo contiene el servidor. En otros términos, cada usuario sólo conoce la existencia del servidor, pero no pueden acceder a otro usuario. De esta forma, los usuarios tienen que hacer las peticiones de servicio al servidor (es decir, el acceso es exclusivamente vía servidor).

### 6.1.1 Arquitectura General

La solución adoptada ha sido usar una arquitectura cliente/servidor donde cada usuario es un cliente, y el sistema es el servidor. En otros términos, los usuarios sólo saben la existencia del servidor, y ellos no tienen el acceso a otros usuarios. Ellos sólo pueden interactuar con el sistema. Por consiguiente, el sistema puede ser dividido en dos: parte del cliente y parte del servidor.

*El Servidor de Ontologías*

En toda arquitectura cliente-servidor, el servidor es el núcleo del sistema y coordina las operaciones realizadas por los usuarios. El servidor almacena la información del sistema como los usuarios y bases de datos de los temas de trabajo. De igual forma, las ontologías proporcionadas por los usuarios también deben ser clasificadas por tema para usarlas en los procesos de integración correspondientes. Se realizarán los procesos de integración de conocimiento cuando un usuario, normal o experto, lo solicite. Podría pasar que, en ese momento en cuestión, varios expertos estén generando sus ontologías sobre ese mismo tema, por lo que el conocimiento almacenado por el sistema se convierta probablemente en obsoleto. Así, cuando la integración es solicitada por un usuario, el sistema debe actualizar su conocimiento en el tema a integrar, obteniendo las aportaciones de conocimiento más recientes de cada experto. Este mecanismo de actualización debe ser del mismo modo transparente para los usuarios. Finalmente, el proceso de integración desde un punto de vista funcional es descrito en la Figura 6.1. En primer lugar, una solicitud de integración de un cierto tema es realizada por un usuario. Entonces, el servidor se actualiza. A continuación se integran todas las ontologías disponibles sobre ese tema. Finalmente, los resultados de la integración se envían al usuario que solicitó la integración.

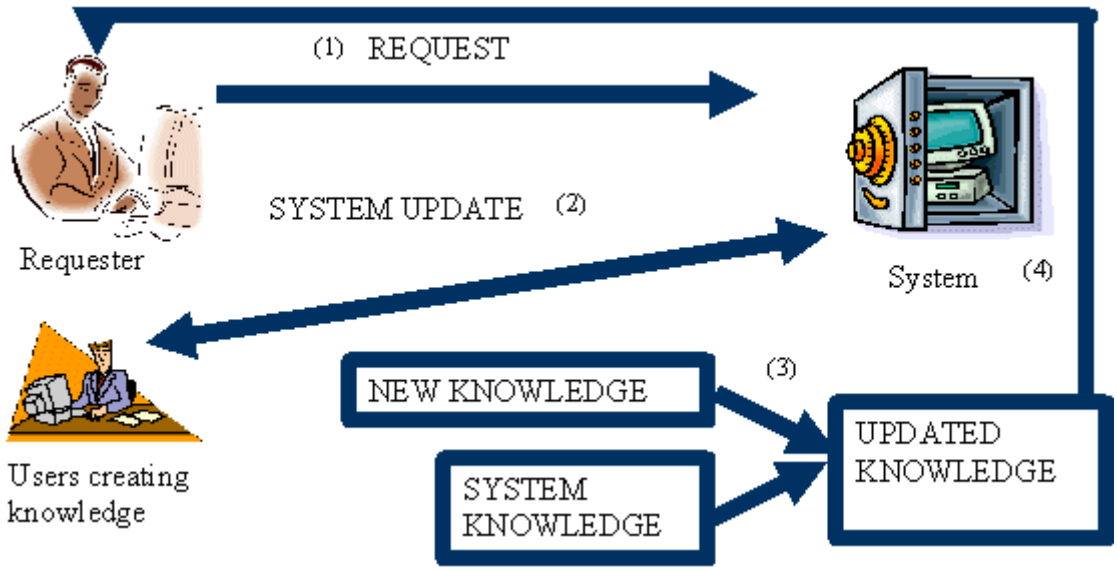


**Figura 6.1 Una visión del proceso de integración**

*Cliente*

El módulo cliente se personaliza según el tipo de usuario que accede al sistema, porque los expertos y los usuarios normales no tienen las mismas opciones de trabajo. Un usuario experto, una vez seleccionado un tema de trabajo, puede especificar el archivo activo para dicho tema. A través de esta selección, el sistema sabe que este usuario tiene potencialmente nuevo conocimiento. Los usuarios expertos pueden actualizar el servidor, bien utilizando la opción de actualización o bien especificando un archivo activo y que un usuario diferente solicite la integración de ese tema.

Los usuarios expertos tienen diferentes opciones que los usuarios normales cuando han especificado un fichero de trabajo. Por lo tanto, los usuarios expertos pueden tener dos modos de trabajo:

A. Modo consulta: El usuario experto no ha seleccionado fichero de trabajo. De esta forma, un usuario experto es equivalente a un usuario normal porque no puede modificar el sistema (no puede enviar una nueva ontología). En este modo, los usuarios expertos

sólo pueden solicitar la integración así como ver los resultados de procesos de integración.

B. Modo operativo: Cuando un usuario experto selecciona una ontología de trabajo se le comunica al servidor que dicho usuario posee conocimiento potencialmente nuevo, por lo que dicho conocimiento puede ser requerido por el servidor en cualquier momento y el experto en cuestión puede actualizar el conocimiento existente en el servidor.

## 6.1.2 Especificación de Ontologías

No se incluye ningún editor de ontologías en esta herramienta, por lo que las ontologías deben ser definidas usando herramientas externas y proporcionarlas al sistema por medio de archivos. Hay dos maneras principales para especificar las ontologías: (1) archivo en formato texto; y (2) editor de ontologías. El formato texto para especificar las ontologías ya fue explicado en el capítulo 2. Con respecto al uso de un editor de ontologías para definir las ontologías de una manera más cómoda, se ha desarrollado un prototipo de un editor en nuestro grupo de la investigación, facilitando la construcción gráfica de ontologías y soportando al modelo ontológico especificado en esta tesis. El modelo de ontología cubierto por este prototipo es una extensión del modelo ontológico propuesto aquí. La Figura 6.2 muestra una imagen del mencionado editor.

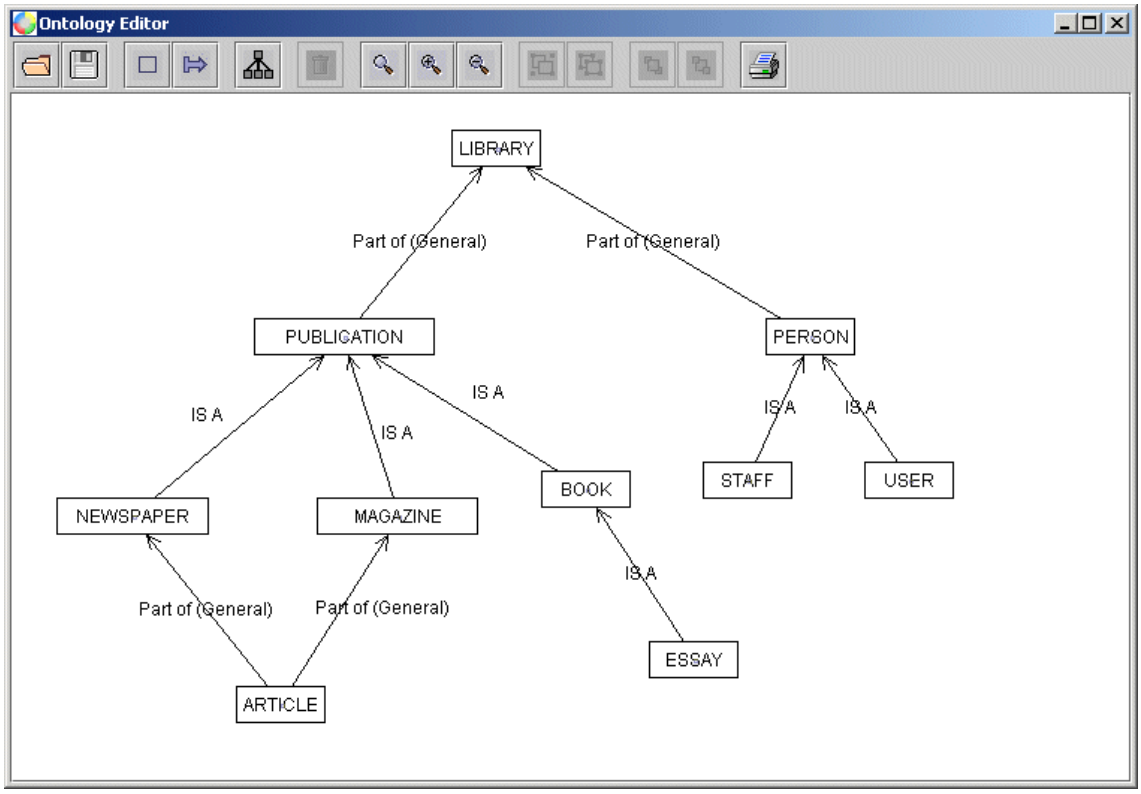


**Figura 6.2 Un editor gráfico para especificar ontologías**

## 6.2 UNA SESIÓN CON ONTOINT

En este apartado se mostrarán las facilidades para la integración de ontologías y gestión del sistema. En primer lugar se describirán las distintas opciones relacionadas con usuarios. Después se describirá una sesión completa para un usuario experto y para un usuario normal.

### 6.2.1 Gestión de Usuarios

En este apartado se describe como los usuarios se pueden dar de alta en el sistema o acceder al mismo. Asimismo, se presentan las opciones de gestión disponibles para el usuario administrador. La Figura 6.3 es la ventana principal de la aplicación. Al hacer clic en el menú Options, los usuarios pueden realizar tres operaciones:

- *Nuevo usuario*: Un nuevo usuario quiere dares de alta
- *Entrar*: Un usuario existente quiere trabajar
- *Salir*: Un usuario quiere abandonar el sistema

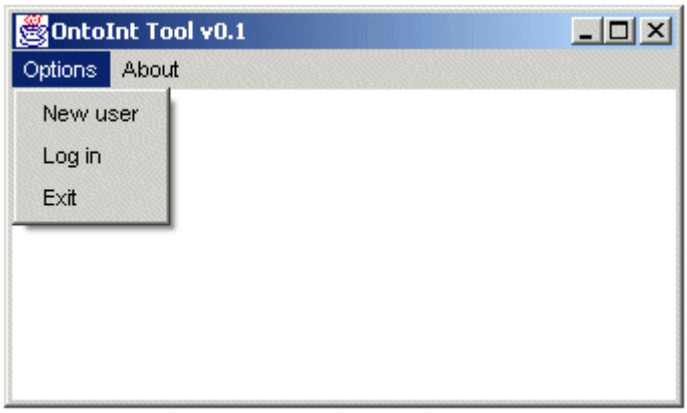


**Figura 6.3 Ventana principal de la aplicación**

Supongamos que un nuevo usuario quiere registrarse. En ese caso aparecerá un cuadro de diálogo (ver Figura 6.4). El nuevo usuario debe introducir algunos datos para ser un usuario de este sistema. Tal información incluye detalles personales y profesionales, como (1) el nombre de usuario; (2) la contraseña; (3) la institución, es decir, afiliación; (4) la dirección profesional; (5) la nacionalidad; (6) la fecha de nacimiento; (7) el lugar de nacimiento; (8) la motivación para usar el sistema; (9) la dirección del correo electrónico; y (10) el sitio web. Se ha dicho previamente que existen tres tipos de usuarios, a saber, los usuarios normales, usuarios expertos, y el administrador. Obviamente, este cuadro de diálogo es sólo para registrarse usuarios no administradores. Por defecto, cada nuevo usuario es un usuario normal y el administrador del sistema decidirá quién es un usuario experto analizando los datos proporcionados en el proceso de registro.

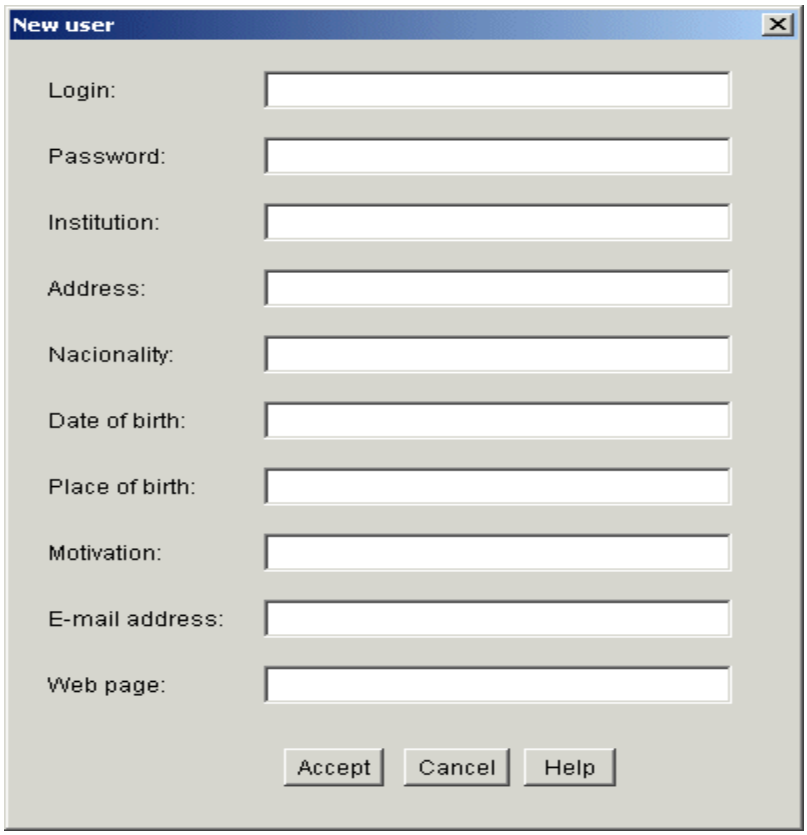


**Figura 6.4 Diálogo de registro**

Una vez que se ha registrado el nuevo usuario en el sistema vuelve a aparecer la ventana principal (Figura 6.3), por lo que el usuario puede acceder al sistema. Para ello, el usuario necesita introducir su nombre de usuario y contraseña en el correspondiente cuadro de diálogo (Figura 6.5).

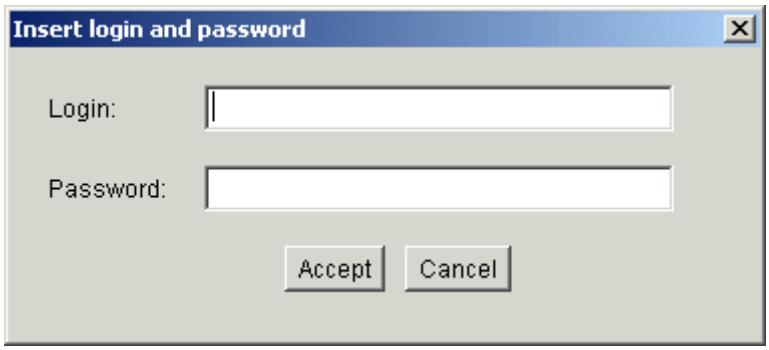


**Figura 6.5 Diálogo de acceso al sistema**

Supongamos ahora que el administrador del sistema ha entrado al sistema. La Figura 6.6 muestra las distintas opciones de trabajo que tiene.

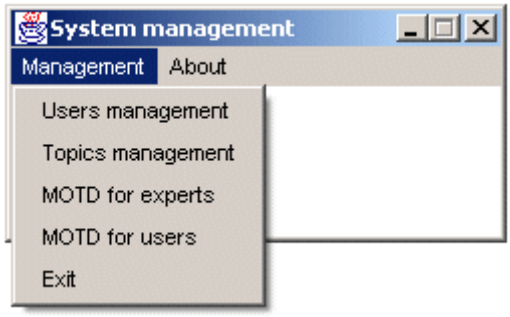


**Figura 6.6 Opciones para gestionar el sistema**

El administrador tiene las siguientes opciones de gestión: (1) gestión de usuarios; (2) gestión de temas de trabajo; (3) Mensaje del día para expertos (MOTD); (4) MOTD para usuarios normales; y (5) salida. A continuación se describen tales opciones.

Gestión de usuarios: El administrador del sistema puede acceder a los datos suministrados por los usuarios. Se pueden distinguir dos áreas en la Figura 6.7. La parte izquierda corresponde al diálogo de gestión de usuarios propiamente dicho, donde aparecen las siguientes opciones: (1) promoción, esto es, convertir un usuario normal en experto; (2) descenso, esto es, convertir un experto en usuario normal; (3) eliminar un usuario del sistema; y (4) ver los detalles de un usuario. Estos detalles de usuario son mostrados en la parte derecha de la Figura 6.7, que se diferencia de la ventana de registro de usuario en dos aspectos: (1) no es editable; y (2) se muestra la información sobre el tipo de usuario.

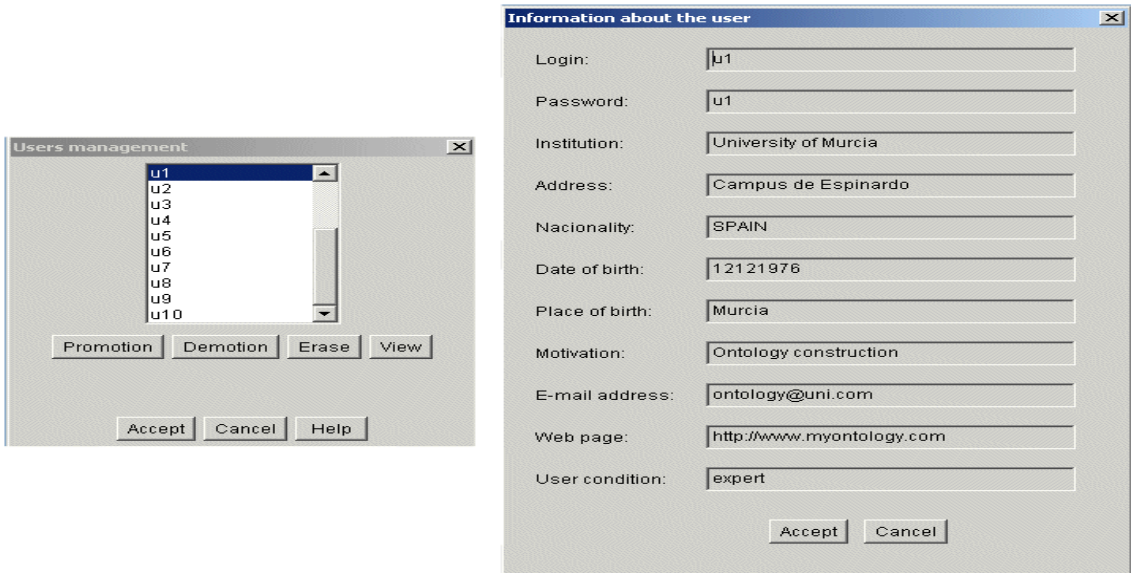


**Figura 6.7 Visión de los detalles de un usuario**

La siguiente opción es la gestión de temas de trabajo. La ventana correspondiente se muestra en la Figura 6.8. Aquí, el usuario puede gestionar los temas sobre los cuales los usuarios expertos pueden crear ontologías y aquellos temas (sugerencias) que todavía no son considerados como temas de trabajo pero sobre los cuales existen usuarios expertos interesados. La ventana se puede dividair a su vez en dos áreas. La parte superior contiene los temas y sugerencias actuales. El administrador puede mover de una categoría a otra usando "🡐" y "🡒". En la parte inferior, el administrador puede crear sus propios temas de trabajo y sugerencias a través de las opciones correspondientes.

