



**VNiVERSiDAD
DSALAMANCA**

UNIVERSIDAD DE SALAMANCA

Departamento de Informática y Automática

**ANALÍTICA VISUAL APLICADA
A LA
INGENIERÍA DE ONTOLOGÍAS**

TESIS DOCTORAL

D. JUAN FRANCISCO GARCÍA NAVARRO

Directores:

DR. D. FRANCISCO JOSÉ GARCÍA PEÑALVO
DR. D. ROBERTO THERÓN SÁNCHEZ

Febrero 2012



**VNiVERSiDAD
D SALAMANCA**

UNIVERSIDAD DE SALAMANCA
Departamento de Informática y Automática

**ANALÍTICA VISUAL APLICADA
A LA
INGENIERÍA DE ONTOLOGÍAS**

TESIS DOCTORAL PRESENTADA POR:
D. JUAN FRANCISCO GARCÍA NAVARRO

Dirigida por:
DR. JOSÉ FRANCISCO GARCÍA PEÑALVO
DR. ROBERTO THERÓN SÁNCHEZ

El doctorando

Salamanca, febrero de 2012

Dr. D. Francisco José García Peñalvo, *Profesor Titular del Departamento de Informática y Automática de la Universidad de Salamanca*

Dr. D. Roberto Therón Sánchez, *Profesor Titular del Departamento de Informática y Automática de la Universidad de Salamanca*

HACEN CONSTAR: *Que D. Juan Francisco García Navarro, ha realizado bajo nuestra dirección el trabajo de investigación y la memoria de la tesis doctoral que lleva por título Analítica Visual aplicada a la Ingeniería de Ontologías, con el fin de obtener el grado de Doctor por la Universidad de Salamanca.*

Y para que surta los efectos oportunos firmamos en Salamanca, a diez de febrero de dos mil doce.

Agradecimientos

Quiero comenzar agradeciendo a mi familia, que siempre me ha apoyado en todas las decisiones que he tomado, y que me han brindado todo su cariño el cual es correspondido de igual forma.

También quiero agradecer de forma muy especial al Consejo Nacional de Ciencia y Tecnología (CONACYT) y al Banco Santander, por el apoyo económico que me brindaron durante el desarrollo de mis estudios de doctorado, sin el cual no hubiera sido posible alcanzar este objetivo.

Otro agradecimiento a mis tutores, el Dr. Francisco José García Peñalvo y el Dr. Roberto Therón Sánchez, que me han ayudado a lo largo de todo este tiempo.

Finalmente, también quiero agradecer a todos mis amigos en Salamanca, que al estar lejos de mi país, ellos se han convertido en una segunda familia para mí, y con quienes he compartido numerosos momentos inolvidables.

Resumen

La representación del conocimiento se ha definido como un conjunto de reglas ontológicas provistas con la capacidad para realizar inferencias. Uno de los esquemas de representación de conocimiento más ampliamente difundido lo constituyen las ontologías. En términos muy generales, las ontologías son ampliamente utilizadas en la Ingeniería del Conocimiento, la Inteligencia Artificial y las Ciencias de la Computación.

Una problemática en la representación del conocimiento desde un punto de vista semántico, es la complejidad de los procesos de creación, gestión, análisis y reusabilidad de los modelos ontológicos que representan estas estructuras de conocimiento. El soporte automático o semiautomático a los procesos relacionados con la representación de conocimiento, suele enfrentar diversos problemas durante el proceso de modelado de datos ontológicos, que acaban por comprometer la comprensión y el desarrollo de los modelos. Esto puede tener como resultado, procesos de desarrollo y gestión complejos, ineficientes y, así mismo, una escasa reutilización de los propios modelos ontológicos. Muchas de las herramientas que soportan estos procesos, no son lo suficientemente robustas para realizar eficientemente todas las tareas involucradas en los flujos de trabajo.

Esta tesis toma como dominio de investigación esta problemática, y plantea como objetivo principal la definición de diversos procesos para la gestión del ciclo de vida de las ontologías, sustentados en un enfoque de Analítica Visual, un área de investigación basada en el uso de representaciones visuales, que se enfoca en el uso del proceso cognitivo humano para el análisis de información.

El enfoque que sigue la Analítica Visual destaca al usuario como punto central en el proceso de análisis, y todos los mecanismos que intervienen en este enfoque se definen tomando en consideración, como aspectos principales, el usuario y la interacción de este con las herramientas de cómputo. Estos mecanismos facilitan el razonamiento analítico, al maximizar la capacidad humana para percibir, entender y razonar acerca de conjuntos de datos complejos

y dinámicos.

Los procesos involucrados en el ciclo de vida de las ontologías (creación, gestión, análisis y reutilización), describen flujos de trabajo formados por diversas actividades. Estas actividades se han definido tomando en consideración las principales metodologías para el desarrollo de modelos ontológicos.

Por otro lado, también es necesario que estas actividades estén soportadas por mecanismos y herramientas que permitan su desarrollo de forma eficiente. Estos mecanismos son principalmente, técnicas de visualización robustas y provistas de una interacción que permita al usuario a través de su capacidad, desarrollar abstracción, concepción, entendimiento, representación y aprendizaje de conocimiento.

A lo largo de esta tesis se detallan los flujos de trabajo y los mecanismos propuestos para el desarrollo de los procesos para el modelado de datos basado en ontologías. Teniendo en cuenta estos mecanismos, se ha implementado una herramienta denominada OWL-VisMod, y se han evaluado desde dos perspectivas. La primera de ellas se basa en la usabilidad de las tareas definidas, mientras que la segunda se enfoca en la funcionalidad de la herramienta y todas las técnicas de visualización e interacción.

La evaluación de usabilidad consiste en el desarrollo del proceso de creación de un modelo ontológico, de acuerdo a las tareas propuestas. Este desarrollo se ha realizado por un grupo de usuarios, quienes emitieron una valoración sobre la usabilidad de la herramienta. Por otro lado, la evaluación de funcionalidad, se ha realizado primero, a través de un proceso de análisis sobre un modelo ontológico existente, de acuerdo al flujo de trabajo definido para este proceso, y segundo, a través de un proceso de creación de un nuevo modelo ontológico de datos, para un sistema real en un dominio específico, en este caso, en el sector del turismo.

Los resultados de las evaluaciones han indicado, por un lado, que los procesos, los flujos de trabajo y las tareas definidas, son coherentes con los objetivos definidos. Este resultado se debe en gran parte a que estas tareas se han adaptado a partir de metodologías previamente definidas. Por otro lado, la evaluación de funcionalidad en el caso de estudio, ha permitido detectar algunos aspectos mejorables, así como también ha servido para la validación general de la herramienta y las diversas técnicas de visualización e interacción implementadas en ella.

Palabras Clave

Representación del Conocimiento, Ontologías, Analítica Visual, Procesos

Abstract

Knowledge Representation has been defined as a set of ontological rules provided with the capacity to perform inference. It basically consists of a set of symbols that represent a specific domain, enriched with a set of functions to perform reasoning. This reasoning aims to perform inference, most of times based on the use of any kind of logics. Ontologies represent one of the most widely used schema representations. They are mainly used in the Knowledge Engineering, Artificial Intelligence and the Computer Science fields.

The main problem with the Knowledge Representation from a semantic point of view, is the complexity of the processes of creation, management, maintenance and reusability of the ontological models. The automatic support to the processes that manage the knowledge, usually have to face diverse problems during the modelling process. These problems affect the comprehension and the development of these ontological models, causing complex and unefficient processes and low levels of reusability of these ontologies.

The current tools that support these processes, have diverse problems, due to they are not robust enough to efficiently develop all the tasks involved in the workflows.

This thesis has taken this problematic as its starting point. It aims to develop the processes for developing and managing the life cycle of ontologies based on a Visual Analytics approach, where the user becomes to be the central point in the analysis process. Furthermore, all the mechanisms involved in this approach, have been defined taking into account, the user and his interaction with the computers. These mechanisms, make easier the analytical reasoning, because they maximise the human capacity to perceive and understand about complex and dynamic data sets.

The processes involved in the ontologies'life cycle, describe the whole sequence of activities in the workflows, to create, manage and reuse ontologies. These activities and workflows have been defined taking into account the main methodologies for developing ontological models. Moreover, each of these ac-

tivities and workflows, is supported by Visual Analytics and interaction techniques.

The user interaction represents a crucial aspect for developing ontology-based data models. The user, supported by computer mechanisms, defines the representation of a certain domain using an ontological model. These mechanisms, are completely defined according to the human capabilities to perform abstraction, conception, understanding, learning and knowledge representation. This user-centered approach, represents the main approach taken of the Visual Analytics field.

Through this thesis, all the workflows for developing ontology-based data models are completely described and detailed. Furthermore, all the mechanisms implemented in a tool called OWL-VisMod, that support the development of these processes are also described. These mechanisms have been evaluated based on two approaches: the first one is the usability of the tool, and the second one is the functionality of the tool and all the visualisation and interaction techniques.

The usability evaluation consists of the development of the creation process of an ontological model, according to the proposed tasks. This development has been performed by a group of users that evaluate the usability of the tool. On the other hand, the functionality evaluation has been performed, first, through an analysis process of an existing ontology, and second, through the creation of a new data ontological model, according to the workflows defined to these processes.

This creation process has been developed over a real scenario, of a recommend system for tourism activities. This evaluation has let to perform diverse tests of the tool, about its capacity to cover all the functionality aspects needed for a real-scenario system.

The results of these evaluations, have indicated firstly, that the processes, workflows and the defined tasks, are coherent to the defined objectives. Secondly, the functionality evaluation in the case-study, has let to detect diverse aspects that were improved or fixed, as well as the validation of the tool and the diverse visualisations and interaction techniques.

Keywords

Knowledge Representation, Ontologies, Visual Analytics, Processes

Índice general

1. Introducción	1
1.1. Presentación	1
1.2. Problemática	8
1.3. Objetivos	13
1.4. Metodología de investigación	14
1.5. Líneas de investigación de la tesis	15
1.6. Contenido	16
2. Ontologías	19
2.1. ¿Qué es una Ontología?	19
2.2. Aplicaciones de las Ontologías	23
2.3. Ingeniería de Ontologías	25
2.3.1. Metodologías para la creación de Ontologías	25
2.3.2. Representación de ontologías	31
2.3.3. Métricas ontológicas	34
2.3.4. Herramientas para la gestión de ontologías	38
2.4. Conclusiones	48
3. Visualización de Información y Analítica Visual	51
3.1. Visualización de Información	52
3.1.1. Proceso de visualización de datos	53

3.1.2.	Modelado de datos	55
3.2.	Analítica Visual	57
3.2.1.	Exploración visual de los datos	59
3.2.2.	Técnicas de Interacción Visual	60
3.2.3.	Consultas Visuales	64
3.3.	Conclusiones	65
4.	Estado del arte	67
4.1.	Creación de modelos ontológicos	68
4.2.	Análisis de ontologías	72
4.3.	Procesos de gestión y reutilización de ontologías	78
4.4.	Problemática visual de las herramientas	81
4.5.	Conclusiones	96
5.	Modelo propuesto para la gestión de ontologías	99
5.1.	Flujos de trabajo para el desarrollo de la Ingeniería de Ontologías	100
5.1.1.	Creación de ontologías	101
5.1.2.	Gestión de ontologías	103
5.1.3.	Análisis de un modelo ontológico	104
5.1.4.	Métricas de evaluación de ontologías	106
5.2.	Soluciones propuestas	109
5.2.1.	Saturación de elementos visuales	109
5.2.2.	Poca expresividad visual	110
5.2.3.	Pérdida del contexto de navegación	111
5.2.4.	Falta de mecanismos de interacción	112
5.3.	Soporte a las fases de implementación y análisis	113
5.3.1.	Transformación de los datos	114
5.3.2.	Soporte a la fase de implementación	115