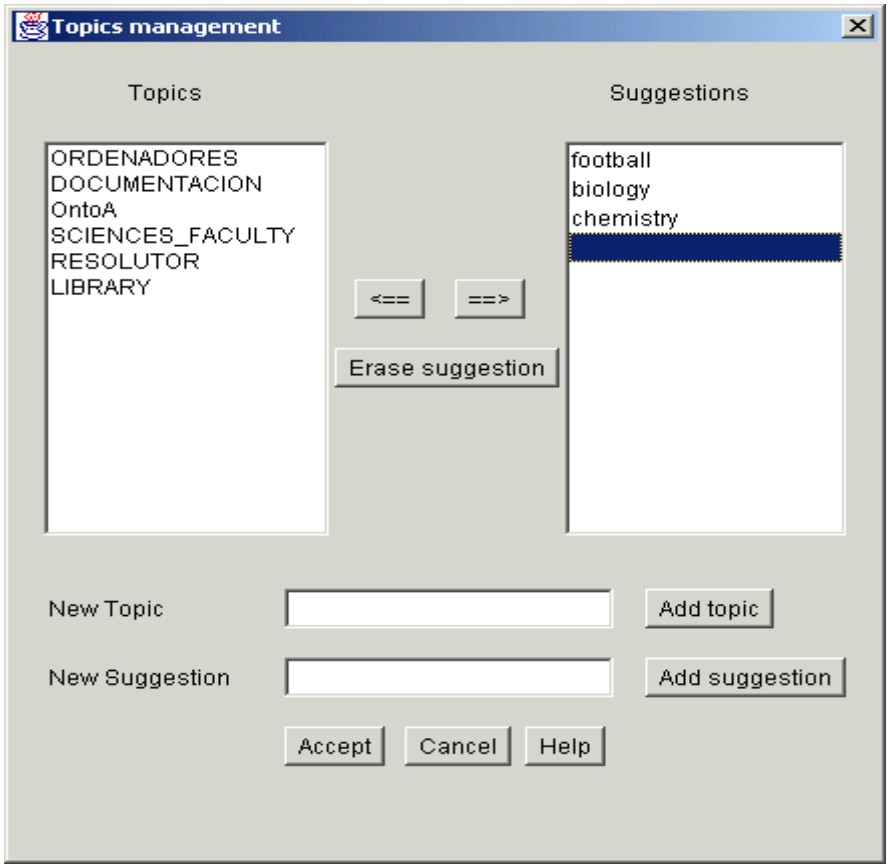


**Figura 6.8 Gestión de temas**

Finalmente, el administrador del sistema puede personalizar los mensajes que los expertos y los usuarios normales recibirán al comenzar una sesión de trabajo. Este mensaje, denominado "Message of the Day", es introducido en el diálogo mostrado en la Figura 6.9 y es diferente para usuarios expertos y normales. Sin embargo, la interfaz para el administrador es idéntica.

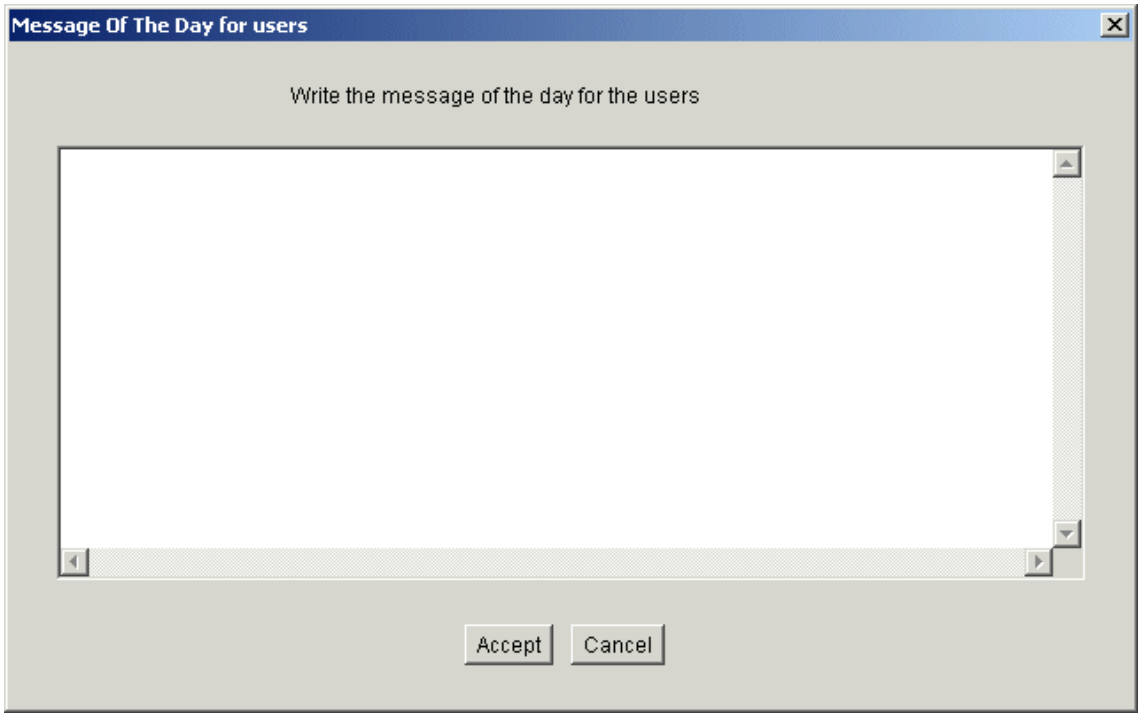


**Figura 6.9 Mensaje del Día**

## 6.2.2 Usuarios expertos y normales

Las opciones para expertos y usuarios son similares. En particular, las proporcionadas a los usuarios son un subconjunto de aquéllas proporcionadas a los usuarios expertos. En esta sección se presentan las opciones de los expertos y se señalará cuáles no están disponibles para los usuarios normales. Así, cuando un experto entra al sistema, se muestra el MOTD (ver Figura 6.10). A través de este mensaje, el administrador del sistema puede comunicarse con los usuarios, para promover que los usuarios trabajen en algún tema en particular o recibir información general sobre el sistema como actualizaciones, estadísticas, etc.

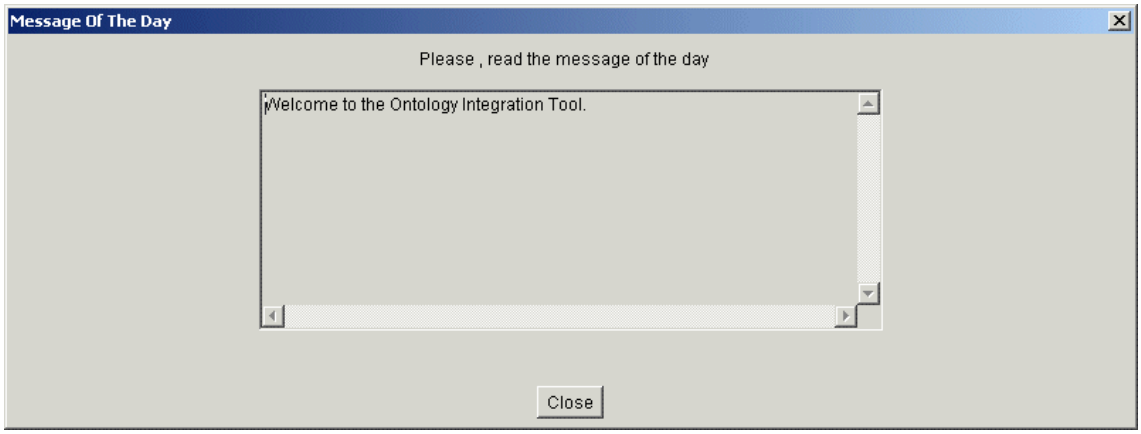


**Figura 6.10. Mensaje del día**

Las diferentes opciones para usuarios expertos se muestran en la Figura 6.11. Un usuario experto puede realizar las siguientes acciones: (1) solicitar la integración; (2) actualizar el servidor; (3) elegir un tema de trabajo; y (4) ver una ontología. En la Figura 6.11 podemos ver que sólo está activada la tercera opción. Esta situación ocurre cuando el experto entra al sistema porque todavía no ha elegido un tema de trabajo.

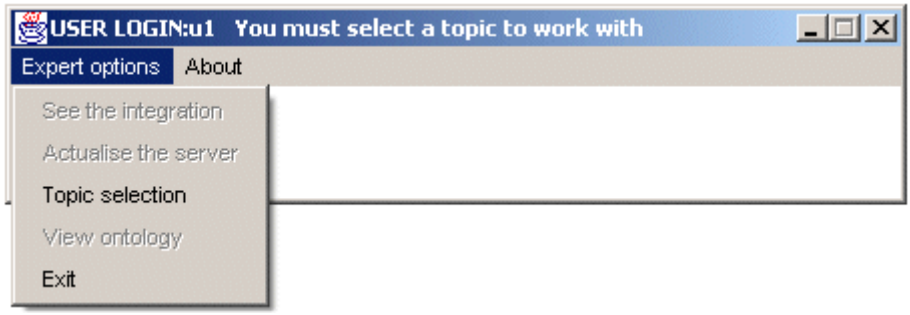


**Figura 6.11 Opciones para expertos**

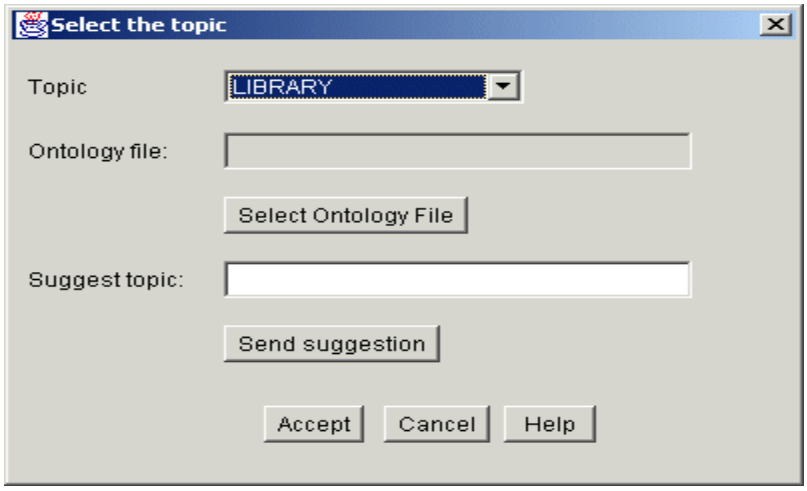


**Figura 6.12 Experto seleccionando tema de trabajo**

La primera acción que el experto debe realizar es seleccionar un tema de trabajo. Haciendo clic en la opción correspondiente, aparece la ventana mostrada en la Figura 6.12, que se divide en tres secciones principales. En la parte superior, el experto debe seleccionar el tema activo entre la lista de temas actualmente disponibles en el sistema. En la parte media, el experto puede escoger un fichero de trabajo, es decir, el archivo de su ontología actual en el tema seleccionado. Finalmente, el usuario experto también puede proponer un tema de trabajo al administrador del sistema haciendo el uso de la facilidad proporcionada en la interfaz. Por lo que respecta a usuarios normales, tienen un diálogo similar para seleccionar el tema de trabajo, pero la diferencia es que los usuarios normales no tienen la posibilidad de establecer un fichero de trabajo.

Supongamos que el experto ha seleccionado "LIBRARY" como tema y ha especificado un archivo como su ontología activa. El sistema mostraría las opciones al experto, pero ahora todas las opciones se habilitarían puesto que existe un tema y fichero de trabajo seleccionados. En caso de no haber seleccionado fichero de trabajo, la opción de actualización de servidor y la opción de ver ontología estarían deshabilitadas. Si el experto selecciona ver la ontología, entonces se mostraría el diálogo representado en la Figura 6.13. Allí, la ontología se muestra de manera jerárquica en la parte izquierda del interfaz. Se mostrarían todas las organizaciones taxonómicas y mereológicas existentes en tal jerarquía. Sin embargo, ambos tipos de

organizaciones se explorarán de forma diferente, debido a la noción de especialización que se introdujo en Capítulo 2 y que está asociada a relaciones taxonómicas.

Por consiguiente, el usuario tendrá que seleccionar el conjunto de atributos por que el cual quiere explorar la jerarquía taxonómica, esto es, expandir el nodo taxonómico actual. En la Figura 6.14 podemos ver un ejemplo de este tipo de selección. En particular, esta figura corresponde a la expansión del concepto "Persona", que tiene cuatro atributos por los que el concepto puede expandirse, "address", "name", "id", y "role". En este caso, el usuario ha seleccionado expandir por el atributo "role". En la parte derecha de la Figura 6.13 se muestra el conocimiento relacionado con el concepto actualmente seleccionado.

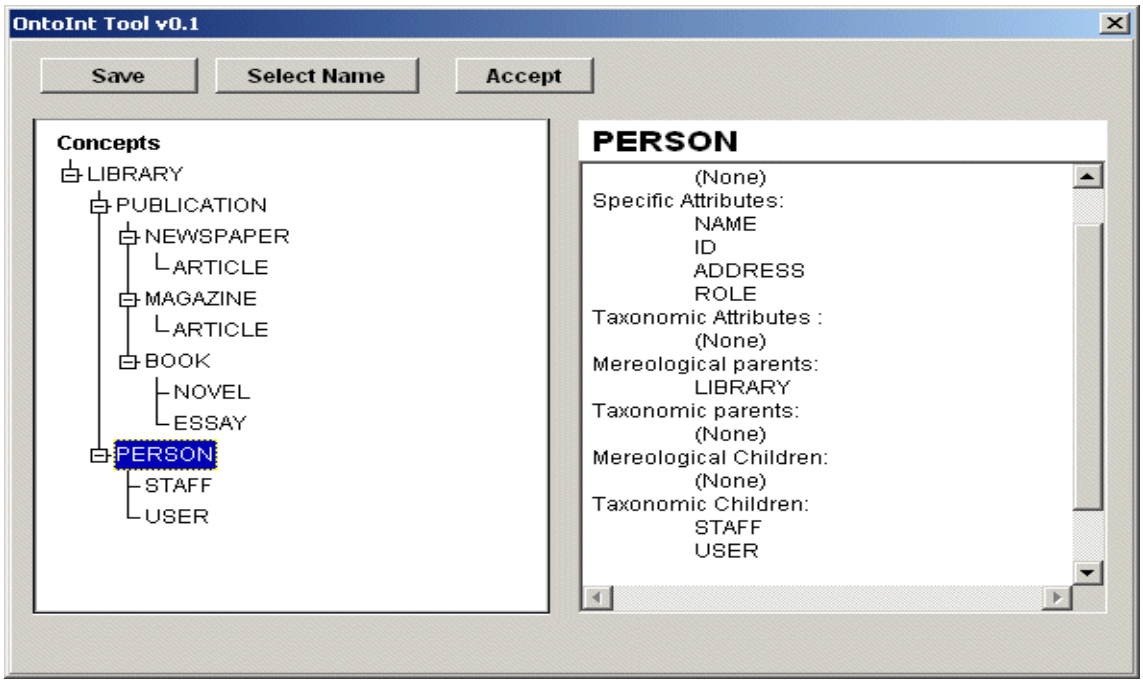


**Figura 6.13 Explorando una ontología**

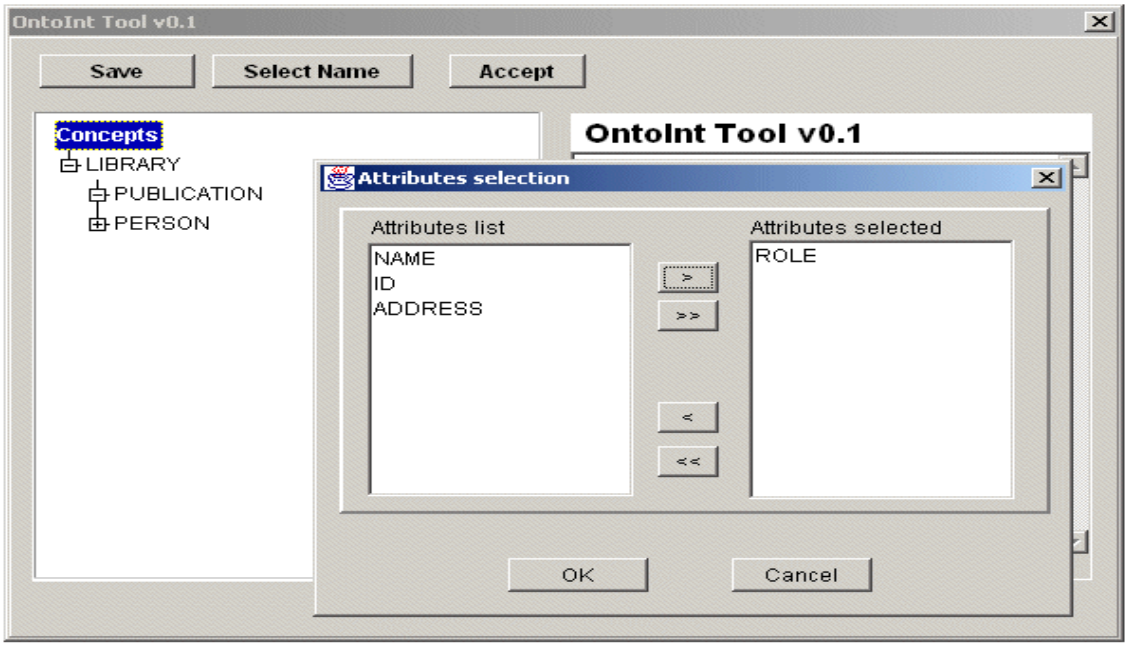


**Figura 6.14 Expansión del nodo taxonómico PERSON**

Tras ver la ontología, el usuario experto puede desear actualizar el servidor. La opción anterior, ver una ontología, no se permite a los usuarios normales. La opción de actualización de servidor tampoco se permite a este tipo de usuarios porque no pueden crear nuevas ontologías. Para actualizar el servidor, el usuario experto enviará su fichero de trabajo al sistema. La idea era construir un sistema de integración de ontologías que pudiera integrarse con otras herramientas ontológicas tales como editores de ontologías para poder contar con más opciones para actualizar el servidor. La última opción, solicitar la integración, está disponible para usuarios y expertos. Cuando el usuario solicita la integración, el sistema lanza el proceso correspondiente de integración, que se ejecuta en el lado del servidor, y los resultados se envían al usuario que solicitó la integración. Entonces, la ontología obtenida como resultado de integrar todas las ontologías guardadas por el sistema en el mismo tema será mostrada al usuario de dos formas distintas, como se muestra en la Figura 6.15.

En el fondo de la Figura 6.15 se puede ver una representación gráfica de la ontología, donde se muestran los conceptos y las relaciones directamente en la superficie de dibujo. El restante conocimiento asociado a los conceptos puede ser consultado haciendo clic sobre el

concepto específico. Los conceptos se representan por medio de rectángulos, y las relaciones por medio de las flechas coloreadas, las taxonómicas en azul y las mereológicas en rojo. El usuario experto podrá modificar esta ontología, por lo que podemos decir que los expertos pueden refinar directamente los resultados de los procesos de integración. En el primer plano de la figura se puede ver la ontología usando la misma herramienta de visualización usada para la opción "ver la ontología".

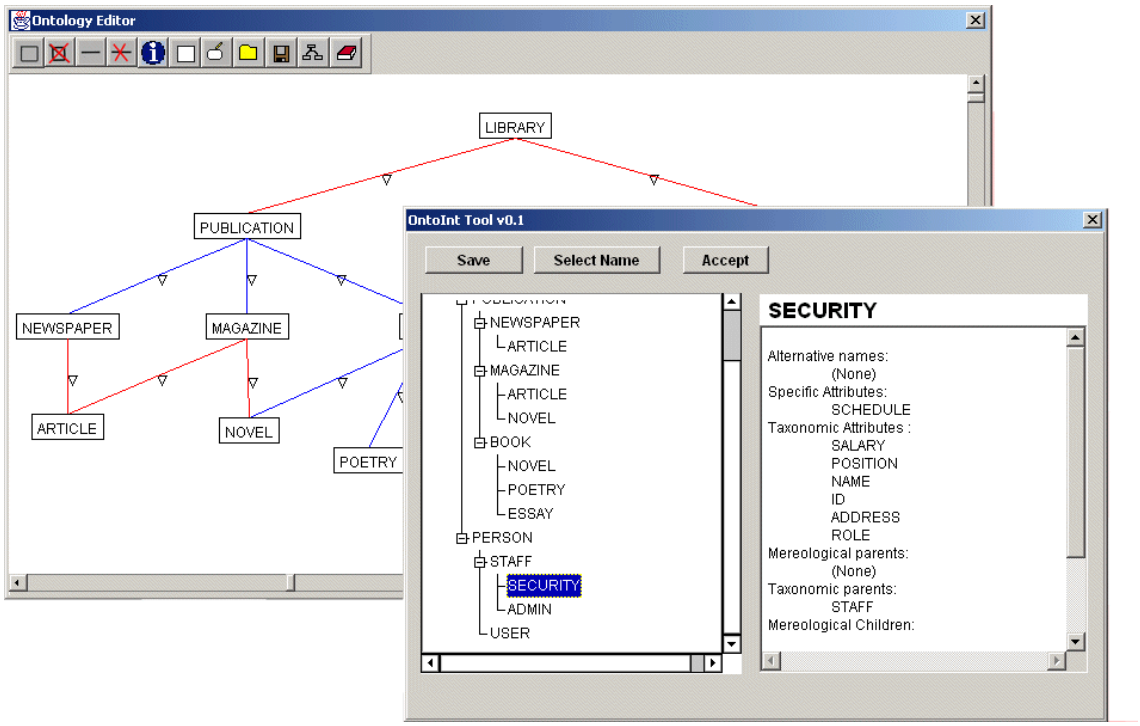


**Figura 6.15 Resultados del proceso de integración**

## 6.3 INTERFAZ WEB

Por otro lado, el servidor de integración de ontologías también puede ser accedido desde la Web, pues se ha desarrollado una interfaz con tal propósito. Los usuarios web tienen las mismas opciones que los usuarios no web. A continuación seguiremos un proceso descriptivo similar al apartado de la aplicación no web. En primer lugar se debe acceder a la página web del Integrador de Ontologías, donde se mostrarán diferentes opciones al usuario. Por un lado se muestra información general (descripción del integrator de ontologías, su propósito, sus creadores, etc) sobre el integrador. Estas son las opciones disponibles desde la página web del integrador:

- Alta de usuario: A través de esta opción, gente interesada en usar el sistema puede obtener una cuenta que le permitirá acceder al mismo.

- Acceso al sistema: Los usuarios entran al sistema a través de esta opción.

- Preguntas más frecuentes (FAQ): Una página está dedicada a explicar diferentes cuestiones relacionadas con la herramienta de integración de ontologías.

- Documentación: Esta sección propociona información técnica sobre cómo usar la herramienta de integración de ontologías, esto es, manuales y especificaciones de ontologías.

- Publicaciones: Diferentes artículos relacionados con esta herramienta y entorno de integración podrán ser accedidos desde aquí.

- Enlaces: En esta sección se pueden encontrar enlaces a diferentes proyectos y sistemas relacionados con ontologías.

- Información de contacto: Se proporciona información para contactar con los autores.

En la Figura 6.16 se muestra la página de inicio. Las opciones aparecen en la parte izquierda de la página. En el marco principal de la misma se muestra información relativa al integrador, cómo usarlo y algunas noticias relacionadas.

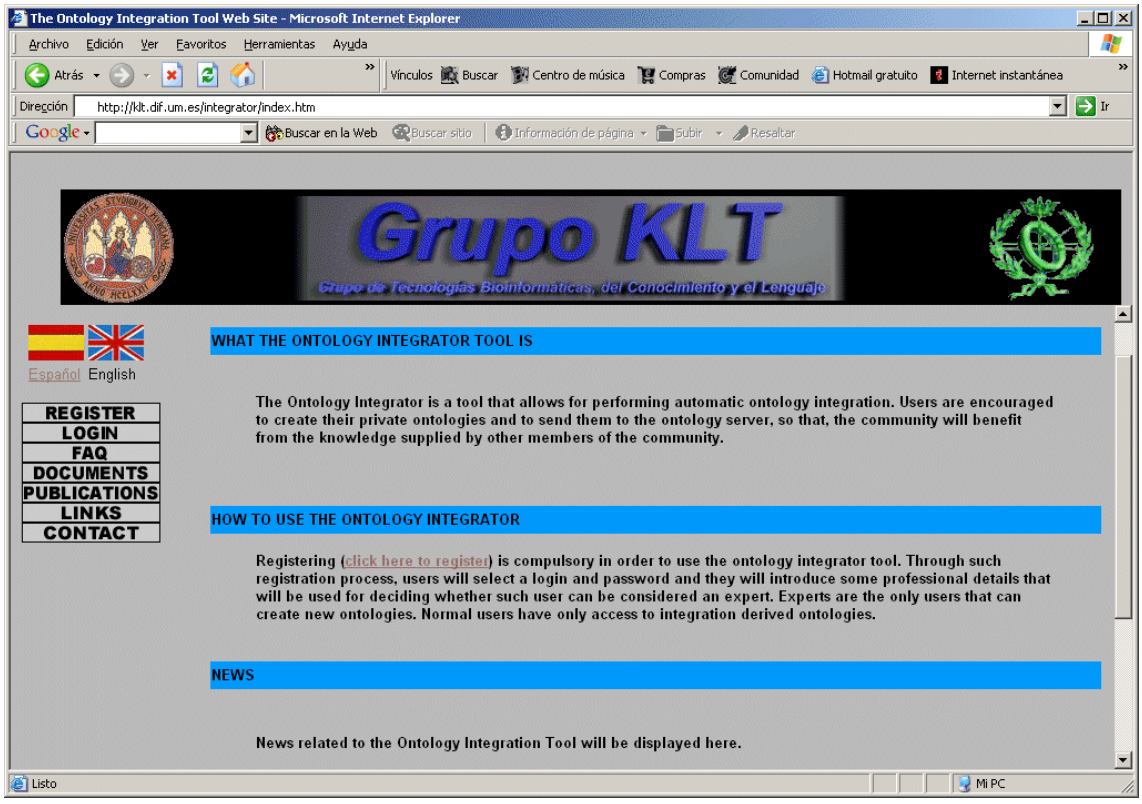


**Figura 6.16 La página de inicio del integrador de ontologías**

### 6.3.1 Registro como usuario

Haciendo clic en el botón "register" o en el enlace "click here to register" se muestra en la Figura 6.17. La información que debe introducir el usuario es la misma que aparece en la Figura 6.4.

**Figura 6.17 Un usuario web dándose de alta en el sistema**

### 6.3.2 Accediendo al sistema

La página mostrada en la Figura 6.18 se muestra cuando se hace clic en la opción "login". Entonces se solicitará al usuario que introduzca su nombre de usuario y su contraseña para poder acceder al sistema.

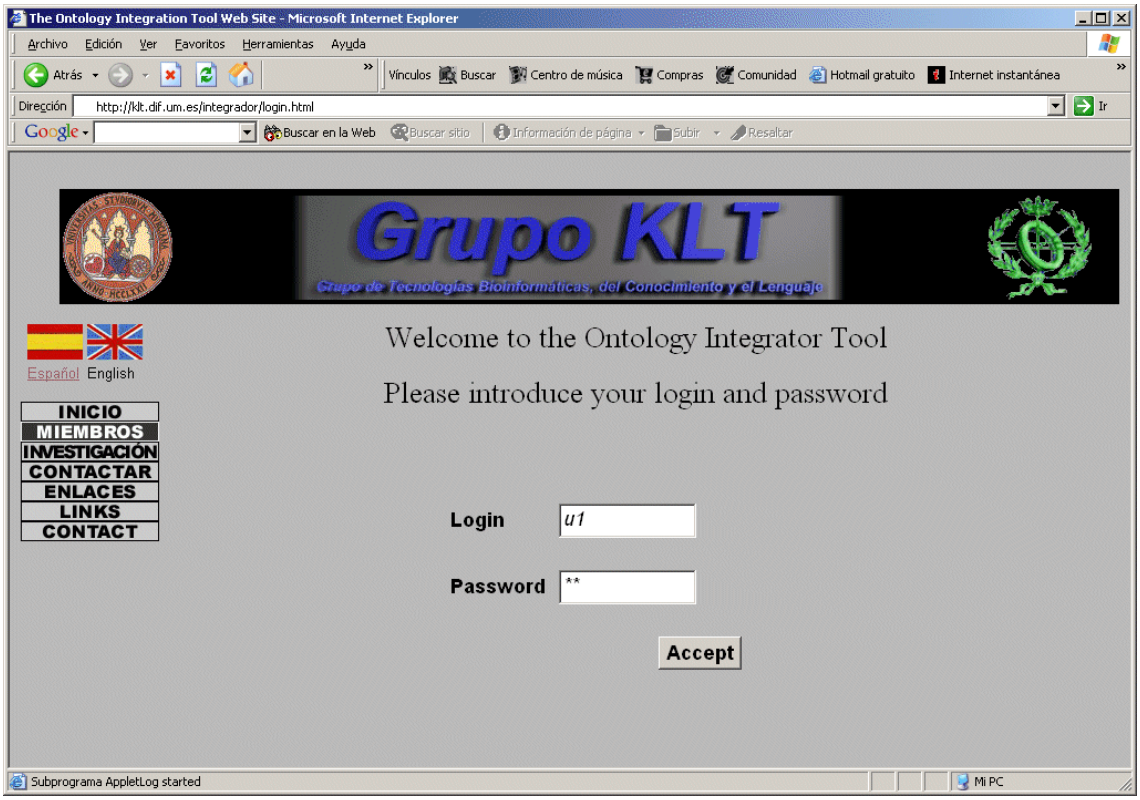


**Figura 6.18 Página de acceso al sistema**

## 6.3.3 Opciones Web para el administrador

Como en la versión no web, el administrador puede gestionar usuarios y temas directamente desde la web. Las tres figuras siguientes (Figura 6.19-6.21) muestran parte de las opciones proporcionadas al administrador. La Figura 6.19 es la página web principal para administradores y en ella se muestran las opciones generales de gestión que tienen los administradores. Una de estas opciones es la gestión de usuarios (ver Figura 6.20). En este caso se pueden realizar las mismas operaciones que en la versión no web. La misma conclusión puede deducirse de la Figura 6.21 que se ocupa de la gestión de temas y sugerencias hecha por los usuarios.

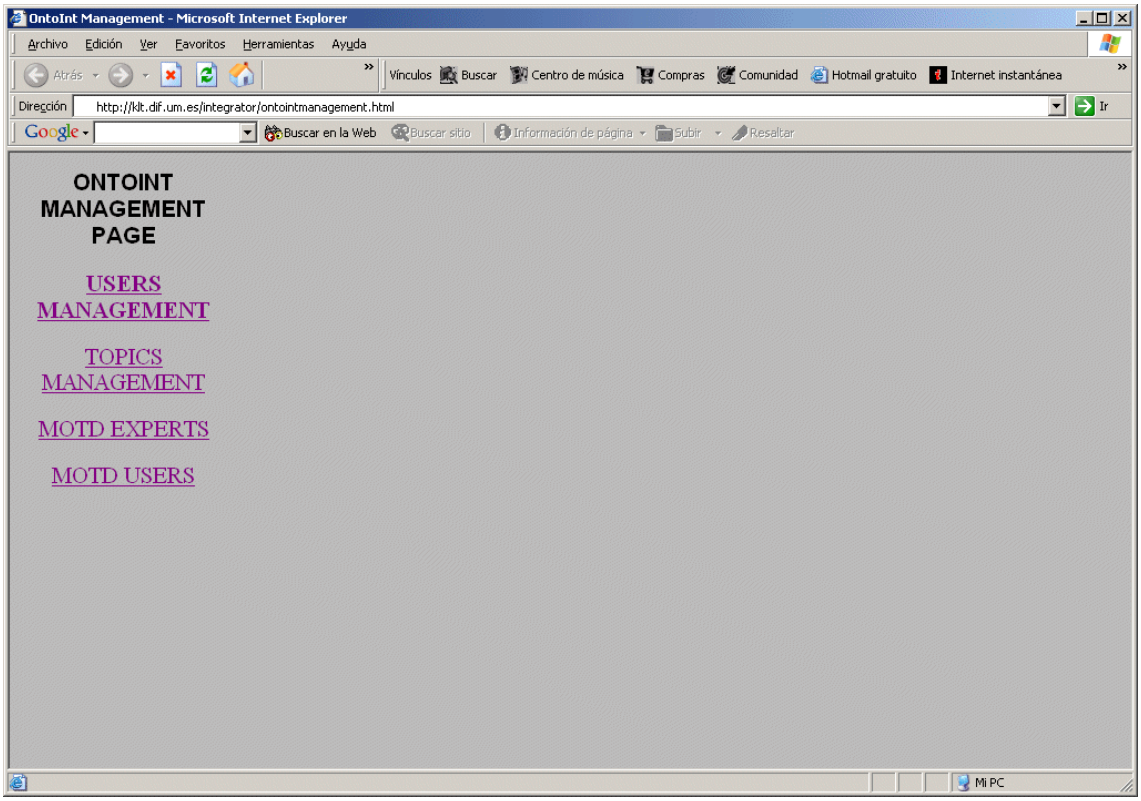


**Figura 6.19 Opciones de administrador**

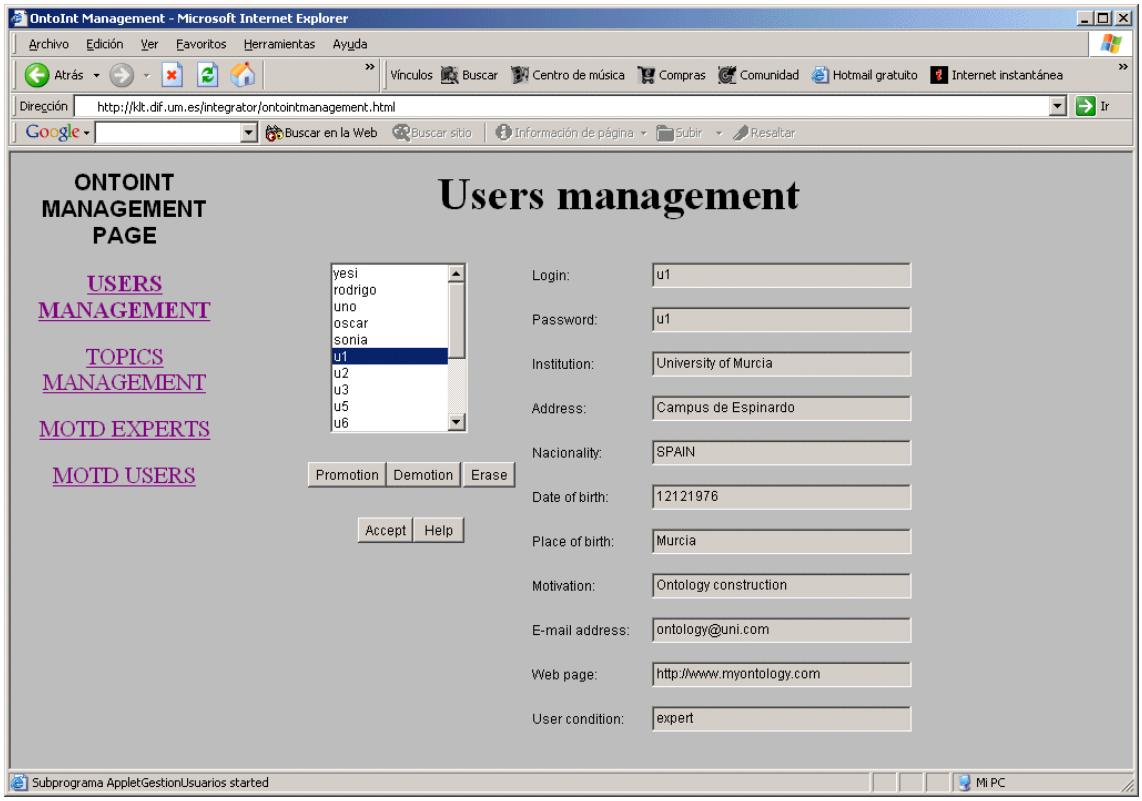


**Figura 6.20 Gestión de usuarios**

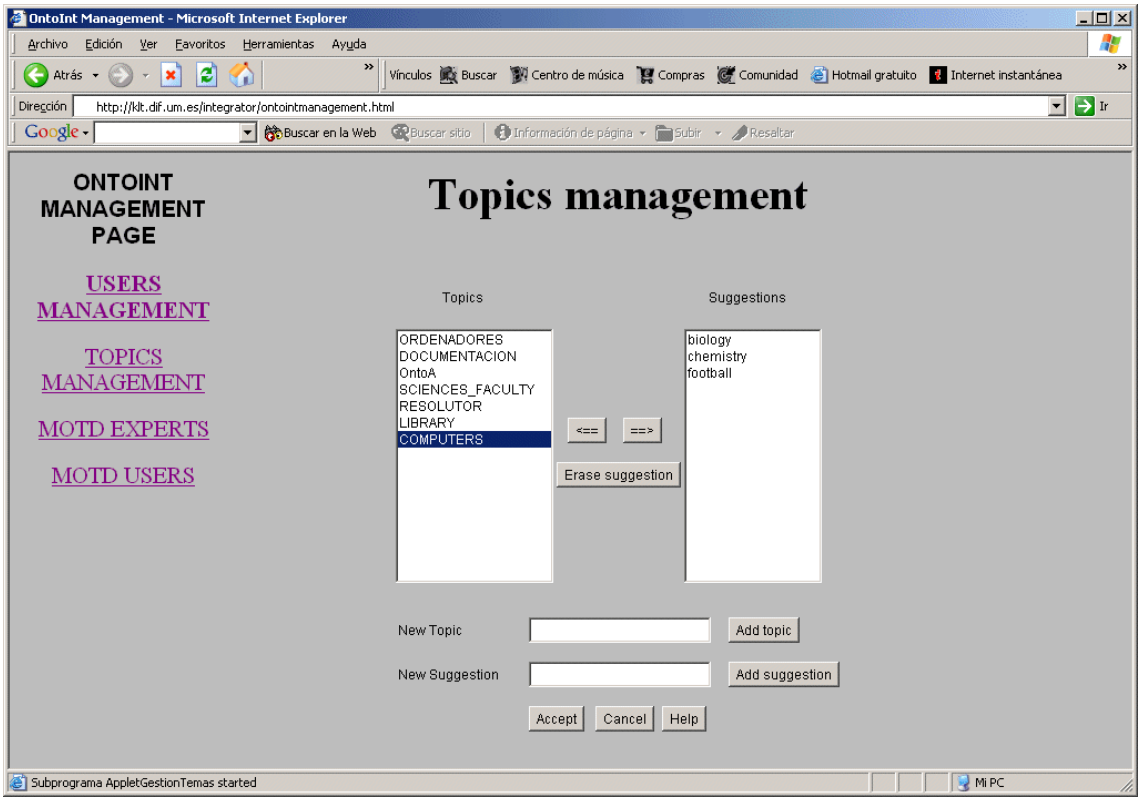


**Figura 6.21 Gestión de Temas**

## 6.3.4 Opciones web para usuarios normales

Una vez que se ha entrado al sistema, el usuario será redirigido atendiendo a su condición de usuario (experto o normal). En caso de usuarios normales se mostrará la Figura 6.22. En tal página se puede ver que los usuarios normales sólo pueden solicitar la integración de algún tema de trabajo. En dicha figura, los temas de trabajo actualmente activos en el sistema son "Ordenadores", "Documentacion", "OntoA", "Resolutor", "Sciences Faculty" y "Library". Por ejemplo, supongamos que el usuario ha seleccionado "Library". Dada esta situación, al hacer clic en el botón "Submit" se solicitará la integración del tema seleccionado.

**Figura 6.22 Página para usuarios normales**

Una vez realizada la petición de integración al sistema, ésta será procesada y al poco tiempo se mostrará otra página web conteniendo el resultado de la petición. Esta página web se generará dinámicamente. Las figuras siguientes ilustran cómo el sistema muestra la información obtenida del proceso de integración cuando este proceso es lanzado por un usuario web. La Figura 6.23 muestra los resultados de integración del tema de trabajo "Library". Exploremos en detalle toda la información proporcionada al usuario. La página resultado se divide en tres partes, superior, media e inferior. En la parte superior se muestra la ontología propiamente dicha. La ontología generada por el sistema como resultado del proceso de integración se muestra en forma arbórea, donde cada nodo representa un concepto de la ontología y cada enlace entre nodos es una relación, taxonómica o mereológica, entre ellos. El conocimiento relacionado con cada concepto aparece en la parte derecha de este componente. Como se puede apreciar en la figura, dicha información se muestra en lenguaje natural, de forma que el conocimiento ontológico sea fácilmente entendible por los usuarios.