5.3.3. Soporte a la fase de análisis	130
5.4. Conclusiones	139
6. Evaluación de usabilidad	143
6.1. Pruebas de Usabilidad	144
6.2. Análisis de un modelo ontológico	146
6.3. Creación de un nuevo modelo ontológico	149
6.4. Evaluación de los procesos definidos	151
6.4.1. Taxonomía de la ontología	151
6.4.2. Zoom semántico	151
6.4.3. Acoplamiento de las clases	152
6.4.4. Satisfacción general de OWL-VisMod	152
6.5. Resultados de la evaluación	152
6.6. Conclusiones	154
7. Caso de Estudio	157
7.1. Proyectos Turimov y Ulises	157
7.1.1. Dominio del problema	158
7.1.2. Taxonomía de las ontologías	165
7.1.3. Propiedades intrínsecas	168
7.1.4. Acoplamiento de los modelos ontológicos	170
7.1.5. Instancias de las clases	171
7.2. Conclusiones	174
8. Resultados y conclusiones	177
8.1. Resultados	178
8.2. Conclusiones	180
8.3. Publicaciones relacionadas con la tesis	183
8.4. Líneas de trabajo futuro	187

A. Manual de usuario	189
A.1. Crear una nueva ontología	190
A.2. Abrir una ontología existente	191
A.3. Guardar una ontología	192
A.4. Configurar una conexión a base de datos	192
A.5. Agregar submodelos al modelo actual	193
A.6. Protocolo FTP (<i>File Transfer Protocol</i>) para transferencia de archivos	194
A.7. Menú visual	195
A.8. Desplegar el código XML	195
A.9. Desplegar la ayuda	195
A.10. Uso del difuminado del fondo	195
A.11. Búsqueda de clases	196
A.12. Desplegar estadísticas	197
A.13. Reiniciar las visualizaciones	197
A.14. Cambiar de visualización	197
A.15. Modificar el esquema de la ontología	198
A.16. Crear una clase como unión o intersección	199
A.17. Navegación en los niveles del <i>treemap</i>	200
A.18. Crear clases enumeradas	201
A.19. Eliminar clases, propiedades o instancias	201
A.20. Opciones de interacción con la <i>table lens</i>	202
A.21. Opciones de interacción con la vista de acoplamiento global	203
A.22. Interacción con las listas circulares	206
A.23. Menús visuales para clases y propiedades	207
B. Glosario de términos y acrónimos	209
C. Carta de colaboración empresa Interra	211

Índice de Tablas

2.1. Resumen de las principales metodologías propuestas, donde se muestran los aspectos más importantes: a) si tienen algún tipo de soporte por parte de alguna herramienta; b) si fomentan el principio de reutilización; c) si abarcan el ciclo de vida completo de las ontologías y d) si están enfocadas en la evaluación de las ontologías.	30
2.2. Clasificación más general de los lenguajes para la representación de ontologías. Se definen tres grandes áreas, por un lado los lenguajes tradicionales, por otro lado los estándares web y las recomendaciones y, finalmente, los lenguajes basados en la Web.	34
2.3. Resumen comparativo de las métricas ontológicas descritas. Se indica si las métricas se basan en la estructura o si son semánticas, si permiten clasificar a las ontologías, si calculan valores de cohesión y acoplamiento.	38
2.4. Resume las características más importantes de las principales herramientas para el desarrollo de la Ingeniería de Ontologías. .	47
4.1. Resumen de la problemática de las herramientas que implementan técnicas de visualización, de acuerdo a la clasificación establecida.	95
7.1. Un resumen del total de instancias de recursos turísticos.	173

Índice de Figuras

1.1. Los procesos básicos en Ingeniería de Ontologías son: creación, gestión, análisis y reutilización. Cada uno de los procesos define un flujo de trabajo formado por varias actividades o tareas. . .	4
1.2. Principales tareas definidas en Methontology, para el proceso de creación de ontologías.	5
1.3. El proceso de creación de ontologías está definido por cinco tareas principales, divididas en dos grandes fases: una fase de abstracción del dominio, y otra fase para la implementación del modelo ontológico, que incluye la definición de la taxonomía de conceptos, la definición de las propiedades intrínsecas a las clases, la definición de las propiedades objeto que representan las relaciones, y la definición de las instancias de las clases, que forman la base de conocimiento.	6
1.4. La fase de implementación se vuelve extremadamente compleja, debido al uso de herramientas poco robustas. La saturación visual de elementos complica en demasía el entendimiento del dominio que se modela.	11
2.1. El esquema de algunos de los principales conceptos de la ontología <i>wine.owl</i> , sus relaciones y las instancias de la clase <i>Wine-Grape</i> , que define los tipos de uvas existentes.	22
2.2. Las seis etapas definidas en la metodología <i>Knowledge Engineering Methodology</i> [124].	26
2.3. (a) Muestra la representación de las métricas con Protégé, mientras que la figura (b) muestra las métricas con la herramienta <i>Ontology Metrics</i> . Ambas herramientas se limitan a desplegar valores numéricos globales a todo el modelo ontológico, por ejemplo el número total de clases, de propiedades, de instancias, etc.	37