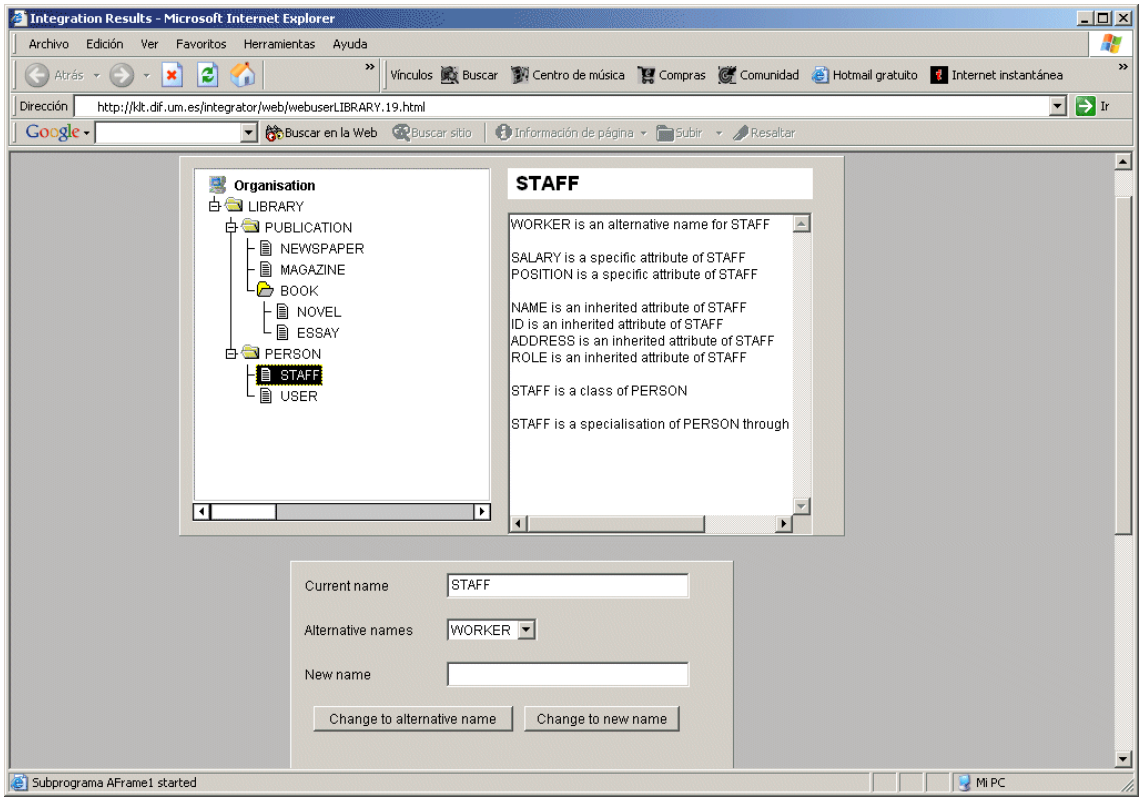


**Figura 6.23 Resultados de la integración**

Por ejemplo, podemos leer que "SALARY es un atributo específico de STAFF" o que "STAFF es un tipo de PERSON".

### *6.3.4.1 Personalización terminológica*

La parte intermedia de esta página, que se muestra en la parte baja de la Figura 6.23, permite a los usuarios elegir la terminología de la ontología. En este componente se muestran tres campos, el nombre actual del concepto seleccionado, su lista de nombres alternativos, y un campo para proponer nuevos nombres para los conceptos seleccionados. Cambiemos el nombre del concepto "staff" para ilustrar los efectos de cambiar el nombre de un concepto. Para ello, le daremos su nombre alternativo "worker". La Figura 6.24 muestra tales modificaciones.

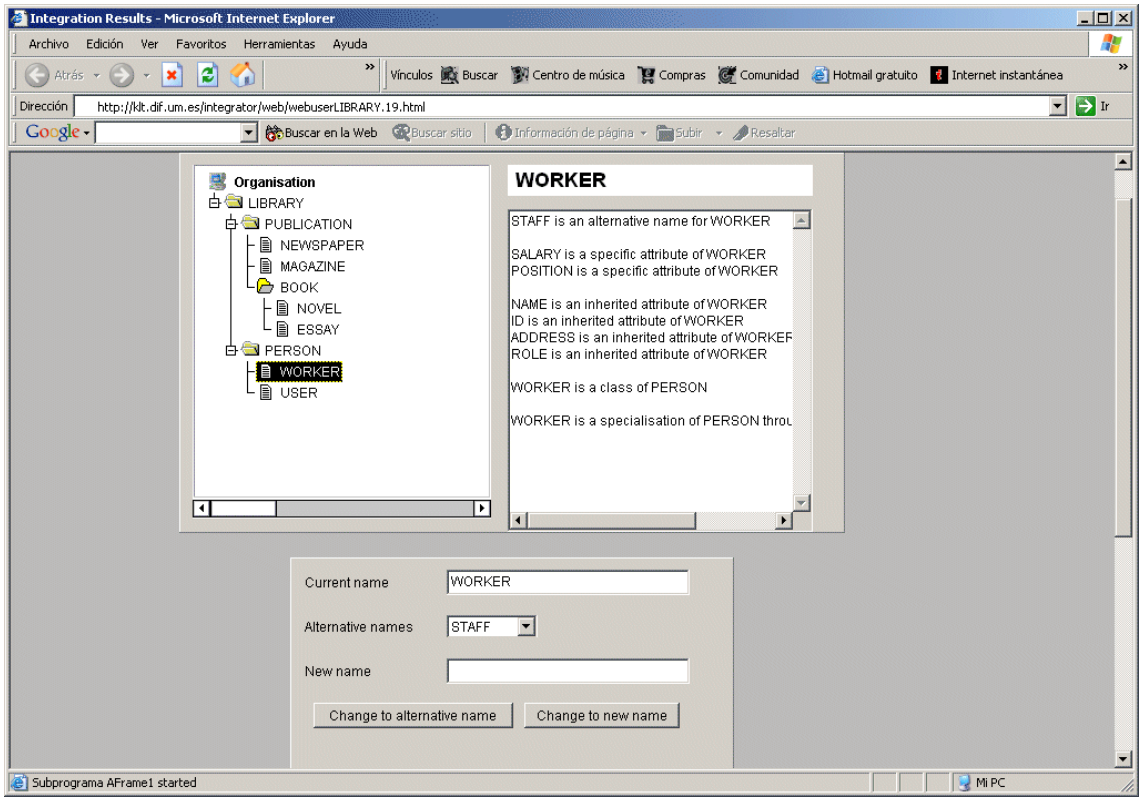


**Figura 6.24 Cambiar "staff" por "worker"**

En la Figura 6.24 podemos ver que el concepto se llama ahora "worker" y que "staff" es ahora el nombre alternativo. Sin embargo, no es la única modificación terminological que se puede realizar. El usuario podría pensar que el término "personal" es más apropiado para el concepto que "worker" o "staff", por lo que ese nombre se introduce en el correspondiente campo de texto. La Figura 6.25 muestra cómo queda la ontología tras esta modificación.

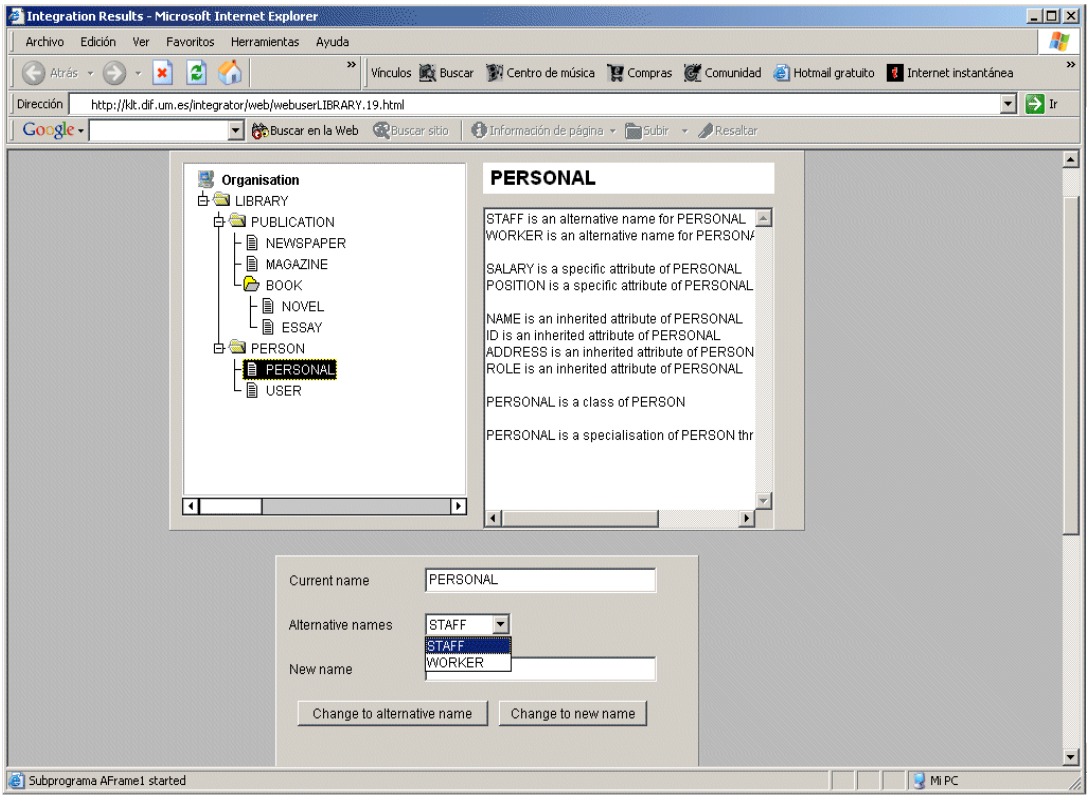


**Figura 6.25 Cambio de "worker" por "personal"**

### *6.3.4.2 Explorando la ontología*

Nos podríamos preguntar cómo explorar la ontología, esto es, llegar a niveles más profundos. Supongamos que la situación actual es la mostrada en la Figura 6.26. Los conceptos son expandidos al hacer doble clic sobre el concepto deseado. Esta expansión es automática en caso de los descendientes mereológicos. Si el concepto tiene hijos taxonómicos, podemos decidir a través de qué atributos queremos explorar, esto es, debemos seleccionar las especializaciones en las que estamos interesados. Cuando expandimos un concepto se activa un último panel.

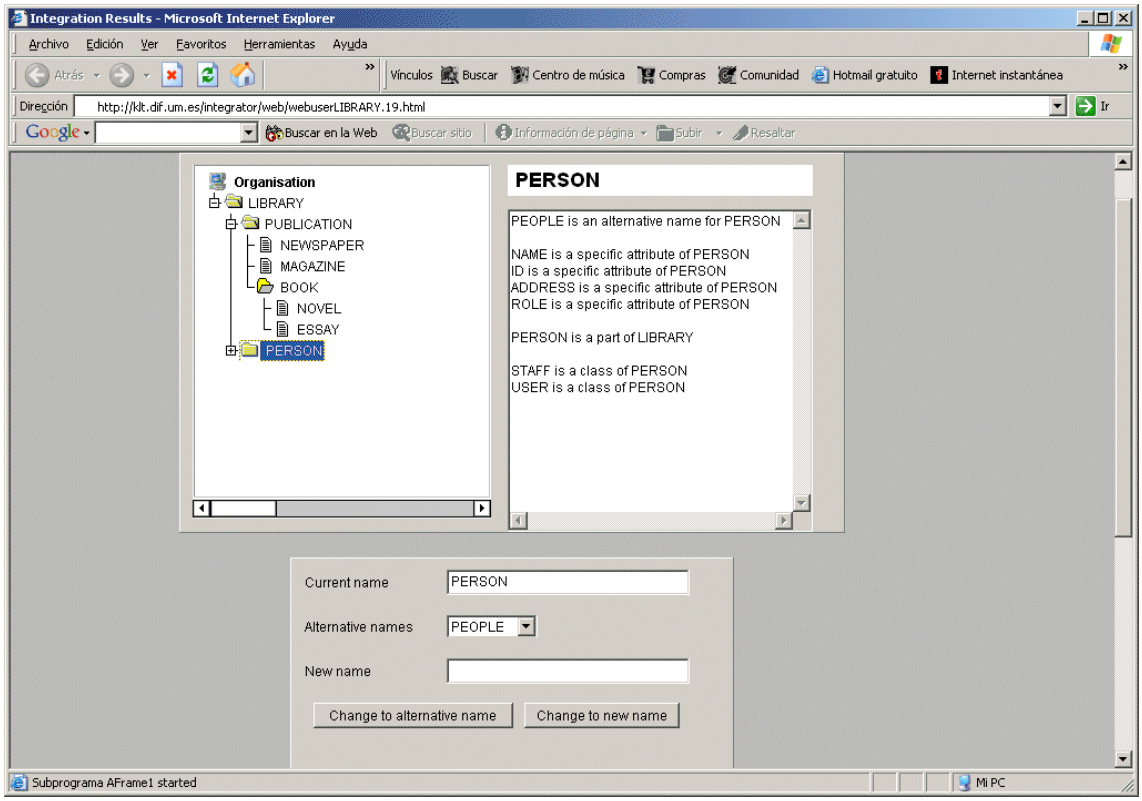


**Figura 6.26 Una ontología parcialmente expandida**

Este panel muestra el conjunto de atributos heredados y específicos del atributo seleccionado (ver Figura 6.27). El siguiente paso es seleccionar el conjunto deseado de atributos, que se muestra en la parte derecha del panel, llamada “atributos seleccionados”.

Podemos seleccionar los atributos uno por uno utilizando el botón “>” , o bien todos los atributos usando el botón “>>”. La Figura 6.28 muestra la selección del atributo “Role” para la expansión del concepto “Person”.

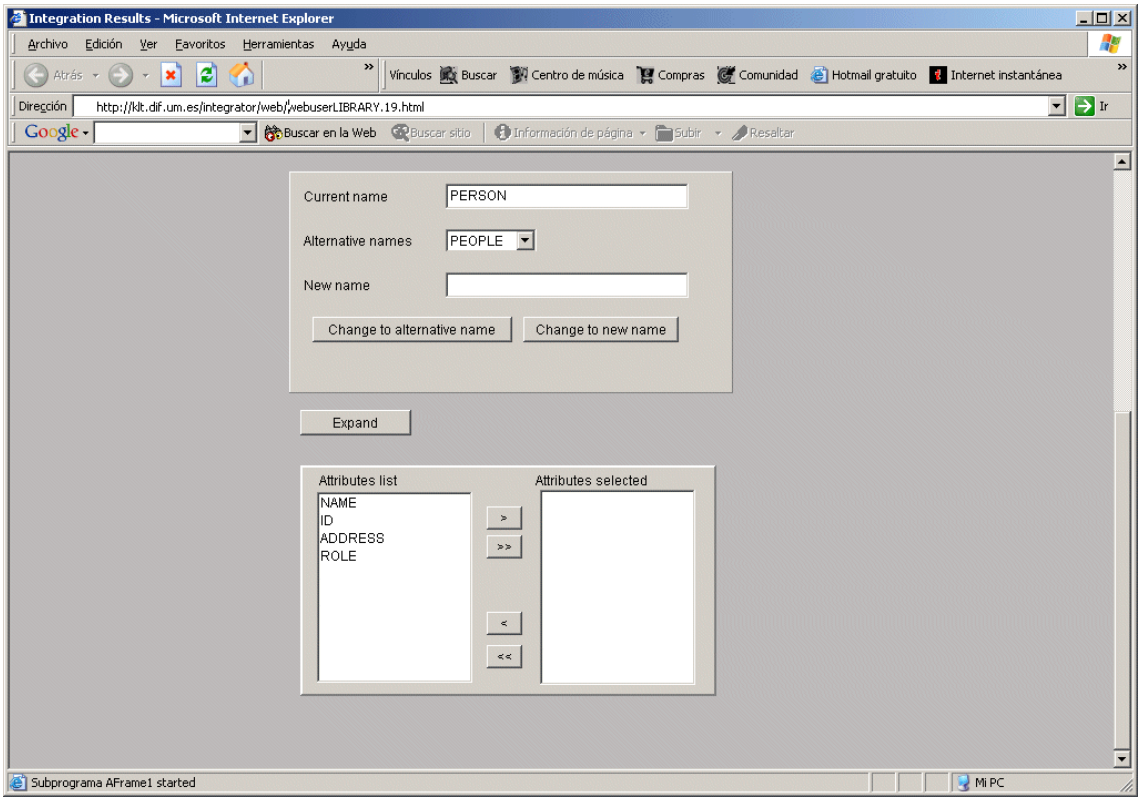


**Figura 6.27 Panel de atributos**

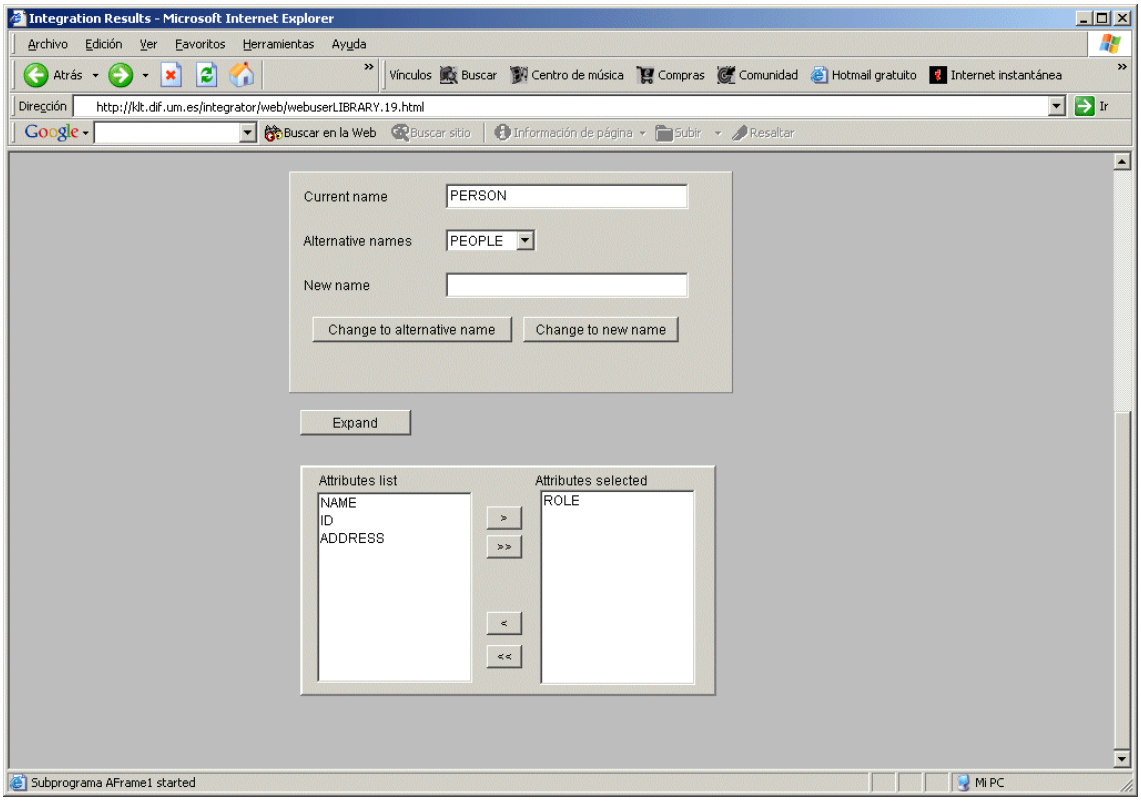


**Figura 6.28 Elección del atributo a expandir**

Al seleccionar los atributos debemos presionar en el botón “Expand” para aplicar la expansión deseada a la ontología. Eso hará que se muestren de entre los hijos taxonómicos del concepto seleccionado, aquellos que representen una especialización del concepto seleccionado a través de al menos uno de los atributos seleccionados (ver Figura 6.29).