2.4.	Una vista de la herramienta Semantic Works. Utiliza representaciones basadas en mapas conceptuales. Define sus propia simbología para la representación de las propiedades, los conceptos, la jerarquía, la unión de clases, etc.	39
2.5.	Una vista de la herramienta Top Braid Composer. Utiliza el mismo diseño de interfaz gráfica de usuario que Protégé. Se muestra del lado izquierdo la jerarquía de clases, al centro las relaciones entre las clases, y del lado derecho se muestran las propiedades objeto y de datos.	40
2.6.	La herramienta Neon Toolkit muestra la jerarquía de los conceptos (en un recuadro en color rojo del lado izquierdo), mientras que del lado derecho despliega las visualizaciones de la ontología con las propiedades.	43
2.7.	Una vista de la herramienta <i>TgVizTab</i> de Protégé. La visualización se enfoca en la representación de los conceptos y las relaciones entre estos.	45
3.1.	Diagrama esquemático del proceso de visualización definido en [274]. Este proceso define cuatro etapas básicas: el almacenamiento de los datos, el preprocesamiento y transformación de los datos, el motor gráfico y finalmente el análisis humano a través de un proceso visual y cognitivo.	54
3.2.	Un modelo de entidad-relación para una base de datos en un contexto académico. Se muestran las entidades persistentes del sistema, y las relaciones existentes entre estas.	56
3.3.	El proceso de exploración visual de los datos bajo el enfoque de la Analítica Visual definido en [165]. Destacan principalmente las fases de transformación de los datos en estructuras más complejas, el modelado de las estructuras, la representación visual y finalmente la fase de análisis para la obtención de conocimiento.	59
3.4.	El <i>Foco+Contexto</i> puede lograrse aplicando una técnica de difuminado. En este caso, algunas clases se difuminan para restarles importancia al quedar en un segundo plano (Class2 y Class3), otras se dejan en su mismo contexto (Class1 y Class6), mientras que otras incluso se resaltan para asignarles un grado de importancia mucho mayor (Class4 y Class5).	61

3.5.	El <i>zoom semántico</i> cambia el tipo y la información desplegada, generalmente se utiliza para mostrar los detalles de un cierto elemento que ha sido seleccionado. También, puede o no, incluir un zoom físico para aumentar el tamaño del elemento seleccionado.	62
3.6.	La distribución física de los elementos sobre el espacio dimensional es una de las variables de la expresividad visual. La Figura (a) distribuye los elementos de forma circular, y los ordena en el mismo sentido en el que giran las manecillas del reloj. La Figura (b) distribuye los elementos de forma lineal, y los ordena en sentido de izquierda a derecha, mientras que la Figura (c) distribuye los elementos de forma completamente aleatoria.	63
4.1.	El editor de ontologías HOZO utiliza listas para desplegar la jerarquía (mostrada en el lado izquierdo de la visualización). En el lado derecho se muestran las propiedades de la clase, la jerarquía de superclases, así como las instancias de la clase seleccionada.	69
4.2.	La herramienta OwlEditor es una aplicación de escritorio que permite editar ontologías OWL en formato textual, sin implementar visualizaciones ni mecanismos de interacción robustos.	71
4.3.	OntoStudio representa la taxonomía de conceptos mediante el uso de listas jerárquicas, como se ilustra en el recuadro definido en color azul. Por otro lado, las propiedades intrínsecas se representan mediante el uso de nodos en un grafo, como se resalta en el recuadro definido en color amarillo, mientras que las relaciones de acoplamiento se resaltan mediante el uso de nodos en color rojo.	77
4.4.	El proceso de reutilización de ontologías comienza con la fase de abstracción del dominio del problema. La segunda fase corresponde a una búsqueda de los modelos ontológicos que puedan reutilizarse. La tercera fase corresponde al análisis de los modelos ontológicos reutilizados, con el objetivo de hacer un diagnóstico sobre los cambios exactos que se requieren. Después, la última fase corresponde a la implementación del nuevo modelo ontológico.	79
4.5.	La herramienta NeOn Toolkit, en donde la saturación de los elementos no permite el análisis adecuado del modelo ontológico.	84

4.6.	Una vista de la herramienta Jambalaya, un <i>plugin</i> de Protégé, en donde los elementos visuales no tienen una distribución física sobre el espacio dimensional.	85
4.7.	AutoFocus utiliza una visualización basada en agrupaciones, para la representación de los resultados de búsquedas semánticas sobre documentos, previamente indexados y clasificados. .	86
4.8.	(a) Un diagrama de estados con el flujo de navegación completo. (b) Un diagrama de estados sin el flujo de navegación (pérdida del contexto). (c) La selección de un elemento con el contexto completo. (d) La selección del mismo elemento que en (c) pero sin el contexto.	87
4.9.	Tres diferentes herramientas que utilizan listas jerárquicas: (a)Protégé, (b) OntoStudio y (c) SWOOP. Esta técnica falla principalmente porque no despliega la jerarquía completa de superclases, para una clase que se haya seleccionado.	89
4.10.	Los árboles de jerarquías representan una solución exitosa a la problemática de la representación de grandes jerarquías.	90
4.11.	Una técnica visual para la representación de grandes jerarquías es el <i>treemap</i> , en este caso se ilustra una vista de la aplicación <i>PhotoMesa</i>	91
5.1.	Las principales tareas para desarrollar la fase de abstracción del dominio del problema, durante el proceso de creación de ontologías. Estas tareas se desarrollan completamente por el diseñador del modelo ontológico.	101
5.2.	Las principales tareas para la fase de implementación de ontologías, dentro de los procesos de creación, gestión y reutilización. Comienza con la definición de la taxonomía de clases, la definición de las propiedades intrínsecas, las relaciones entre clases, y finalmente, la implementación de instancias que definen la base de conocimiento.	102
5.3.	El proceso de gestión de ontologías comienza con una fase de análisis del modelo ontológico que se quiere actualizar, con el objetivo de hacer un diagnóstico sobre los cambios exactos que se requieren.	104