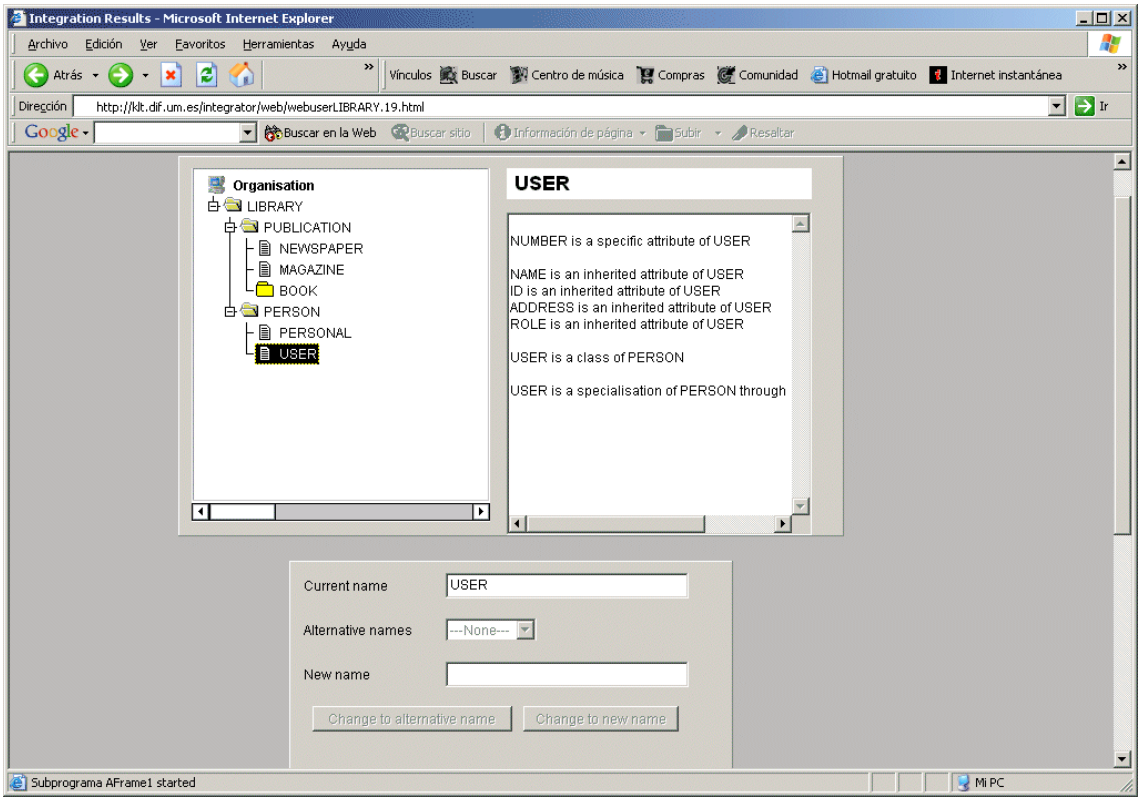


**Figura 6.29 La ontología expandida**

## 6.3.5 Opciones Web para usuarios expertos

Todas las opciones mostradas previamente para usuarios normales están también disponibles para usuarios expertos. Sin embargo, éstos tienen una serie de facilidades extra. La página de inicio para usuarios expertos se muestra en la Figura 6.30. En esta figura podemos ver que los expertos también pueden elegir ficheros de trabajo y enviarlos al servidor de ontologías para que estas ontologías puedan tomar parte de los correspondientes procesos de integración.

**Figura 6.30 Página de inicio para expertos**

Finalmente, debería señalarse que la interfaz web ha sido desarrollada tanto en castellano como en inglés, aunque la herramienta sólo está disponible, por el momento, en inglés. La página de inicio en castellano se muestra en la Figura 6.31.

**Figura 6.31 La página de inicio en castellano**

## 6.4 COMPARACIÓN CON OTRAS HERRAMIENTAS DE INGENIERÍA ONTOLÓGICA

En (Duineveld et al, 2000) se evalúan algunas herramientas de ingeniería ontológica en función de diferentes parámetros como interfaces, facilidades ontológicas y facilidades de cooperación. Los sistemas evaluados son Ontolingua (Rice et al, 1996; Farquhar et al, 1997), WebOnto (Domingue, 1998), ProtégéWin (Eriksson et al, 1999), OntoSaurus (ISX, 1991), y ODE (Fernández et al, 1999). La información de este apartado se basa en los resultados presentados en este estudio. A continuación compararemos nuestro entorno y sistema con estas herramientas.

## 6.4.1 Propiedades ontológicas

Esta sección compara las facilidades ofrecidas por las herramientas ontológicas previamente nombradas para el desarrollo de ontologías del dominio. Todas las herramientas anteriores incluyen la relación taxonómica. La herencia múltiple también es soportada por todos ellos aunque, por ejemplo, ODE no incluye ningún motor de inferencia para la propagación automática de la herencia. La mayoría de ellos permite la posibilidad de definir particiones exhaustivas y/o descomposiciones disjuntas, aspecto que nuestro sistema no proporciona. La relación mereológica no está incluida en estas herramientas debido a la falta de consenso en cuando a la semántica de la relación parte-de. Sin embargo, esta relación ha sido definida y empleada en nuestro entorno, donde se ha usado un modelo simple de mereología. Se entiende que la relación mereológica se compone de distintas subrelaciones (ver Artale et al, 1996; Fridman-Noy y Hafner, 1997). Estos autores declaran que hay una única relación parte-de sino que hay diversas modalidades de relación entre una parte y su todo, como Componente/Objeto (por ejemplo, motor/coche), Miembro/Colección (p.ej, árbol/bosque), Porción/Masa (p.ej., porción/pastel), Constituyente/Objeto (p.ej, acero/bicicleta, Característica/Actividad (p.ej., coger/apilar), y Lugar/Área (p.ej., cudad/país). Parte de nuestro trabajo futuro está dedicado a extender nuestro modelo ontológico para cubrir todos estos tipos de relaciones entre partes y todos.

## 6.4.2 Facilidades de cooperación

En este apartado se analizan las facilidades ofrecidas por los distintos sistemas para construir ontologías de forma cooperativa.

### 6.4.2.1 *Ontolingua*

En Ontolingua, varios usuarios pueden editar la misma ontología al mismo tiempo, aunque los cambios hechos por un usuario no son visibles de forma automática para el resto de usuarios mientras que, en nuestro enfoque, cada usuario tiene una ontología privada y la cooperación se instrumentaliza a través del mecanismo de integración. En nuestro enfoque, los usuarios no han de ser notificados de los cambios hechos por otro usuario, sino que cada uno trabaja de forma independiente, lo cual simplifica su trabajo. Ontolingua también proporciona una política de acceso dependiente del usuario. Además, una ontología puede ser bloqueada de forma que los usuarios no pueden editarla, sino únicamente consultarla. De hecho, se utilizan restricciones de acceso en Ontolingua cuando un usuario está modificando la ontología. En nuestro entorno estas restricciones tienen una naturaleza diferente, puesto que tenemos diferentes tipos de usuarios, expertos y normales. Los usuarios normales no poseen ontologías mientras que los usuarios expertos sí poseen ontologías sobre algún tema de los propuestos en el sistema.

### *6.4.2.2 WebOnto*

El enfoque de cooperación usado en WebOnto es similar al usado en Ontolingua puesto que los usuarios modifican la misma ontología, pero cuando un usuario la modifica los otros usuarios solo pueden explorarla (en Ontolingua los usuarios reciben notificación de modificaciones, pero no quedan bloqueados de la misma forma que en WebOnto). Nuestro enfoque puede ser considerado más apropiado para los usuarios, porque pueden modificar sus ontologías tan pronto como tengan conocimiento para añadir, modificar, o eliminar, sin ser bloqueados por otros usuarios.

*6.4.2.3 OntoSaurus*

Los usuarios cooperan para construir la misma ontología de forma que también se bloquea a los usuarios cuando es necesario, aunque se proporciona la posibilidad de enviar emails a otros usuarios del sistema. El método de bloqueo es diferente al usado en Ontolingua o WebOnto porque los cambios no son hechos públicos inmediatamente a los otros usuarios, y no pueden explorar la ontología hasta que el usuario que está modificando la ontología haya acabado tales modificaciones.

Finalmente debemos señalar que ni Protégé-Win ni ODE facilitan la construcción cooperativa de ontologías.

## 6.5 RESUMEN

En este capítulo se ha presentado una herramienta para integrar ontologías. Dicha herramienta cumple los requisitos impuestos por el entorno de integración, por lo que esta herramienta es apropiada para integrar ontologías según el modelo de integración ontológico propuesto en los capítulos anteriores. En particular, la herramienta permite la construcción colaborativa de ontologías reusando ontologías ya existentes e integrando todas ellas en una ontología global. Se pueden resumir las propiedades de la herramienta de integración de ontologías como sigue:

- **Privacidad**: Como se mencionó anteriormente en esta tesis, cada ontología no derivada de la integración es privada y sólo el experto que la creó puede acceder a ella. Esta propiedad queda garantizada en la herramienta por el modo de acceso de los usuarios al sistema y por la forma en que las ontologías son almacenadas en el servidor.

- **Compartición y reutilización de conocimiento**: El marco de integración de ontologías promueve tales propiedades porque se reutilizan ontologías ya existentes para generar la

ontología global, y el conocimiento creado por cada experto se pone a disposición del resto a través de los procesos de integración. La herramienta hace uso de ontologías creadas por varios expertos para ejecutar procesos de integración cuando éstos son requeridos por los usuarios.

- **Máximo contenido informativo**: El entorno ontológico pretende proporcionar a los usuarios el máximo conocimiento posible. La herramienta ha sido diseñada para cumplir tal requisito.

- **Dependencia de usuario:** El marco de integración de ontologías personaliza la terminología usada por el usuario en caso que tal usuario haya construido un ontología en el tema a integrar. La herramienta también personaliza tal terminología, por lo que dicho objetivo se satisface.

- **Distribución de usuarios:** En el marco de integración se supone que el conjunto de usuarios expertos se encuentran geográficamente distribuidos. Este requerimiento lo cumple el sistema puesto que ha sido diseñado e implementado para funcionar en una intranet/internet.

- **Construcción de ontologías**: El marco ontológico especifica los procesos para construir e integrar ontología. También se ha propuesto un lenguaje sencillo para especificar ontologías, y tal lenguaje ha sido incorporado en la herramienta de integración. Además, las ontologías integradas son expresadas en dicho lenguaje, de forma que estas pueden ser modificadas usando el mismo tipo de editores de ontologías.

Se han presentado dos versiones de la herramienta. La primera es una clásica aplicación en la que el usuario ejecuta un software cliente para trabajar con la herramienta de integración de ontologías. La segunda versión está orientada a la web, y se ha realizado un portal web para la herramienta, por lo que los usuarios pueden trabajar directamente desde sus navegadores web. Los usuarios tienen las mismas facilidades en ambas herramientas, aunque la construcción del interfaz web permite, posiblemente, acceder al sistema de forma más cómoda. En este capítulo

se ha descrito una sesión de trabajo para cada tipo de usuario para cada versión de la herramienta. Finalmente, a través de la herramienta presentada en este capítulo se cumple unos de los objetivos de esta tesis, esto es, proporcionar una herramienta software que posibilite la construcción cooperativa de ontologías del dominio a través de procesos de integración de conocimiento.

# CAPÍTULO 7

# CONCLUSIONES

# Y

# LINEAS FUTURAS

Este trabajo representa un intento de simplificación de los procesos de creación cooperativa de conocimiento. En la actualidad existen diferentes enfoques para realizar tales procesos, aunque ninguno de ellos puede ser considerado como exitoso ni ha sido considerado como estándar, debido a que cada uno de ellos presenta ciertas desventajas. Como ejemplos de tales inconvenientes podemos destacar los reducidos modelos de conocimiento empleados, o el bloqueo de los expertos cuando otro experto está editando la ontología común. Asimismo, el análisis de dichos enfoques revela la existencia de dos tendencias: (a) desarrollo cooperativo de una misma ontología; (b) integración de ontologías.

Entre los enfoques pertenecientes a la primera tendencia podemos destacar Ontolingua (Farquhar et al, 1997), WebOnto (Domingue, 1998) y OntoSaurus (ISX, 1991). Por otra parte, las metodologías de integración de ontologías más conocidas son PROMPT (Fridman-Noy and Musen, 2000), Chimaera (McGuiness et al, 2000) y FCA-Merge (Stumme and Maedche, 2001). Sin embargo, tales enfoques no son automáticos, de forma que la participación de los expertos es obligatoria en casi todos los pasos del proceso. FCA-Merge es el enfoque que incluye más pasos automáticos. De hecho, la participación del experto sólo se requiere para integrar la ontología una vez construidos automáticamente los enrejados de conceptos.

Además, ninguno de los enfoques anteriores proporciona un acceso personalizado al conocimiento, de forma que los expertos no pueden acceder al mismo en función de sus preferencias. Por lo tanto, existe la necesidad de soluciones para la construcción cooperativa de conocimiento. Todos estos factores nos han llevado a proponer una metodología para desarrollar ontologías de forma cooperativa a través de procesos automáticos de integración de conocimiento, y que permite acceder al mismo de manera personalizada.

La metodología propuesta proporciona un entorno formal para desarrollar ontologías de forma cooperativa basado en la integración de piezas de conocimiento (p.ej., ontologías). El objetivo de estos procesos de integración es obtener ontologías que contengan el mejor y mayor conocimiento posible, de forma que dichas ontologías puedan ser consideradas apropiadas para

el dominio en cuestión. Toda vez que se han construido buenas ontologías, estas pueden ser usadas para desarrollar sistemas para la gestión de conocimiento. Otro objetivo de este trabajo es proporcionar una herramienta eficiente y potente para representar el conocimiento necesario a los desarrolladores de sistemas para la gestión de conocimiento para su creación.

Veamos como las ontologías pueden ser usadas en sistemas para la gestión de conocimiento. Según O'Leary (O'Leary, 1999), dichos sistemas deben facilitar los siguientes aspectos:

- Conversión de datos y texto en conocimiento. Por una parte, datos y conocimiento han de ser convertidos en conocimiento. Por otro lado, las ontologías son la tecnología más apropiada para representar conocimiento. Por lo tanto, las ontologías son útiles para esta tarea.

- Conversión del conocimiento individual y de grupo en conocimiento accesible. Las ontologías pueden ser usadas para representar dicho conocimiento. Además, este proceso puede beneficiarse de los procesos de integración de conocimiento.

- Conectar usuarios y conocimiento a otros usuarios y a otro conocimiento. La riqueza de la semántica asociada a las ontologías incrementaría la eficiencia del proceso.

- Comunicación de información entre diferentes grupos. En principio, no es objetivo de las ontologías resolver este asunto.

- Creación de nuevo conocimiento útil para la organización. Se debería usar ontologías para esta tarea, y si el conocimiento proviene de diferentes fuentes, se debería aplicar procesos de integración de conocimiento.

En definitiva, las principales aportaciones de esta tesis son las siguientes:

- Especificación de un modelo ontológico formal. Dicho modelo está compuesto por conceptos, atributos específicos y heredados, relaciones taxonómicas y mereológicas, y axiomas estructurales.

- Especificación de un marco formal para la integración y personalización de conocimiento. A través de dicho entorno, un conjunto de ontologías especificadas por expertos del dominio y cubriendo el mismo dominio son integradas automáticamente en una ontología común. El conocimiento resultante se personaliza de acuerdo al individuo que solicitó la integración.

- Desarrollo de un sistema que implemente el entorno de integración y personalización anterior. Este sistema permite la construcción cooperativa de ontologías a través de procesos de integración de ontologías.

- Definición de un DTD XML para facilitar la interoperabilidad de nuestro sistema y las ontologías con otras herramientas para la gestión de ontologías.

- Validación de los elementos anteriores. En primer lugar, se ha validado el proceso de construcción de ontologías en diferentes dominios y se han obtenido conclusiones semejantes en cada caso. Posteriormente, la validez y utilidad del enfoque de integración fue validado de dos maneras. El primer aspecto se comprobó a través del análisis de los resultados obtenidos en procesos de integración en diferentes dominios, obteniendo semejantes conclusiones en cada caso. El segundo aspecto se comprobó al solicitar a varios expertos que construyeran una ontología en un dominio común para después realizar varias iteraciones de integración y analizar los resultados obtenidos. Finalmente, el entorno de integración ha sido aplicado satisfactoriamente en un dominio biológico para construir una ontología de forma cooperativa. También se ha usado el marco de integración como núcleo de un sistema para la gestión de conocimiento en Unidades de Cuidados Intensivos.

Por lo que respecta a trabajo futuro, se pueden señalar diferentes aspectos. Por ejemplo, el modelo ontológico puede ser mejorado de distintas maneras. El modelo ontológico actual no permite definir axiomas no estructurales. Sin embargo, no es la única mejora que se puede realizar con respecto al modelo ontológico ya que, por ejemplo, se podrían añadir más relaciones al mismo. Concretamente, una extensión del modelo original está en fase de desarrollo. Dicha extensión añadirá relaciones temporales a las relaciones taxonómicas y mereológicas ya existentes. En (Gómez-Pérez et al, 2000) se identifican las relaciones interconceptuales más comunes en dominios reales. Sería deseable cubrir al menos esos tipos de relaciones en el modelo.

Otra posible mejora al modelo ontológico está relacionada con proporcionarle estructura interna a los atributos, que actualmente sólo se identifican mediante cadenas de caracteres. Sería deseable añadir valores a los mismos, así como disponer de distintos tipos de atributos.

Otras mejoras estarían relacionadas con los procesos de integración de conocimiento. En este trabajo se descarta el conocimiento inconsistente del proceso de integración. Puesto que las ontologías pueden ser parcialmente inconsistentes, sería una mejora el poder descartar únicamente aquellas partes inconsistentes en vez de la ontología completa. Para ello, las ontologías fuente podrían ser modificadas de forma que se eliminara de ellas el contenido inconsistente. Actualmente, únicamente el conocimiento completamente consistente tomaría parte en procesos de integración. Esta mejora implicaría añadir un paso adicional al proceso de integración de ontologías. Este paso se situaría después de seleccionar el mejor subconjunto de ontologías candidatas porque este subconjunto contendría la mayor cantidad de ontologías compatibles, de forma que sólo habría que modificar las ontologías no incluidas en dicho conjunto para que fueran compatibles con el resto.

En resumen, el marco formal para la integración de ontologías presentado en esta tesis se puede situar en el contexto de Ingeniería Ontológica y Gestión del Conocimiento puesto que los resultados obtenidos son significantes para ambas áreas de investigación. En primer lugar, es significante para la Ingeniería Ontológica porque proporciona un modelo ontológico y un marco ontológico formales para el desarrollo cooperativo de ontologías. El desarrollo de ontologías es

una tarea complicada, por lo que dicho desarrollo cooperativo y automático es altamente beneficioso para esta área. Finalmente, esta metodología de construcción de conocimiento contribuye a mejorar el desarrollo de sistemas para la gestión de conocimiento, como se explicó anteriormente. En particular, el uso del entorno presentado en esta tesis simplifica la mayoría de aspectos relacionados con conocimiento en este tipo de sistemas.

# CAPÍTULO 8

# RESUMEN EN INGLES / SUMMARY IN ENGLISH

The motivation for this thesis stems from the need of solutions for developing knowledge to build knowledge management systems. In particular, the main objective of the work here presented is to enhance knowledge integration processes. With this, the development of knowledge management systems would be more efficient. In this work, we are going to work from an ontological engineering perspective. The improvements here proposed are due to the drawbacks found in current collaborative ontology development methodologies such as Ontolingua (Farquhar et al, 1997), WebOnto (Domingue, 1998), Prompt (Fridman-Noy and Musen, 2000) or FCA-Merge (Stumme and Maedche, 2001).

In order to achieve this objective, the following methodology will be followed:

- Analysis of the state of art in Ontological Engineering. This implies the study of available methodologies for the different activities of the ontology lifecycle, such as design, creation, or evaluation. The study of current languages for representing ontologies is also included in this analysis.

- In-depth analysis of current collaborative knowledge construction methodologies. The study has been based on factors such as the knowledge model supported by each methodology or the degree of domain experts' participation during the process.

- Definition and formalisation of a framework for integrating ontologies. Such process has been divided into two parts: (1) definition and formalisation of an ontology model; (2) definition and formalisation of a methodology for integrating ontologies, which are specified according to the mentioned ontology model. The participation of the domain expert in the integration process has been limited to supplying a private ontology, since the ontologies will be automatically

integrated. Moreover, the results of integration processes have been adapted to each particular expert.