5.4. El proceso de análisis de ontologías define cuatro actividades principales: el análisis de la taxonomía de los conceptos, la búsqueda específica de clases en el modelo, la búsqueda de relaciones y el análisis de las instancias.	105
5.5. Esquema que muestra la definición del acoplamiento en las ontologías OWL. La primera métrica <i>CBE-out</i> , indica que la clase se localiza en el dominio de la relación, mientras que la segunda métrica <i>CBE-in</i> , indica que la clase se localiza en el rango de la relación. La métrica <i>SC</i> , indica que la clase está acoplada consigo misma, tanto en el dominio como en el rango. Finalmente, la métrica <i>CBE-io</i> indica que hay una relación de propiedades inversas entre dos clases.	107
5.6. (a) Muestra un esquema en el que toda la información se muestra al mismo tiempo, lo que complica en demasía el proceso de abstracción. (b) Muestra un enfoque diferente, en el que el usuario sigue un flujo de trabajo definido para ir conociendo parte del sistema modelado.	110
5.7. La arquitectura de OWL-VisMod representa una capa superior sobre Jena, que se encarga de administrar los modelos ontológicos directamente.	114
5.8. El flujo de trabajo que define las principales actividades que se desarrollan en el proceso de creación de ontologías. Cada actividad tiene asociadas algunas herramientas, técnicas o visualizaciones, que soportan su desarrollo.	116
5.9. Una vista general de una jerarquía de clases, con una visualización <i>treemap</i> en OWL-VisMod. Los cuatro niveles superiores se han colocado por encima del <i>treemap</i> , el cual ha sido calculado a partir del quinto nivel. Esta funcionalidad permite recorrer en profundidad todos los niveles internos de la taxonomía. . . .	117
5.10. Una secuencia de la navegación sobre los niveles internos del <i>treemap</i> . (a) muestra el <i>treemap</i> a partir del tercer nivel de la jerarquía, los dos niveles superiores se muestran en la parte de arriba. (b) muestra el <i>treemap</i> a partir del sexto nivel de la jerarquía, los cinco niveles superiores se muestran en la parte de arriba.	118

5.11. Una vista general de la jerarquía de clases mediante el uso de una visualización basada en árboles jerárquicos. La jerarquía de la clase seleccionada se ha resaltado con recuadros en color rojo. El <i>treemap</i> se mantiene en un segundo plano, como efecto de la interacción que hace la transición entre las dos vistas.	120
5.12. Una vista de la implementación del <i>zoom semántico</i> en una visualización en donde el contexto completo permanece.	122
5.13. Una vista general del flujo de navegación, que comienza desde una vista general hacia lo particular. Una vez presentados los detalles, el flujo permite regresar hacia la vista global, y repetir esta tarea de navegación indefinidamente.	123
5.14. (a) muestra el <i>zoom</i> semántico sobre una clase llamada <i>Restaurante</i> , con nueve propiedades objeto (coloreadas en tono rojo), y seis propiedades de datos (coloreadas en tono verde). La Figura (b) muestra el <i>zoom</i> semántico sobre la misma clase <i>Restaurante</i> , con una selección de la propiedad <i>gastronomía</i> que tiene un total de nueve valores de instancias. La consulta visual despliega todas aquellas instancias de restaurantes, que tienen definida en su <i>gastronomía</i> al valor “casera”.	124
5.15. Esta técnica de <i>zoom</i> semántico se basa en la distribución radial de sus elementos para representar las instancias de la clase seleccionada. Las clases disjuntas y las clases equivalentes a la clase seleccionada se muestran en una representación de agrupaciones de clases.	126
5.16. La visualización del acoplamiento de una clase de acuerdo a las métricas CBE. El acoplamiento de la clase seleccionada se representa con las propiedades alrededor de esta. En cada propiedad se colocan las clases que representan el acoplamiento ya sea CBE-out o CBE-in.	127
5.17. El <i>zoom</i> semántico sobre el acoplamiento de la clase <i>Persona</i> . En esta imagen se ha enfocado la propiedad <i>actividadPreferida</i> , que tiene a la clase <i>Persona</i> en el dominio y a la clase <i>Actividad</i> en el rango.	129
5.18. El proceso de análisis de un modelo ontológico está definido por cuatro actividades principales: el análisis de la taxonomía de las clases, el análisis de las propiedades de datos, el análisis de las instancias y el análisis del acoplamiento de las clases.	130

5.19. El <i>zoom</i> semántico usado para realizar consultas visuales (<i>visual querying</i>). La Figura (a) ilustra la consulta de todos los valores para cada una de las propiedades para la instancia seleccionada, mientras que la Figura (b) ilustra la consulta de todas las instancias que tienen un específico valor para una de las propiedades.	132
5.20. (a) La vista con el acoplamiento global de todas las clases en la ontología <i>turisem.owl</i> . (b) El acoplamiento de la clase llamada <i>Lugar</i> , la clase con más relaciones en este modelo ontológico. . .	134
5.21. La representación de las propiedades se basa en el uso de agrupaciones de clases en el dominio de la propiedad y agrupaciones de clases en el rango de la propiedad, en las que puede haber una o más clases. En este caso, la propiedad se llama <i>sexo</i> , es funcional y tiene a la clase <i>Persona</i> en el dominio y a la clase <i>Sexo</i> en el rango. Conceptualmente representa el valor del género sexual de una persona (masculino o femenino).	135
5.22. La vista del acoplamiento de la clase <i>Lugar</i> , desde la perspectiva del <i>zoom</i> semántico. Esta clase tiene trece propiedades objeto, y cuatro propiedades de datos.	137
5.23. Una vista de la tabla que despliega los valores numéricos de las diversas métricas calculadas por OWL-VisMod. La visualización <i>table lens</i> se basa en la distorsión de la fila y la columna seleccionada, a la cual se provee con el foco principal, sin perder el contexto global. Los valores nominales se despliegan, lo que representa una implementación de la técnica de las consultas visuales (<i>visual querying</i>) descrita en la sección 3.2.3. Las columnas se asocian con colores diferentes para mejorar la capacidad de análisis del usuario.	138
6.1. Se muestra la jerarquía de una clase en el octavo nivel de la ontología SAO. Se ilustra toda la jerarquía de las superclases en los niveles previos.	147
6.2. Se muestra el acoplamiento global de la ontología SAO, con treinta y cinco clases con relaciones de acoplamiento, ochenta y una propiedades, de las cuales, treinta y seis son propiedades objeto y cuarenta y cinco propiedades de datos.	148

6.3.	Árbol con la vista de la representación jerárquica de las clases en uno de los modelos ontológicos creado por un usuario para la evaluación de usabilidad.	149
6.4.	Vista detallada del <i>zoom</i> semántico de uno de los modelos creado por los usuarios. La visualización incluye el uso de la técnica de consultas visuales, en la que se muestra una consulta de una instancia de la clase <i>Profesor</i> y los valores de esta instancia para cada una de las propiedades.	150
6.5.	La gráfica de los resultados de las preguntas cerradas, se resaltan en color rojo aquellas preguntas en las que ha habido que poner mayor atención y en amarillo otras preguntas que también se consideraron como importantes.	153
7.1.	Los submodelos ontológicos de persona (<i>turisem_personas.owl</i>) y dispositivos (<i>turisem_dispositivos.owl</i>). Las principales clases son <i>Persona</i> , <i>Dispositivo</i> , <i>PlataformaSoftware</i> y <i>PlataformaHardware</i>	159
7.2.	La jerarquía completa del modelo ontológico <i>wine.owl</i> ; se ha extendido la rama con la jerarquía de las clases relacionadas con gastronomía definidas en esta ontología.	161
7.3.	(a) muestra las instancias de la clase <i>WineGrape</i> , que corresponden a los tipos de uvas. (b) muestra una vista del <i>zoom semántico</i> que representa el acoplamiento de la clase <i>Wine</i> . . .	163
7.4.	Una vista del acoplamiento global de la ontología <i>wine.owl</i> , en donde se ha resaltado la clase <i>Wine</i> y se muestran sus relaciones. En este caso hay tres relaciones definidas por las propiedades objeto: <i>hasColor</i> , <i>hasWineDescriptor</i> y <i>madeFromGrape</i>	164
7.5.	Una vista de la representación de la propiedad <i>madeFromGrape</i> , con la clase <i>Wine</i> en el dominio y la clase <i>WineGrape</i> en el rango. 165	
7.6.	El flujo de trabajo de las tareas para el proceso de creación de los modelos ontológicos.	166
7.7.	La jerarquía completa con los tres submodelos, <i>turisem.owl</i> , <i>turisem_dispositivos.owl</i> y <i>turisem_personas.owl</i> . El modelo ontológico completo contiene un total de ciento cuarenta y tres clases. 167	

7.8.	(a) muestra el <i>zoom</i> semántico sobre la clase <i>Dispositivo</i> , con cinco propiedades de datos. La Figura (b) muestra el <i>zoom</i> semántico sobre la clase <i>Lugar</i> , con siete propiedades objeto y cuatro propiedades de datos.	169
7.9.	(a) <i>Zoom</i> semántico sobre la clase <i>Actividad</i> , con siete propiedades objeto y cuatro propiedades de datos. La Figura (b) muestra el <i>zoom</i> semántico sobre la clase <i>Recorrido</i> , con ocho propiedades objeto y diez propiedades de datos.	171
7.10.	(a) <i>Zoom</i> semántico sobre la clase <i>Restaurante</i> , en donde se muestran los valores para cada de una las propiedades de una de las instancias seleccionada. La Figura (b) muestra el <i>zoom</i> semántico sobre la clase <i>Hotel</i> , en donde se muestran todas las instancias de hoteles con una clasificación de cuatro estrellas.	172
7.11.	La <i>table lens</i> muestra las instancias de la clase <i>Monumento</i> ; son dieciséis valores de diversos monumentos históricos ubicados dentro de la ciudad de Salamanca.	174
A.1.	Las opciones del menú <i>File</i> que contienen funciones para (a) abrir un archivo, (b) abrir una dirección URL y (c) guardar un modelo en un archivo con extensión “.owl”, y también para crear una nueva ontología.	190
A.2.	Las opciones del menú <i>File ->Database</i> que contiene las diferentes opciones para la funcionalidad de la base de datos.	191
A.3.	Las opciones del menú <i>File ->Add submodel</i> para agregar un submodelo al actual modelo ontológico.	193
A.4.	Las opciones del menú la funcionalidad del protocolo para transferencia de archivos FTP.	194
A.5.	Menú visual que contiene las principales funciones, como crear, modificar o eliminar clases, propiedades, instancias, enumeraciones o clases definidas como unión o intersección.	195
A.6.	El código OWL de la ontología <i>turisem.owl</i> se muestra con la opción <i>File ->OWL code</i>	196
A.7.	Las figuras (a) y (b) muestran las opciones <i>Help ->Help</i> y <i>Help ->About</i> . La figura (c) muestra la opción que permite habilitar o deshabilitar la opción del difuminado del fondo. Esta técnica puede deshabilitarse para optimizar el rendimiento de las visualizaciones, especialmente para grandes ontologías.	197