- Design and implementation of a software application to allow the cooperative construction of ontologies. A web interface for facilitating the use of the application from web browsers has been provided. In this system, experts are not allowed to access other experts' knowledge except through the results of ontology integration processes.

The different parts of the formal framework for integrating knowledge have been validated. The validation methodology can be summarised as follows: (1) validation of ontology construction methodology; (2) evaluation of the quality of the ontologies obtained through integration processes; (3) evaluation of the usefulness of the integration process.

The previous issues have been successfully accomplished, and the results obtained are presented in this thesis. Let us now describe hoe each objective was achieved.

# STATE OF ART

The state of art in ontological engineering has been analysed. The first part of such analysis was centred in the origin of the notion of ontology. In literature, different definitions from different nature (i.e., linguistics, philosophy, artificial intelligence) can be found. Philosophy-related definitions are more ambitious since philosophical ontologies account for pure concepts of the understanding or of reason. In Artificial Intelligence, two main approaches are distinguished. On the one hand, some computing researchers consider that ontologies define the basic terms and relations of a particular domain. On the other hand, the philosophical AI community defends the original philosophical notion of ontology and uses ontologies in a more formal manner. In fact, ontology has a philosophical origin, since Thales of Miletos was who first mentioned this concept when searching for the answer to the question "what is everything

composed of?". Since then, a huge amount of theories for accounting existence have been proposed. It can be said that the most significant ontological theory has been Aristotle's theory of ens, proposed in Aristotle's Metaphysics. Modern philosophers such as Leibniz or Kant made important apportions to make the concept of ontologies more popular. It was not until the 90's that ontologies were used in AI, when scientists realised of the need of formalise knowledge for developing intelligent systems. Now, ontologies are the most widespread technology for representing knowledge in AI because ontologies provide structured and formal representation of knowledge, and ontological representations are shareable and reusable.

Moreover, different classifications of ontologies have been proposed, according to different criteria. One of such classifications distinguishes between descriptive and formal ontology. While descriptive ontologies are concerned with the collection of information, formal ontologies distill, filter, codify and organise the results of descriptive ones. In AI, ontologies are heavyweight in case they include axioms or lightweight when no axioms are included. Axioms are one of the elements used to specify ontologies. Other important ontological elements are: classes (or concepts), relations, functions and instances. Different ontological models have been proposed and differences between such models have a two-folded nature: (1) type of ontology to model; and (2) richness of the ontology model, that is, type, quantity and purpose of the elements included in the model. Relations are an important element in an ontology model provided that they are the instrument for relating ontological entities. Amongst the possible relations, two have been distinguished as the most important ones, taxonomy, which is a theory of classification, and mereology, which is a theory of parts.

Due to the increasing importance of ontologies, different mechanisms and methodologies for designing and building ontologies have been proposed. Some of them were developed to obtain particular ontologies whereas other methodologies had a more general purpose. Furthermore, methodologies for evaluating the quality of the ontologies built have been reported, since the objective is to design good and useful ontologies. An important issue to point out is the role, use and importance of ontologies in the development of information systems. Ontologies have been mainly used as: (1) a representation vocabulary and content to be discussed; (2) a content theory; (3) performing ontological analysis to clarify knowledge structure; and (4) knowledge structure in the Semantic Web.

The most significant part for this thesis of the state of art analysis is the revision of different methodologies for the collaborative development of ontologies. Two main approaches are distinguished. On the one hand, different people develop collaboratively the same ontology. This has different drawbacks, since developers have to be locked when other developers are making their contribution to the ontology, so that mechanisms for synchronising such aspects must be provided. On the other hand, different people develop their own ontology and the resulting ontologies are integrated into a single one. In this case, developers are not locked but mechanisms for integrating private ontologies must be provided. The generation of the common ontology is more difficult than in the former approach, because developers are likely to use different terminologies. This aspect is more controlled in the former since developers work on the same ontology, that is, they have direct access to the knowledge supplied by the collaborators. Ontology integration is a process which is gaining significance in the latter years. At the beginning it was considered as an activity whereas it is now considered a whole process. The analysis of ontology integration systems, the main drawback found is that none of them is capable of performing automatic ontology integration. The user is who must make the decisions about which concepts to merge. Most of these systems deal only with taxonomic relations so that the knowledge model is not very realistic. Hence, one of the objectives of this thesis is to provide a framework for developing ontologies in a cooperative way, through automatic ontology integration processes. The cooperative development of ontologies contributes to obtain better ontologies since these would incorporate knowledge from different expert sources. On the other hand, the most currently used ontology languages have been briefly described. In particular, the following languages for specifying ontologies were analysed: XOL (XML-based Ontology Exchange Language), SHOE (Simple HTML Ontology Extension), OML (Ontology Markup Language), RDF(S) (Resource Description Framework), OIL (Ontology Interchange Language), and DAML+OIL. The underlying knowledge model for each language has been described, and pieces of ontologies have been shown coded in the different languages.

Finally, different aspects related to Knowledge Management have been discussed, because ontologies have been found to be important for performing Knowledge Management activities and facilitating the development of Knowledge Management systems, which is an objective of this thesis.

# THE ONTOLOGY MODEL

This model aims to be useful for specifying domain ontologies. This ontology model contains the following knowledge elements: (1) concepts, accounting for the domain entities; (2) attributes, which are properties of concepts and they can be either specific or inherited; (3) relations, which establish hierarchies in the domain provided that the two relations included in the model, namely, taxonomy and mereology, are hierarchical relations; and (4) structural axioms, that is, axioms drawn from the proper structure of the ontology. A set of assumptions, axioms, and properties concerning the structure of ontologies have been formalised. Moreover, a set of ontological functions have been defined in order to provide mechanisms for working with such ontologies. Finally, it has been demonstrated that these ontologies can be considered Scott domains or Information Systems. This fact facilitates the exploitation of ontologies because, for instance, topological spaces could be defined on ontologies, improving in this way the amount of operations that can be performed on the ontologies. At this point, ontologies have been specified from a theoretical point of view.

Different ontology languages were analysed when reporting the state of art in ontological engineering. However, none of them was found optimal to represent ontologies specified according to the ontology model. Therefore, a simple language for specifying our ontologies has been proposed. This language was designed with the unique objective of facilitating the construction of our ontologies so that it is not capable of handling other types of ontologies. Therefore, in order to promote the interoperability of our ontologies, an XML DTD has been developed and it has also been presented in this chapter.

# THE INTEGRATION FRAMEWORK

A framework for integrating domain ontologies has been formalised. Such framework is comprised of different sets of functions, each set having different and specific purposes. On the one hand, there is a need for functions to be applied on a single ontology in order to obtain the different semantic components included in ontologies. On the other hand, a framework for integrating ontologies also requires functions for comparing the conceptual structures contained in a set of ontologies. This family of functions can also be split into different groups. The first group permits to perform comparisons between individual ontological elements belonging to different ontologies (i.e., concepts or sets of concepts). The second group is built on top of the former one so that inferences among different ontologies can be performed. Such inferences allow for deciding whether different ontologies are redundant or inconsistent. These two resulting labels are problems which have to be taken into account when developing a cooperative framework. Two ontologies are redundant when they both describe the domain in the same manner. Two ontologies are inconsistent when both domain descriptions represent incompatible viewpoints. A third problem arises when only parts of both domain specifications are equivalent. In this case we are facing synonymy. The framework presented includes mechanisms for solving such problems inherent to cooperation.

The basic operation in this process is the *integration* of a set of predefined ontologies. In this sense, we must clarify that the term "ontology integration" must be understood in a different sense to that underlying the ontology inclusion term in the Ontolingua Server (Farquhar *et al*., 1997). In this framework, the term "ontology inclusion" is related to the reuse of ontologies in order to create other ones, so that the task of creating an ontology is less costly. The inclusion process is performed by a vocabulary translation process in order to adapt the terminology used in the first ontology to that of the ontology into which it is to be included. Thus, this translation is reflected in the axioms, so that those belonging to one ontology can be included in the other one. However, in the present work, the choice of the most appropriate terminology to be included in the integration-derived ontology that a certain end-user (i.e. not an agent) will finally be shown is a principled, dynamic process.

To be more precise, this choice is a function of various parameters: (1) the agent that requests integration; (2) every ontology present in the integration-derived ontology has to be consistent with the rest of ontologies present in the integration-derived ontology; (3) no single ontology present in the integration-derived ontology can be redundant with the rest of ontologies present in the integration-derived ontology; and (4) the amount of knowledge that a certain ontology contains. The fourth parameter is evaluated by the maximum information content introduced in this paper, while the second and third are methods used to solve some of the most significant problems with co-operative work, inconsistency and redundancy respectively.

The first parameter can be split into two parts, namely, framework-side and user-side. The framework-side is only triggered when the user-side does not establish any constraint concerning the terminology to use, so that the framework applies the maximum information content principle. The user-side part is in charge of deciding which terminology to use. Nowadays, the unique facility that the system provides for this is the use of the terminology of the expert user who applied for integration as the reference terminology. Therefore, an increase in the client facilities, such as allowing the requesting user to choose the terminology to use from terminology used by other experts, should be provided.

Hence, the framework was designed to hold two integration principles:

(1) user orientation

(2) maximum information content

The former is guaranteed by the fact that the integration framework uses as a semantic and terminological reference for the process the knowledge supplied by the user who makes the integration request. Moreover, the framework uses only the latest versions of the ontologies created by the users. The later is guaranteed by the different criteria employed along the process.

The proper framework for integrating ontologies can be divided into different phases, namely, *selection*, *instantiation* and *transformation*. The integration process works on a set of incoming ontologies that are integrated so that a global, transformed ontology is produced. Hence, the selection process is in charge of deciding which ontologies finally take part in the integration process. For this purpose, we must calculate, for each ontology, its set of *compatible ontologies*. Once this calculus has been made, the *best subset of ontologies* must be selected. After accomplishing this task, the instantiation process, by which terminology is harmonized along the set of ontologies to integrate, is triggered. Finally, all the ontologies are merged and the final, transformed ontology is produced. The proposed and formalized ontology integration framework has also been set into a context by comparing it with a significant amount of different collaborative knowledge construction frameworks and tools. For each of these frameworks, the main differences with respect to our approach have been pointed out. Partial examples of some functions belonging to the framework have been proposed to illustrate the formalisation process.

A detailed practical integration example has been included in this thesis. Such example describes the modus operandi of the integration framework, and it covers all the significant integrations problems, such as synonymy or inconsistency. The selected domain was “Library”. Thus, four ontologies were created and four ontology integration processes were executed, each process having a different ontology as the reference.

# VALIDATION

The framework for integrating knowledge has been evaluated through a two-folded validation procedure. First, the integration framework has been validated through measuring and analysing the knowledge contained in the private ontologies and in the integrated one. The second subtask can be defined as the comparison of results achieved in different domains. Such subtask requires the definition of computable ontological parameters. Such parameters were the measurement elements for quantifying knowledge associated to each ontology.

For the first task, two independent experiments were carried out. In the first one, three application domains were chosen. A set of undergraduate and postgraduate students were asked to interview different experts and to build cooperatively domain ontologies from such interviews by making use of the integration framework proposed. The objective was to analyse the benefits provided by integration processes according to the knowledge gained by the ontology builders. In order to measure such knowledge gain, relative measurements have been established, which are related to the amount of knowledge entities specified for each knowledge category (i.e., concepts, attributes and relations) and to the size of the ontologies. Firstly, a local gain was calculated, that is, the knowledge gained for one particular knowledge category by one particular ontology. Next, such gain was extended to a set of ontologies, each measurement being performed for a single knowledge category. Finally, a global gain measurement was defined, this accounting for the whole set of knowledge categories and the whole set of ontologies. From the results presented in this section, we can come to the conclusion that integration processes are beneficial for knowledge workers due to the fact that final, global, transformed, integration-derived ontologies possess more knowledge than individual ones. The global knowledge gain value can indicate different things. In case of a large knowledge gain, it would be the case in which there is less overlapping among the integrated ontologies. This low degree of overlapping could exist due to two main reasons. On the one hand, experts could specify different parts of the domain. On the other hand, it might also happen that the integration framework had not been capable of detecting the existing synonymous concepts. In case a low knowledge gain is achieved, this implies that the domain ontologies overlap significantly, either from the definition of the ontologies or by the existence of synonymous concepts. In both cases, this would mean that there exists a high degree of agreement among experts.

An analysis by knowledge category has also been performed. By the experiments performed, we can conclude that it is easier to agree on the concepts rather than on the attributes. Results concerning relations are closer to those for concepts. This fact is not surprising since in the ontological model presented here, relations are established among concepts. An analysis by domain nature can also be performed, since similar domains have produced similar results. For the second experiment, a group of experts built individual ontologies on the computers domain and two integration processes were carried out. The

ontologies used in the second integration process were refinements of the defined for the first iteration. The purpose of this experiment was to analyse the effects of integration processes on ontology refinement processes and how they can help to achieve consensus among experts.

The second objective of this validation was to evaluate the quality of the ontologies built. For such purpose, four heterogeneous application domains were selected and thanks to the direct participation of experts, different ontologies have been cooperatively built. After the construction process, the quality of the ontologies was evaluated. For such evaluation, the quality criteria mentioned in (Crow and Shadbolt, 1999), namely, precision, breadth, consistency, completeness, and utility, were employed. It was shown that the ontologies built by using the ontological model and the integration framework held all these criteria so that it can be said that the framework proposed in this thesis generates ontologies holding such quality criteria.

In order to end with the pure validation, some general conclusions can be drawn from the experiments carried out. First, direct interaction with experts facilitates content focalization. This interaction permits to understand better the domain and to avoid the appearance of spurious attributes or taxonomic relations. Second, the good results obtained through this validation of the integration framework constitute a partial justification of the achievement of the initial goals.

The second part of the validation of the framework corresponds with the description of different applications of the integration framework. Two applications have been presented in this part of the chapter. The first one corresponds to the cooperative construction of molecular biology ontologies which demonstrate how the framework presented has been used in order to build two different partial ontologies, one concerning the cell and one concerning the protein. Since the objective of this thesis was not building particular ontologies on a specific domain, the ontologies built were large ones. These ontologies were built by the experts with the aim of assessing the utility of the integration processes in ontology construction processes. The second application of the ontological model and the integration framework was as a part of a framework for facilitating knowledge management in Intensive Care Units (ICUs).

# THE SYSTEM

An ontology integrator software tool has been presented. Such tool fulfils the requirements imposed by the integration framework, so that, this tool is appropriate for integrating ontologies according to the ontological integration framework proposed. In particular, the tool allows for the collaborative construction of ontologies by reusing already existing ontologies and integrating all of them into a single, global ontology. We can summarise the adequacy of the ontology integration tool as follows:

- **Privacy**: As it was aforementioned in the thesis, each non-integration-derived ontology is private and only the expert who built it is given access to it. This feature is guaranteed by the tool since users must log into the system and ontologies are stored into the system by keeping track of the one who created which ontology.

- **Knowledge sharing and reuse**: The ontology integration framework promotes such features because existing ontologies are reused in order to build the global one, and the knowledge created by each expert is available to the community through integration processes. The tool makes use of the ontologies created by the different expert users in order to launch integration processes when these are requested by users.

- **Maximum information content**: The ontological framework is aimed at providing users with the maximum possible knowledge. The tool has been designed to meet such requirement.

- **User dependence:** The ontology integration framework customises the terminology to the one used by the user in case such user has built ontologies for the integrating topic. The tool also customises such terminology, therefore meeting this framework objective as well.

- **User distribution:** The integration framework supposes that the set of expert users are geographically distributed. This requirement is met by the tool because the tool has been designed and implemented to work on an intranet/internet.

- **Ontology construction**: The ontological framework specifies the processes for building individual ontologies and integrating those ontologies. A simple language for specifying ontologies has also been proposed in this thesis and such language has been incorporated into the ontology integration tool. Moreover, integrated ontologies are also expressed in such language, so that these new ontologies can be modified by using the same ontology edition tools.

Two versions of the tool have been presented. The first one was a classical application in which the user launches the client software in order to work with the ontology integration tool. The second version is web-oriented, since a web site has been provided for the tool and users can request for integration directly from their web browsers. Users have the same facilities with both tools, although the construction of the web ontology integration interface can allow users to have access to the system in an easier manner.

# CONCLUSIONS AND FUTURE WORK

This work is an attempt to simplify collaborative knowledge creation processes. There exist different approaches for performing such processes, although none of them can be considered to be successful, since all of them present significant drawbacks. Examples of such drawback are the reduced knowledge models for which these approaches have been designed or the need for blocking experts when other experts are modifying the collective ontology. Furthermore, the analysis of the existing approaches reveals the existence of two trends: (a) collaborative development of a common ontology; and (b) ontology merging. Amongst the approaches belonging to the first trend, Ontolingua (Farquhar et al, 1997), WebOnto (Domingue, 1998) and OntoSaurus (ISX, 1991) can be highlighted. Amongst the ontology

integration methodologies, PROMPT (Fridman-Noy and Musen, 2000), Chimaera (McGuiness et al, 2000) and FCA-Merge (Stumme and Maedche, 2001), are the most well known. However, such approaches are not automatic, so that the participation of experts is obliged at almost every step of the process. FCA-Merge is the approach which performs automatically more steps. In particular, expert participation is only requested for merging the ontology from the automatically created concept lattices. Moreover, no knowledge personalisation is really provided by these approaches, so that experts cannot access the global knowledge in a customised manner. Therefore, there is a need for solutions for the collaborative knowledge construction. All these factors led us to propose a methodology for developing ontologies in a cooperative way that performs automatically knowledge integration processes, and that allows for accessing the knowledge in a personalised manner.

The methodology here proposed provides a formal framework for developing ontologies in a cooperative way and based on the integration of knowledge pieces (i.e., ontologies). The objective of these integration processes is to obtain ontologies which contain the largest and best knowledge portion so that these ontologies can be considered appropriate for the particular domain being treated. Once good ontologies have been built, these can be used for developing knowledge management systems. In fact, another objective of this work is to supply developers of knowledge management systems with an efficient and powerful tool for representing the knowledge necessary for the system. Let us consider now how the ontologies can be used in knowledge management systems. According to O'Leary (O'Leary, 1999), knowledge management systems must facilitate:

- Conversion of data and text into knowledge. On the one hand, data and text have to be converted into knowledge. On the other hand, the ontology is the most appropriate technology for representing knowledge. Therefore, ontologies are useful in this step.

- Conversion of individual and group's knowledge into accessible knowledge. Ontologies can be used for representing both individual and group's knowledge. Moreover, ontology integration processes can enhance this process.

- Connection of people and knowledge to other people and other knowledge. The richness of the semantics associated to ontologies would increase the efficiency of the process.

- Communication of information between different groups. In principle, ontologies does not attempt to solve communication issues.

- Creation of new knowledge that would be useful for organisations. Ontologies should be used for this purpose, and if knowledge comes from different sources, ontology integration processes can be applied.

The main apportions of this thesis are the following:

- A formal ontology model has been specified. The model is comprised of concepts, (inherited and specific) attributes, (taxonomic and mereological) relations, and structural axioms.

- A formal framework for integrating and personalising knowledge has been also specified in this work. Through this framework, a set of ontologies specified by domain experts and covering the same domain are automatically integrated into a single, common ontology. The resulting knowledge is personalised according to the individual who requested for the integration.

- A system implementing the previous framework has been developed. This system allows for the collaborative construction of ontologies through the performance of ontology integration processes.

- An XML DTD has been defined to facilitate the interoperability of our system and ontologies with other ontology management tools.

The previous elements have been validated in different ways. First, the ontology construction process was validated by building different ontologies in different domains and reaching similar conclusions in each case. Second, the validity and usefulness of the integration approach was validated in two manners. The first one was based on analysing the results obtained in different integration processes in different domains, reaching similar conclusions. The second one was performed by asking different experts to build an ontology in a common domain, then performing integration iterations and analysing the findings obtained. Third, the integration framework was successfully applied in a biological domain in order to build an ontology in a cooperative manner. Finally, the integration framework has been used as the kernel for a system for managing knowledge in Intensive Care Units.