A.8. La opción del menú <i>Options</i> , permite reiniciar las visualizaciones a los valores por omisión (Figura (a)), mostrar estadísticas del modelo ontológico cargado (Figura (b)) y la búsqueda de clases en la taxonomía del modelo ontológico (Figura (c)).	198
A.9. La opción del menú para cambiar las diferentes visualizaciones disponibles en OWL-VisMod. Estas visualizaciones son el árbol de jerarquía (Figura (a)), el <i>treemap</i> (Figura (b)), la <i>table lens</i> y la vista del acoplamiento global (Figura (c)).	199
A.10. Las diferentes opciones del esquema de una ontología, para crear clases, propiedades o instancias. Este submenú se invoca al presionar click con el botón derecho del mouse, sobre el área de la visualización.	200
A.11.(a) El lenguaje OWL-DL permite crear clases definidas como la unión o la intersección de otras clases existentes en el modelo. (b) El lenguaje OWL-DL permite definir clases enumeradas. . .	200
A.12. La navegación a través de los diversos niveles del <i>treemap</i> se realiza con la opción <i>Treemap zoom</i> , invocada con el click derecho del <i>mouse</i> , colocado sobre la visualización.	201
A.13. La opción <i>Delete elements</i> permite eliminar clases, propiedades e instancias del modelo ontológico. La figura (a) muestra la opción que permite eliminar una clase, mientras que la figura (b) muestra las opciones para eliminar ya sea una propiedad o una instancia.	202
A.14. La principal funcionalidad del <i>table lens</i> incluye cambios de colores para las columnas (figura (a)), ocultar determinadas columnas (figura (b)) y el filtrado de datos (figura (c)).	203
A.15. El filtrado de clases permite analizar las clases en un cierto rango de acuerdo a las métricas de acoplamiento.	204
A.16. El filtrado de propiedades permite ocultar todas aquellas aristas que no sean del interés del usuario.	204
A.17. Las opciones en la vista de acoplamiento global de los modelos ontológicos. La figura (a) muestra la opción para realizar búsquedas de clases o propiedades. La figura (b) muestra la opción para tensar las aristas que representan las propiedades, y la figura (c) muestra la opción para cambiar el tamaño de las esferas que representan a las clases acopladas.	205

A.18.La opción que modifica el radio de la circunferencia con las clases, en la vista de acoplamiento global de los modelos ontológicos.	206
A.19.Las listas circulares que muestran las clases (figura (a)) y las propiedades (figura (b)) de los modelos ontológicos. En la lista de propiedades, el color identifica el tipo de la propiedad, el color rojo identifica a las propiedades objeto, mientras que el color verde identifica a las propiedades de datos.	207
A.20.Las figuras (a), (b) y (c) muestran el menú visual para la funcionalidad de las clases; rotar la lista, filtrar clases o la búsqueda de clases respectivamente. Las Figuras (d), (e), (f), (g) y (h) muestran el menú visual para la funcionalidad de las propiedades; rotar la lista, filtrar propiedades, búsqueda de propiedades, crear propiedades objeto y finalmente crear propiedades de datos respectivamente.	208

Capítulo 1

Introducción

Este primer capítulo comienza con una introducción a los principales conceptos relacionados con la tesis; por un lado, la representación del conocimiento, y por otro lado, las ontologías. La segunda sección se enfoca en describir la problemática que ha originado el desarrollo de esta tesis. La tercera sección describe los objetivos específicos del trabajo de tesis, para posteriormente dar paso a la metodología y las líneas de investigación. El capítulo finaliza con el contenido general de la tesis.

1.1. Presentación

La representación del conocimiento (*Knowledge Representation* o KR, por sus siglas en inglés) es un área de la Inteligencia Artificial, cuyo objetivo fundamental es representar el conocimiento de forma que facilite el desarrollo de inferencia, generalmente mediante el uso de algún tipo de lógica. Esta lógica proporciona una semántica formal a los símbolos en el dominio de discurso, así como operadores y cuantificadores que proporcionan un significado a las sentencias en la lógica. En términos generales, puede decirse que la representación del conocimiento es un conjunto de reglas ontológicas provistas con la capacidad para realizar inferencias [52].

Es importante distinguir claramente entre información y conocimiento; información se refiere a un conjunto de datos o valores, ya sea numéricos o de cualquier otro tipo, que se tienen como entrada al sistema, y que no han sido interpretados todavía. En contraste, el conocimiento es un conjunto de datos de primer orden, que modelan de forma estructurada los valores de un cierto dominio, o que surgen con la interpretación de los datos básicos. Este conoci-

miento es el resultado de la aplicación de un proceso de interpretación de la información (los datos básicos), también denominado proceso de inferencia.

La representación del conocimiento consta básicamente de un conjunto de símbolos de un dominio específico de aplicación, aunado a las funciones que permiten desarrollar el proceso de inferencia para conseguir un razonamiento. La representación del conocimiento puede verse desde dos perspectivas diferentes, aunque no excluyentes una de la otra. Por un lado, la representación enfocada al procesamiento de los datos de forma automatizada por los ordenadores, los que se encargan de desarrollar inferencia y obtener conocimiento sin la intervención humana; y, por otro lado, la representación del conocimiento desde la perspectiva del usuario como el centro principal del proceso de gestión y adquisición del conocimiento.

Un claro ejemplo del enfoque basado en la representación de conocimiento de forma automatizada, es la Web Semántica [26], en la que los ordenadores se comunican entre sí, intercambian información, la procesan, realizan un proceso de inferencia y, como resultado, se obtiene un cierto conocimiento. Esta capacidad de conducir razonamiento automatizado sin necesidad de la intervención humana, se logra proveyendo a los ordenadores de una representación de conocimiento basada en el uso de colecciones estructuradas de información, así como un conjunto de reglas de inferencia, las cuales añaden una lógica sobre los datos. El resultado puede resumirse como proporcionar a los ordenadores con una capacidad de toma de decisiones.

Algunos autores como Kashyap et al. [157], han documentado que en ciertos contextos como la biomedicina [136, 158], el diseño y construcción de ontologías, exige demasiados recursos en términos de coste y tiempo, por lo que, en estos contextos se plantea la menor intervención humana posible.

Por el contrario, el enfoque basado en la representación del conocimiento desde la perspectiva del usuario, requiere la implementación de mecanismos enfocados completamente a hacer uso de la capacidad cognitiva humana para desarrollar el proceso de creación de los modelos de datos. Estos mecanismos son completamente diferentes a los mecanismos para la representación automatizada, aunque, el resultado final sea siempre un modelo de datos formado por colecciones estructuradas de información, representadas de forma procesable por los ordenadores. La principal diferencia entre los dos enfoques, radica en quién es el encargado de realizar el proceso: el usuario o el propio ordenador.