Regarding future work, different issues should be pointed out. First, the ontology model could be improved in several ways. In fact, the current model does not allow for the definition of non-structural axioms. Nevertheless, this is not the unique improvement that can be made concerning the ontology model. More relations could be added to the model. In particular, an extension to the original model, which contains taxonomic and mereological relations, is being designed to cover temporal relations. In (Gómez-Pérez et al, 2000), the most common types of relations are identified. It would be desirable to cover at least those types of relations in the model.

Another possible improvement to the ontology model is concerned with providing internal structure to attributes. Currently, attributes are just strings identified by their name, whereas it would be desirable to have attribute values and different types of attributes.

Another thread for improving the work here presented is related to the ontology integration processes. Here, inconsistent knowledge is discarded from the integration process. Since ontologies can only be partially inconsistent, it would be better to discard only the inconsistent pieces of the ontologies rather than the complete ontology. Thus, source ontologies would be modified in such a way that the inconsistent parts would be pruned. As it happens now, only consistent knowledge would take part in integration processes. This improvement could by made by adding an extra step to the ontology integration process. This step could be

allocated after selecting the best subset of candidate ontologies because this subset would contain the highest amount of compatible ontologies, so that only the non-included ones would have to be modified in order to be compatible with the rest of ontologies.

In summary, the formal framework for integrating ontologies presented in this thesis can be situated in the context of Ontological Engineering and Knowledge Management, provided that the results obtained are significant for both research areas. First, it is significant for Ontological Engineering because it provides an ontology model and an ontological framework that allows for collaborative ontology development. The development of ontologies is a hard task, so that such collaborative and, moreover, automatic development is highly beneficial for this research field. Finally, this knowledge construction methodology helps to improve the development of Knowledge Management Systems as it has been explained before, since the usage of the framework presented in this thesis simplifies most of the knowledge-related issues this type of systems possesses.

# REFERENCIAS

## Referencias


Abecker, A., Aitken, S., Schmalhofer, F., Tschaitschain B. (1998). KARATEKIT: Tools for the Knowledge-Creating Company. In *Proceedings of the 11th Banff Knowledge Acquisition for Knowledge Based Systems Workshop, vol. 2, KM-1.1-KM-1.18*, Banff, Canada.

Albertrazzi, L (1996) "Material and Formal Ontology", in R. Poli, P. Simons (eds.), *Formal Ontology:199-232*, Kluwer, Dordrecht.

Altman, R.B., Valencia, A., Miyano, S., Ranganathan. S. (2001) Challenges for Intelligent Systems in Biology. *IEEE Intelligent Systems 16(6), 14-20*.

Andrade, M.A., Valencia, A. (1998): Automatic extraction of keywords from scientific text: application to the knowledge domains of protein families. *Bioinformatics 14(7): 600-607*.

Arpírez, J. C., Gómez-Pérez, A., Lozano, A., Pinto, H. S. (1998) (ONTO)2Agent: An ontology-based WWW broker to select ontologies. *Workshop on Applications of Ontologies and Problem-Solving Methods. European Conference on Artificial Intelligence (ECAI'98):16-24*. Brighton (United Kingdom).

Artale, A., Franconi, A., Guarino, N., Pazzi, L. (1996). Part-Whole Relations in Object-Centered Systems: An Overview. *Data and Knowledge Engineering.* North Holland, Elsevier.

Aussenac-Gilles, N., Biebow, B., Szulman, S (2002) Modelling the travelling domain from a NLP description with Terminae. *Workshop on Evaluation of Ontology Tools, European Knowledge Acquisition Workshop*, Sigüenza, Spain.

Baker, P.G., Goble, C.A., Bechhofer, S., Paton, N.W., Stevens, R., Brass, A. (1999) An ontology for bioinformatics applications. *Bioinformatics 15(6): 510-520*.

Baumgarten, A.G. (1740) *Metaphysica*.

Beckman, T. (1997a). A Metodology of Knowledge Management. In Proceedings of the International Association of Science and Technology for Development (IASTED) AI and Soft Computing Conference. Banff, Canada.

Beckman, T. (1997b). Expert System Applications: Designing Innovative Business Systems through Reengineering. *Handbook on Expert Systems.* Liebowitz, J. ed. CRC Press.

Beckman, T. (1998b). Designing Innovative Business Systems through Reengineering. Tutorial. *Fourth World Congress on Expert Systems.* Mexico City.

Benjamins, V.R., Fensel, D. (1998). Community is Knowledge! in $(KA)^2$. In Proceedings of the 11th Banff Knowledge Acquisition for Knowledge Based Systems Workshop, vol. 2, KM-2.1-KM-2.18, Banff, Canada.

Benjamins, V.R., Fensel, D., Gómez-Pérez, A., Decker, S., Erdmann, M., Motta, E., Musen, M. (1998) The Knowledge Annotation Initiative of the Knowledge Acquisition Community $(KA)_2$. In:

*Proceedings of the 11th Banff Knowledge Acquisition for Knowledge-Based System Workshop (KAW 98)*, Banff, Canada.

Bernaras, A., Laresgoiti, I., Corera, J. (1996) Building and Reusing Ontologies for Electrical Network Applications. In *Proceedings of the European Conference on Artificial Intelligence (ECAI96):298-302*.

Berners-Lee, T. (2001) The Semantic Web. *The Scientific American*, May.

Borgo, S., Guarino, N., Masolo, C. (1996) Stratified Ontologies: the case of physical objects. In *Proceedings of the Workshop on Ontological Engineering. Held in conjunction with ECAI96. Pages 5-15*. Budapest.

Borst, W.N (1997). *Construction of Engineering Ontologies for Knowledge Sharing and Reuse*. CTIT Ph.D-thesis series No.97-14. University of Twente. Enschede, The Netherlands.

Brooking, A. (1999) Corporate Memory: Strategies for Knowledge Management. International Thomson Business Press.

Bylander, T., Chandrasekaran, B. (1988). Generik tasks in knowledge-based reasoning: The right level of abstraction for knowledge acquisition. In B.R. Gaines and J. Boose (Eds.), *Knowledge Acquisition for Knowledge Based Systems, Volume 1:65-77*. London Academic Press.

Campoy-Gómez, L. (2002) Esquema formal para la integración de componentes reutilizables y compartibles de conocimiento en las gestión de memorias corporativas. Tesis Doctoral, Universidad de Murcia, España.

Caussanel, J., Chouraqui, E. (1999) Model and Methodology of Knowledge Capitalization for Small and Medium Enterprises. *Proceedings of Knowledge Acquisition Workshop'99*.

Chandrasekaran, B., Johnson, T. R., Benjamins, V. R.(1999) Ontologies: what are they? why do we need them?. *IEEE Intelligent Systems and Their Applications. 14(1). Special Issue on Ontologies:20-26*.

Chaudhri, V.K., Farquhar, A., Fikes, R., Karp, P.D., Rice, J.P. (1998) *Open Knowledge Base Connectivity 2.0.* Knowledge Systems Laboratory, January, 1998.

Cocchiarella, N. (1995) "Knowledge Representation in Conceptual Realism", in *International Journal of Human-Computer Studies, vol. 43 (1995):697-721*.

Couturat, L. (1903) Opuscules et fragments inédits de Leibniz, Paris, France.

Crow, L., Shadbolt, N. (2001). Extracting focused knowledge from the semantic web. *International Journal of Human-Computer Studies, 54:155-184*.

Davis, R., Sorbe, H., Szolovits, P. (1993). What is a Knowledge Representation? *AI Magazine. Spring, 17-33*.

## Referencias


Deutsch, R.W. (1999). *Knowledge Management Handbook*. Jay Liebowitz Ed. CRC Press LLC.

Dignum, V., Heimannsfeld, K. (1999) Knowledge management for requirements engineering. *Procedings of Knowledge Acquisition Workshop (KAW'99)*, 1999.

Dojat, M., Pachet, F., Guessoum, Z., Touchard, D., Harf, A., Brochard, L. (1997) NéoGanesh: a working system for the automated control of assisted ventilation in ICUs. *Artificial Intelligence in Medicine 11:97-117*.

Dojat, M., Ramaux, N., Fontaine, D. (1998) Scenario recognition for temporal reasoning in medical domains. *Artificial Intelligence in Medicine 14: 139-155.*

Domingue, J. (1998). Tadzebao and WebOnto: discussing, browsing, and editing ontologies on the web. In *Proceedings of the 11th Workshop on Knowledge Acquisition, Modelling and Management, KAW'98*, Banff, Canada.

Duineveld, A.J., Stoter, R., Weiden, M.R., Kenepa, B., Benjamins, V.R. (2000). WonderTools? A comparative study of ontological engineering tools. *International Journal of Human-Computer Studies 52:1111-1133*

Eriksson, H., Fergeson, R., Shahar, Y., Musen, M.A. (1999). Automatics generation of ontology editors. In *Proceedings of 12th Banff Workshop on Knowledge Acquisition, Modelling and Management:4.4*

Eschenbach, C., Heydrich, W. (1995). Classical mereology and restricted domains, *International Journal of Human-Computer Studies*, *43: 723-740.*

Euzenat, J. (1996). Corporate Memory through Cooperative Creation of Knowledge Based Systems and Hyper-Documents. In *Proceedings of the 10th Workshop on Knowledge Acquisition, Modelling and Management: 36.1-36.20*, Banff, Canada.

Farquhar, A., Fikes, R. & Rice, J. (1997). The Ontolingua Server: a tool for collaborative ontology construction. *International Journal of Human-Computer Studies 46, 707-727*

Farquhar, A., Fikes, R.E., Rice, J. (1996). The Ontolingua server: A tool for collaborative ontology construction. In B. R. Gaines and M.A. Musen (Eds.), *Proceedings of the 10th Banff Knowledge Acquisition for Knowledge-based Systems Workshop, Vol.2*, University of Calgary, pp. 44-1-44-19. SRDG Publications.

Fensel, D., Decker, S., Erdmann, M., Studer, M. (1998). Ontotobroker: Or How to Enable Intelligent Access to the WWW. In *Proceedings of the 11th Banff Knowledge Acquisition for Knowledge Based Systems Workshop, vol. 2, SHARE-8.1-SHARE-8.18*, Banff Canada

Fensel, D., Musen, M.A. (2001). A Brain for Mankind. *IEEE Intelligent Systems 16(2):24-25*.

Fernández, M., Gómez-Pérez, A., Juristo, N. (1997) METHONTOLOGY: From Ontological Art to Ontological Engineering. Workshop on Ontological Engineering. Spring Synposium Series de la AAAI (American Association for Artificial Intelligence):33-40. Stanford, USA.

Fernández-Breis, J.T., Martínez-Béjar, R. (2000a). A cooperative tool for facilitating Knowledge Management. *Expert Systems with Applications 18(4):315-330.*

Fernández-López, M., Gómez Pérez, A., Rojas Amaya, M. D. (2000). Ontologies crossed life cycles. *Workshop on Knowledge Acquisition, Modelling and Management (EKAW):65-79*. Editor Springer Verlag. Jean Les Pins (Francia).

Fernández-López, M., Gómez-Pérez, A., Pazos-Sierra, A., Pazos-Sierra, J. (1999) Building a Chemical Ontology Using METHONTOLOGY and the Ontology Design Environment. *IEEE Intelligent Systems & their applications. January/February Pages 37-46.*

Ferrater Mora, José (1963). On the Early History of Ontology. *Philosophy and Phenomenological Research 24: 36-47*.

Francoforti (1613) Lexicon philosophicum, quo tanquam clave philosophiae fores aperiuntur, Informatum opera studio Rodolphi Goclenii.

Fridman-Noy, N., Hafner, C.D. (1997). The State of the Art in Ontology Design. A Survey and Comparative Review, *AI Magazine, Fall 1997*.

Fridman-Noy, N., Hafner, C.D. (1997). The State of the Art in Ontology Design. A Survey and Comparative Review, *AI Magazine, Fall 1997*.

Fridman-Noy, N., Hafner, C.D.(2000) Ontological Foundations for Experimental Science Knowledge Bases. *Applied Artificial Intelligence **14(6)**, 565-618*.

Fridman-Noy, N., Musen, M.A. (1999). An Algorithm for Merging and Aligning Ontologies: Automation and Tool Support. 16th *National Conference on Artificial Intelligence, Workshop on Ontology Management,* Orlando, FL.

Fridman-Noy, N., Musen, M.A. (2000). PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. 17th *National Conference on Artificial Intelligence*, Austin, Texas.

Gangemi, A., Pisanelli, D.M., Steve, G., (1999) An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies. *Data and Knowledge Engineering 31(2): 183-220*.

Ganter, B., Wille, R. (1999) Formal Concept Analysis: Mathematical foundations. Springer, Berlin-Heidelberg 1999.

Garcia, C. (1996). Cooperative Building of an Ontology within Multi-Expertise Framework. In *Proceedings of COOP'96, p. 435-454*, Juan-Les-Pains, France.

Gerstl, P., Pribbenow, S. (1995). Midwinters, end games and body parts: A cassification of part-whole relations. *International Journal of Human-Computer Studies, 43, 865-889*.

Gilmour, R.(2000): Taxonomic markup language: applying XML to systematic data *Bioinformatics 16: 406-407*.

## Referencias


Giudicelli, V., Lefranc M.P. (1999): Ontology for immunogenetics: the IMGT-ONTOLOGY. *Bioinformatics 15(12): 1047-1054*.

GO (2000). GO General Documentation. http://www.geneontology.org.

Gómez Pérez, A. (1999) Evaluation of Taxonomic Knowledge in Ontologies and Knowledge Bases. *Twelfth Workshop on Knowledge Acquisition, Modeling and Management (KAW'99)*. Alberta (Canada).

Gómez-Pérez, A. (2001) Evaluation of Ontologies. International Journal of Intelligent Systems vol 16:391-409.

Gómez-Pérez, A. Moreno, A., Pazos, J., Sierra-Alonso, A.. (2000) Knowledge Maps: An essential technique for conceptualisation, *Data & Knowledge Engineering, 33(2), 169-190*.

Gómez-Pérez, A., Benjamins, V.R. (1999). Overview of knowledge sharing and reuse components: ontologies and problem-solving methods. In V.R. Benjamins, B.Chandrasekaran, A.Gómez-Pérez, N.Guarino and M.Uschol.d (Eds), *Proceedings of the IJCAI-99 workshop on Ontologies and Problem-Solving Methods*, Stockholm, Sweden.

Gómez-Pérez, A., Corcho, O.(2002) Ontology languages for the Semantic Web. *IEEE Intelligent Systems 17(1):54-60*.

Gruber, T. R. (1993) "A translation approach to portable ontology specifications". *Knowledge Acquisition Vol. 5:199-220*.

Gruber, T. R. (1995) Towards Principles of the Design of Ontologies Used for Knowledge Sharing. *International Journal of Human Computer Studies. 43:907-928.*

Gruninger, M. Fox, M.S. (1995). The logic of enterprise modelling. In J. Brown & D.O. Sullivan, Eds. *Reenginering the Enterprise: 83-98*. London: Chapman & Hall.

Grunstein, M. (1995) La capitalisation des connaissances de l'enterprise, système de production de connaissances. *L'enterprise apprenante et les Sciences de la Complexité*. Aix-en-Provence.

Guarino, N. (1995). Formal Ontology, Conceptual Analysis and Knowledge Representation. *International Journal of Human and Computer Studies, 43(5/6): 625-640*

Guarino, N. (1998). Formal Ontology and Information Systems. *Proceedings of FOIS'98: 3-15*

Guarino, N. (1999). The Role of Identity in Ontology Design. *Proceedings of the Workshopon Ontologies and Problem Solving Methods (KRR5)*. Stockholm, Sweden. August

Guarino, N., Carrara, M., Giaretta, P. (1994). Formalizing Ontological Commitment. In *Proceedings of National Conference on Artificial Intelligence (AAAI-94):560-567*. Seattle, Morgan Kaufmann.

Guarino, N., Welty, C. (2000). Identity, Unity, and Individuality: Towards a Formal Toolkit for Ontological Analysis. In H. Werner (ed.) *ECAI-2000: The European Conference on Artificial Intelligence:219-223*. IOS Press, Berlin, Germany.

Guarino, N., Welty, C. (2001). Identity and subsumption. In R. Green, C. A. Bean, and S. Hyon Myaeng (eds.), *The Semantics of Relationships: An Interdisciplinary Perspective,* Kluwer.

Guarino, N.; Welty, C. (2000) A Formal Ontology of Properties. In R. Dieng and O. Corby (eds.), Knowledge Engineering and Knowledge Management: Methods, Models and Tools. 12th International Conference, EKAW2000:97-112. Springer Verlag.

Guerrini, V. H. and Jackson, D. (2000): Bioinformatics and extended markup language (XML), *Online Journal of Bioinformatics 1:1-13*.

Hameed, A., Sleeman, D., Preece, A. (2001). Detecting Mismatches Among Experts' Ontologies Acquired through Knowledge Elicitation. In *Proceedings of 21st SGES International Conference on Knowledge Based Systems and Applied Artificial Intelligence*, Cambridge, U.K.

Hibbard, J. (1997). Knowing What We Know. *Information Week*. October 20.

Horrocks, I., Fensel, D., Broekstra, J., Decker, S., Erdmann, M., Goble, C., van Harmelen, F., Klein, M., Staab, S., Studer, R., Motta, E. (2000) *OIL: The Ontology Inference Layer*. Technical Report IR-479, Vrije Universiteit Amsterdam, Faculty of Sciences.

Horrocks, I., van Harmelen, F. (2001) *Reference Description of the DAML+OIL OntologyMarkup Language*, draft report, www.daml.org/2000/12/reference.html (current Jan. 2002).

http://www.formalontology.it

Huber, G. (1991) Organizational Information Systems: Determinants of their Performance and Behaviour. *Management Science 28:135-155*.

ISX Corporation (1991). *LOOM Users guide*, version 1.4

Jovanovic, A. (1998). Building and Practical Implementations of Corporate Memory Systems: from Basic Idea to the Full-Scale System. Tutorial. Presented at the *4th World Congress on Expert Systems.* Mexico City.

Kant, I. (2001) (Published) *Lectures on metaphysics - Part III Metaphysik L2*. Cambridge University Press.

Karp, P.D. (1993): A qualitative biochemistry and its application to the regulation of the trytophan operon. *Artificial Intelligence and Molecular Biology: 289-325*.

Karp, P.D. (2000): An ontology for biological function based on molecular interactions. *Bioinformatics 16(3): 269-285.*

Karp, R., Chaudhri, V., Thomere, J. (1999) *XOL: An XML-Based Ontology Exchange Language*

*(version 0.4)*, www.ai.sri.com/~pkarp/xol (current Jan. 2002).

Kent, P. (1998) *Conceptual Knowledge Markup Language (version 0.2)*. www.ontologos.org/CKML/CKML%200.2.html (current Jan. 2002).

Knight, K., Luk, S. (1994). Building a Large Base for Machine Translation. In Proceedings of the American Association of Artificial Intelligence Conference (AAA-94):773-778. Seattle, WA.

Kremer, S. (2001). Spatio-Temporal Connextionist Networks: A Taxonom and Review. *Neural Computation 13:249-306*.

Lassila, O., Webick, R. (1999) *Resource Description Framework (RDF) Model and Syntax Specification.* W3C Recommendation, Jan. 1999,www.w3.org/TR/PR-rdf-syntax (current Jan. 2002).

Lenat, D.B., Guha, R.V. (1990) *Building large knowledge-based systems*. Addison-Wesley Publising Company, Inc. 1990.

Liebowitz, J. and Beckman, T. (1998). *Knowledge Organizations: What Every Manager Should Know*. St. Lucie Press.

Lindberg, D.A.B., Humphrey, B.L., McCray, A.T. (1993). The Unified Medical Language System. *Methods of Information in Medicine. 32, 281-291*.

Luke, S., Heflin, J. (2000) *SHOE 1.01 ProposedSpecification, SHOE Project*, Feb. 2000. www.cs.umd.edu/projects/plus/SHOE/spec1.01.htm (current Jan. 2002).

Macintosh, A. (1996). *Position Paper on Knowledge Assets Management*. Artificial Intelligence Applications Institute, University of Edimburgh, Scotland. May 1996

Maedche, A., Staab, S. (2001) Ontology learning for the Semantic Web. *IEEE Intelligent Systems, 16(2):72-79*.

Mahesh, K. (1996). *Ontology Development for Machine Translation: Ideology and Methodology*. New Mexico State University, Computing Research Laboratory, MCC-96-292.

Martín-Rubio, F. (1998). *La gestión del conocimiento corporativo: una tecnología emergente*. Servicio de Publicaciones. Universidad de Murcia. España.

McGuiness, D.L., Fikes, R., Rice, J., & Wilder, S. (2000). An environment for merging and testing large ontologies. In Anthony Cohn, Fausto Giunchiglia, and Bart Selman, editors, KR2000: *Principles of Knowledge Representation and Reasoning:483-493*, San Francisco, USA.

McGuinness, D.L., Fikes, R., Rice, J., Wilder, S. (2000). An Environment for Merging and Testing LargeOntologies. In *Proceedings of Knowledge Representation*.

Meinong. A (1921) Self-presentation - published in: Raymond Schmidt (ed.) *Die Deutsche Philosophie der Gegenwart in Selbstdarstellung - vol. I*

Miller, G. A. (1995) WordNet: a lexical database for English. *Communications of the ACM 38 (11): 39 – 41*.

Mizoguchi, R., Vanwelkenhuysen, J., Ikeda, M. (1995). Task Ontology for Reuse of Problem Solving Knowledge. *Towards Very Large Knowledge Bases: KnowledgeBuilding and Knowledge Sharing: 46-59.*

Muller, H.J., Schappert, A. (1999) The Knowledge Factory – A Generic Knowledge Management Architecture. Proceedings. of *International Joint Conference on Artificial Intelligence*, 1999

Nagendra Prassad, M.V.N., Plaza, E. (1996) Corporate Memories as Distributed Case Libraries, *Proceedings of Knowledge Acquisition Workshop*, Banff, Alberta, Canada, November 9-14, 1996, p. 40-1 40-19.

Nakata, K., Voss, A., Juhnke, M., Kreifelts, T. (1998). Knowledge Extraction, Integration and Management. In *Proceedings of the Second International Conference on Practical Aspects of Knowledge Management (PARM '98): 20-1, 20-11*, Bassel, Switzerland.

National Library of Medicine (1999). *UMLS Knowledge Sources*, available from the NLM, Bethesda, Maryland.

Neches, R., Fikes, R.E. Finin, T., Gruber, T.R., Senator, T., Swartout, W.R. (1991) "Enabling technology for knowledge sharing". *AI Magazine. 12(3):36-56*.

Newell, A. (1982) The Knowledge Level. *Artificial Intelligence, 18 (1)*.

Nonaka, I. (1994) A Dynamic Theory of Organizational Knowledge Creation. *Organization Science 5(1):14-37*.

Nonaka, I., Takeuchi, H. (1995). The Knowledge Creating Company: How Japanese Companies Create the Dynamics of Innovation. Oxford University Press.

O'Leary, D.E. (1999). Reengineering and Knowledge Management. In Fensel and Studer (Eds.) Knowledge Acquisition, Modeling and Management, *Lecture Notes in Artificial Intelligence 1621 1-12*, Springer, Germany

O'Dell, C. (1996). A Current Review of Knowledge Best Practice. Conference on Knowledge Management and the Transfer of Best Practices. Business Intelligence, London. December 1996

Petrash, G. (1996). Managing Knowledge Assets for Value. Knowledge-Based Leadership Conference. Linkage Inc., Boston. October 1996.

Pinto, H.S., Martins, J.P. (2001). Ontology Integration: How to perform the Process. In Proceedings of International Joint Conference on Artificial Intelligence, Seattle, Washington, USA

Pisanelli, D., Gangemi, A., Steve, G. (1998). An Ontological Analysis of the UMLS Methatesaurus. Proceedings of AMIA 98 Conference, 1998.

## Referencias


Poli, R. (2000) Levels of Reality. BISCA 2000: Bolzano International Schools in Cognitive Analysis "Dependence and Dynamic Categories".

Poli, R. (2001) Alwis: Ontology for Knowledge Engineers, PhD thesis, Utrecht

Poli, R. (2002) Ontological methodology. International Journal of Human-Computer Studies, 56(6):639-664.

Poli, R. (2002a) Descriptive, Formal, and Formalized Ontologies. In D. Fisette, ed., *Edmund Husserl's Logical Investigations 1901-2001: Origins and Posterity of Phenomenology*, Kluwer.

Quine, W.O. (1961). *From a Logical Point of View, Nine Logico-Philosophical Essays*. Cambridge, Massachusetts: Harvard University Press

Rector, A., Solomon, W., Nowlan, W., Rush, T. (1995). A Terminology Server for Medical Language and Medical Information Systems. *Methods of Information in Medicine, Vol. 34, 147-157.*

Reimer, U. (1998). Knowledge Integration for Building Organizational Memories. In Proceedings of the 11th Banff Knowledge Acquisition for Knowledge Based Systems Workshop, vol 2, KM-6.1-KM-6.20, Banff, Canada.

Rice, J., Farquhar, A., Piernot, P., Gruber, T. (1996). Lessons learned using the web as an application interface. In *CHI96*.

Schulz, S., Hahn, U. (2001) Parts, Locations and Holes-Formal Reasoning about Anatomical Structures. *Lecture Notes in Artificial Intelligence 2101: 293-303*.

Shadbolt, N.R., O'Hara, K., Crow, L. (1999). The experimental evaluation of knowledge acquisition techniques and methods: history, methods, and new directions, *International Journal of Human-Computer Studies, 51:729-755*.

Shahar, Y (1997), A framework for knowledge-based temporal abstraction*, Artificial Intelligence 90(1-2):79-133*

Shahar, Y. (1999) Timing is everything: Temporal reasoning and temporal data maintenance in medicine. *Joint European Conference on Artificial Intelligence in Medicine and Medical Decision Making:30-46*. AIMDM'99, Aalborg, Denmark.

Shahar, Y. (2000) Dimension of Time in Illness: An Objective View. *Annals of Internal Medicine 132 (1):45-53*.

Shahar, Y., Miksch, S., Johnson, P. (1998) The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines. *Artificial Intelligence in Medicine 14:29-51*.

Shahar, Y., Musen, M.A. (1996) Knowledge-Based Temporal Abstraction in Clinical Domains. *Artificial Intelligence in Medicine, 8(3): 267-298*.

Shaw, M.L.G., Gaines, B.R. (1989). A Methodology for recognising conflict, correspondence, consensus and contrast in a Knowledge Acquisition System. *Knowledge Acquisition, 1(4):341-363*.

Simons, P. (1987). *Parts: A study in Ontology.* Clarendon Press, Oxford.

Smith, B. (1978) An essay in formal ontology, *Grazer Philosophische Studien 6: 39-62*.

Smith, B., Mulligan, K. (1983). Framework for Formal Ontology. *Topoi 2:73-85*.

Sowa, J. F. (1984). *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley Publishing Company, Inc., USA.

Sowa, J.F. (2000) *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, Brooks Cole Publishing Co., Pacific Grove, CA.

Spender, C. (1996) Organizational Knowledge, Learning and Memory: Three Concepts in Search of a Theory, *Joumal of Organizational Change Management 9, 63-78*.

Staab, S., Schnurr, H.P., Studer, R., Sure, Y.(2001) Knowledge Processes and Ontologies, *IEEE Intelligent Systems, 16(1)*.

Steels, L. (1993) Corporate Knowledge Management. Proceedings of *International Symposium on the Management of Industrial and Corporate Knowledge,* Compiègne, France, pp 223-235

Stein, E. and Zwass, V. (1995) Actuallizing Organizational Memory with Information Technology. *Information Systems Research*, 6:2, 85-117, 1995.

Studer, R., Benjamins, R., Fensel, D. (1998) Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering 25(1-2):161-197*.

Stumme, G., Maedche, A. (2001). FCA-Merge: A Bottom-Up Approach for Merging Ontologies JCAI '01 - *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, Seattle, USA, August, 1-6, 2001, San Francisco/CA: Morgen Kaufmann

Swartout, B., Patil, R. Knight, K., Russ, T. (1997) Toward distributed use of large-scale ontologies. In *AAAI-97 Spring Symposium Series on Ontological Engineering*.

Tennison, J., Shadboldt, N. (1998). APECKS: a Tool to Support Living Ontologies. In *Proceedings of the 11th Workshop on Knowledge Acquisition, Modelling and Management*, Banff, Canada.

Tobin, D. (1998). The Knowledge-Enable Organization: Moving from Training to Learning to Meet Business Goals. *AMACOM*.

Turban, E. (1992) Expert Systems and Applied Artificial Intelligence. Macmillan

Uckun, S., Dawant, B.M., Lindstrom, D. (1993) Model-based diagnosis in intensive care monitoring: The YAQ approach. Artificial Intelligence in Medicine 5:31-48.

Uschold, M., Groninger, M. (1996) Ontologies: Principles Methods and Applications. *Knowledge Engineering Review. Vol. 2.*

Uschold, M., King, M. (1995) Towards a Methodology for Building Ontologies. *Workshop on Basic Ontological Issues in Knowledge Sharing*.

Van der Spek, R., Spijkervet, A. (1997). Knowledge Management: Dealing Intelligently with Knowledge. *Knowledge Managemente and Its Integrative Elements*. Liebowitz & Wilcox, eds. CRC Press

Van Heijst, G., Van der Spek, R., Kruizinga, E. (1996) Organizing Corporate Memories. In B.Gaines, M.Musen eds, *Proceedings of Knowledge Acquisition Workshop*, 1996, Banff, Canada, November, pp.42-1 42-17.

Van Heijst, G., Schreiber, A. T., Wielinga, B. J. (1997). Using explicit ontologies in KBS development, *International Journal of Human-Computer Studies*, *45: 183-292*.

Visser, P.R.S., Jones, D.M., Bench-Capon, T.J.M., Shave, M.J.R. (1998). Assessing Heterogeneity by Classifying Ontology Mismatches, Formal Ontology in Information Systems. In *N. Guarino (ed.), Proceedings FOIS'98, Trento, Italy), IOS Press, Amsterdam, The Netherlands, pp.148-162*.

Welty, C., Guarino, N. (2001) Supporting Ontological Analysis of Taxonomic Relationships. *Data and Knowledge Engineering 39(1):51-74*.

Westbrook, J.D., Bourne, P.E. (2000): STAR/mmCIF: An ontology for macromolecular structure. *Bioinformatics 16(2): 159-168*.

Wiederhold, G. (1994). Interoperation, Mediation and Ontologies. In *Proceedings of FGCS'94 Workshop on Heterogeneous Cooperative Knowledge-Bases:33-48*, Tokyo, Japan.

Wielinga, B.J., Schreiber, A.T., Breuker, J.A. (1992). KADS: A Modelling Approach to Knowledge Engineering. *Knowledge Acquisition Journal, 4(1): 5-53*.

Wiig, K. (1993). Knowledge Management Foundations: Thinking about Thinking: how people and organizations create, represent, and use knowledge. Schema Press, Arlington, Texas

Wiig, K. (1997). Knowledge Management: Where Did It Come From and Where Will It Go?. *Expert Systems with Applications Vol. 14.*

Wille, R. (1982) Restructuring lattice theory: an approach based on hierarchies of concepts. In: I. Rival (ed.): *Ordered sets: 445-470*. Reidel, Dordrecht-Boston.

Winskel, G. (1994). Foundations of computing: The formal semantics of programming languages. MIT Press, USA.

Winslow, C. and Bramer, W. (1994). Future Work: Putting Knowledge to Work in the Knowledge Economy..

Winston, M, Chaffin, R., Hrrmann, D. (1987). A taxonomy of part-whole relations. *Cognitive Science 11, 417-444*.

Woolf, H. (1990). Webbsters New World Dictionary of the American Language, G. And C. Merrian.

Yu-N, C., Abidi, S.S.R. (1999) Evaluating the Efficacy of Knowledge Management Towards Healthcare Enterprise Modelling. In Proceedings of the International Joint Conference on Artificial Inteligence.

Zarri, G. (1996). Building up and Making Use of Corporate Knowledge Repositories. *KAW'96.*

## Referencias

# APÉNDICE

# EJEMPLOS DE FORMALIZACIÓN DE ONTOLOGÍAS

En este apéndice se expresan formalmente algunas de las ontologías usadas en capítulos anteriores. Para ello, se hará uso del formalismo de tripletes definido en el capítulo 2. En primer lugar se especificarán las ontologías usadas en los capítulos 3 y 4 para ilustrar el entorno de integración. Posteriormente se especificará la ontología final obtenida como resultado del segundo experimento en el campo bioinformático.

# A. ONTOLOGIAS PARA ILUSTRAR EL PROCESO DE INTEGRACIÓN

## ONTOLOGÍA A

### Ontología

C = {library, publication, person, newspaper, magazine, book, staff, user, novel, essay, article}

P= {(publication, library, PART-OF); (person, library, PART-OF); PART-OF(article, newspaper, PART-OF); (article, magazine, PART-OF); (newspaper, publication, IS-A); (magazine, publication, IS-A); (book, publication, IS-A); (staff, person, IS-A); (user, person, IS-A); (novel, book, IS-A); (essay, book, IS-A)}.

## Conceptos

Library

A= {}

Re= { publication, person}

Publication

A= {title, editor, pages}

Re= { library, newspaper, magazine, book}

Person

A= {name, id, address, role}

Re= {library, staff, user}

Newspaper

A= { date, title, editor, pages}

Re= { publication , article}

Magazine

A={area, periodicity, title, editor, pages}

Re={publication, article}

Book

A={ author, ISBN, style, title, editor, pages}

Re={publication, novel, essay}

Staff

A={ salary, position, name, id, role, address}

Re={ person}

User

A={ number, name, id, role, address}

Re={ person}

Novel

A={ chapters, author, ISBN, style, title, editor, pages}

Re={ book }

Essay

A={ topic, author, ISBN, style, title, editor, pages}

Re={ book }

Article

A= { length, author, keywords}

Re={newspaper, magazine}

# ONTOLOGÍA B

## Ontología

C = {library, publication, people, newspaper, magazine, staff, user, apportion}

P= {(publication, library, PART-OF); (people, library, PART-OF); PART-OF(apportion, newspaper, PART-OF); (apportion, magazine, PART-OF); (newspaper, publication, IS-A); (magazine, publication, IS-A); (staff, person, IS-A); (user, people, IS-A)}.

## Conceptos

Library

A= {}

Re= { publication, people}

Publication

A= {title, editor, pages}

Re= { library, newspaper, magazine}

People

A= {name, id, address, role}

Re= { library, staff, user}

Newspaper

A= { date, title, editor, pages}

Re= { publication , apportion}

Magazine

A={area, periodicity, title, editor, pages}

Re={publication, apportion}

Staff

A={ salary, position, name, id, role, address}

Re={ people}

User

A={ number, name, id, role, address}

Re={ people}

Apportion

A= { length, author, keywords}

Re={newspaper, magazine}

# ONTOLOGÍA C

## Ontología

C = {library, publication, person, newspaper, magazine, staff, user, apportion}

P= {(publication, library, PART-OF); (person, library, PART-OF); PART-OF(apportion, newspaper, PART-OF); (apportion, magazine, PART-OF); (newspaper, publication, IS-A); (magazine, publication, IS-A); (staff, person, IS-A); (user, person, IS-A)}.

## Conceptos

Library

A={}

Re={ publication, person}

Publication

A= {title, editor, pages}

Re= { library, newspaper, magazine, book}

Person

A= {name, id, address, role}

Re= {library, staff, user}

Newspaper

A= { date, title, editor, pages}

Re= { publication , article}

Magazine

A={area, periodicity, title, editor, pages}

Re={publication, article}

Book

A={ author, ISBN, style, title, editor, pages}

Re={publication}

Staff

A={ salary, position, name, id, role, address}

Re={ person}

User

A={ number, name, id, role, address}

Re={ person}

Article

A= { subject}

Re= {newspaper, magazine}

# ONTOLOGÍA D

## Ontología

C = {library, publication, person, newspaper, magazine, article, novel, book, poetry, staff, security, admin}

P= {(publication, library, PART-OF); (person, library, PART-OF); PART-OF(article, newspaper, PART-OF); (article, magazine, PART-OF); (newspaper, publication, IS-A); (magazine, publication, IS-A); (novel, magazine, IS-A); (poetry, book, IS-A); (staff, person, IS-A); (security, staff, IS-A); (admin, staff, IS-A)}.

## Conceptos

Library

A={}

Re={ publication, person}

Publication

A= {title, editor, pages}

Re= {library, newspaper, magazine, book}

Person

A= {name, id, address, role}

Re= {library, staff, user}

Newspaper

A= { date, title, editor, pages}

Re= { publication , article}

Magazine

A={area, periodicity, title, editor, pages}

Re={publication, article, novel}

Novel

A={ chapter, comments, area, periodicity, title, editor, pages}

Re={magazine}

Staff

A={ salary, position, name, id, role, address}

Re={ person}

Book

A={ author, ISBN, style, title, editor, pages}

Re={ publication, poetry}

Article

A= { length, author, keywords}

Re={ newspaper, magazine}

Poetry

A={ topic, rhyme, author, ISBN, style, title, editor, pages}

Re={ book}

Security

A={ schedule, salary, position, name, id, role, address}

Re={ staff}

Admin

A={ office, salary, position, name, id, role, address}

Re={ staff}

# B. ONTOLOGÍA FINAL SOBRE PROTEÍNAS

## Ontología

C = {protein, structure, primary, secondary, tertiary, quaternary, alpha-helix, lateral chain, central structure, beta-sheet, anti-parallel, parallel, composition, amino acid, radical, carbon, coo-, nh3+, alanine, valine, proline, glycine}.

P= {(structure, protein, PART-OF); (composition, protein, PART-OF); (amino acid, composition, PART-OF); (radical, composition, PART-OF); (carbon, composition, PART-OF); (coo-, composition, PART-OF); (nh3+, composition, PART-OF); (primary, structure, IS-A) ; (secondary, structure, IS-A) ; (tertiary, structure, IS-A) ; (quaternary, structure, IS-A) ; (alpha-helix, secondary, IS-A) ; (beta-sheet, secondary, IS-A) ; (lateral chain, alpha-helix, IS-A) ; (central structure, alpha-helix, IS-A) ; (anti-parallel, beta-sheet, IS-A) ; (parallel, beta-sheet, IS-A) ; (alanine, amino acid, IS-A) ; (valine, amino acid, IS-A) ; (proline, amino acid, IS-A) ; (glycine, amino acid, IS-A)}.

## Conceptos

Protein

A: { name, function }

Re: { structure, composition}

Structure

A:{ structure, type}

Re: {protein, primary, secondary, tertiary, quaternary}

Primary

A: { order, structure type}

Re: { structure}

Secondary

A: { Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type}

Re: { structure, alpha-helix, beta-sheet}

Tertiary

A: { spatial form, structure type}

Re:{ structure}

Quaternary

A: { number chains, structure type}

Re:{ structure }

Alpha-helix

A: { Ccap, Ncap, geometry, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type }

Re: {secondary, alpha-helix, beta-sheet}

Lateral chain

A:{ Ccap, Ncap, geometry, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type}

Re:{ alpha-helix}

Central structure

A:{ Ccap, Ncap, geometry, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type }

Re:{alpha-helix}

Beta-sheet

A:{ Stability, Orientation, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type }

Re:{secondary, anti-parallel, parallel}

Anti-parallel

A:{ Stability, orientation, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type }

Re:{ beta-sheet }

Parallel

A:{ Stability, orientation, Phi, psi, residues translation , length, N-terminus charge, C-terminus charge, Structure_Type }

Re: {beta-sheet}

Composition

A:{ distribution }

Re:{ protein, aminoacid }

Amino acid

A:{ Property, function, name, composition}

Re:{ composition, alanine, glycine, valine, proline, carbon, radical, coo-, nh3+}

Radical

A: { name, composition}

Re:{ amino acid}

Carbon

A :{ atomic mass }

Re :{ amino acid}

coo-

A:{ molecular mass, purity}

Re: { amino acid}

nh3+

A :{molecular mass, purity}

Re :{ amino acid }

Alanine

A:{ GCx, Property, function, name, composition }

Re:{ amino acid}

Valine

A:{ GUx, Property, function, name, composition}

Re:{ amino acid}

Proline

A:{ CCx, Property, function, name, composition}

Re:{ amino acid}

Glycine

A:{ GGx, Property, function, name, composition}

Re:{ amino acid}