El tamaño o la complejidad del dominio de aplicación, suelen ser los aspectos que definen quién es el encargado del proceso de representación del conocimiento. Para modelos pequeños o medianos (formados por algunos centenares de conceptos), suele ser el usuario el que construye los modelos, mientras que,

por el contrario, para modelos de dominio de aplicación demasiado grandes (miles o millones de conceptos), o demasiado complejos, puede resultar prácticamente imposible su desarrollo por parte de un usuario y se tiene que recurrir al uso de procesos automatizados.

Una de las principales formas de representación del conocimiento son las ontologías. Estas colecciones estructuradas de información se pueden definir como una conceptualización taxonómica de conceptos, propiedades y relaciones entre ellos, así como un conjunto de reglas de inferencia [12]. En términos generales, los sistemas basados en conocimiento necesitan requerimientos especiales para poder desarrollar interoperabilidad. Estos sistemas operan y se intercomunican mediante sentencias en una representación de conocimiento formal, llamada ontología [116].

Guarino [119] analiza las diferentes definiciones de ontología y las contrasta con las bases de conocimiento. Aunque existen diversas posibles interpretaciones, para el desarrollo del presente trabajo se asume la interpretación de que una ontología es un sistema conceptual, que se puede entender como una base de conocimiento en particular, por lo que se refiere a una o a otra de forma indistinta, de manera similar a otros trabajos [190, 191]. Aunque se puede hacer una diferencia más específica; por ejemplo, de acuerdo a los autores Fridman y McGuinness [79], una base de conocimiento está formada por una ontología y un conjunto de instancias de las clases, las cuales contienen valores específicos de información en las propiedades o *slots*, y en las restricciones.

Las ontologías se han sido utilizado ampliamente en el contexto de los sistemas de administración del conocimiento [25]; estas, representan un aspecto crucial en las tecnologías semánticas, incluso desde el punto de vista de la industria. Las tecnologías semánticas se han utilizado en diferentes áreas de aplicación, como por ejemplo en plataformas de tecnologías de información [48], en plataformas en el área de Biología [243], en el análisis de redes sociales [101], entre otras.

Un dominio de aplicación que también ha utilizado ontologías es el turismo. Un caso de estudio en este dominio de aplicación, se describe en capítulos posteriores. Básicamente, se han definido algunos modelos ontológicos para un sistema de recomendación, en el que, los usuarios reciben información sobre lugares turísticos, espectáculos, entre otras actividades a realizar, de acuerdo a un perfil personalizado. En este caso específico, las ontologías definen por un lado, un modelo de conocimiento que permite establecer perfiles de usuario, y por otro lado, una base de conocimiento con información sobre lugares y actividades turísticas. El sistema realiza un proceso de inferencia a partir del perfil personalizado (aficiones o preferencias principalmente), lo que resulta

en un conjunto de actividades que pueden ser del interés del usuario. Este sistema ejemplifica el uso de ontologías para inferir un cierto conocimiento automatizado.

Este trabajo de tesis abarca el enfoque en el que es el propio usuario el que desarrolla los procesos en el ciclo de vida de las ontologías. En este caso, es este usuario quien modela y construye la base de conocimiento. Para poder crear estas ontologías, es necesario proporcionar a los usuarios con diferentes mecanismos que los apoyen en todas las tareas necesarias, como por ejemplo, metodologías, lenguajes y herramientas.

La gran mayoría de las propuestas de metodologías han aparecido desde la década de los años noventa. Fernández [192] elaboró un resumen sobre las primeras propuestas metodológicas para ontologías, que más tarde complementó y se detalla en [193]. Las más importantes son Methontology [49], OCML [201], On-To-Knowledge [250, 68] o DOGMA [148]. Sin embargo, uno de los principales problemas, es que en la actualidad no hay una metodología estandarizada para la creación de ontologías, sino más bien, un conjunto de propuestas diversas, entre las que el usuario tiene que identificar la que mejor se adapte a sus necesidades.

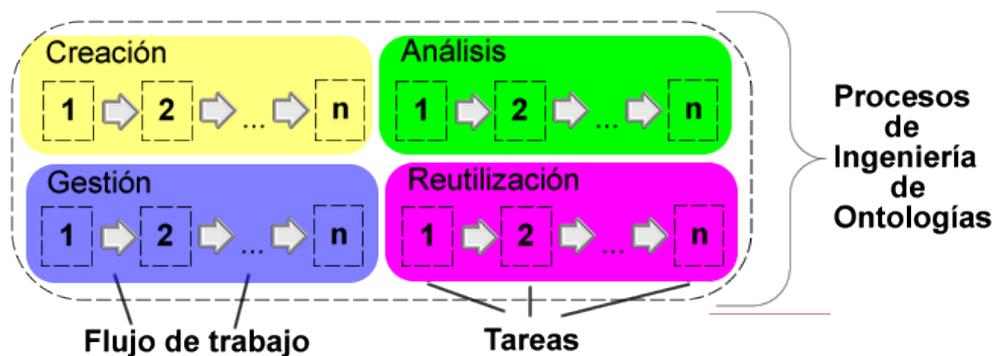


Figura 1.1: Los procesos básicos en Ingeniería de Ontologías son: creación, gestión, análisis y reutilización. Cada uno de los procesos define un flujo de trabajo formado por varias actividades o tareas.

Una oferta muy útil para los usuarios sería la definición de los procesos necesarios para realizar el modelado de ontologías, aunado a una herramienta específica que soporte estos procesos de forma eficiente. La Figura 1.1 muestra un esquema general básico, que contiene los principales procesos involucrados en el modelado y desarrollo de ontologías. En este esquema, se muestran los cuatro procesos básicos para el modelado de datos ontológicos: la creación, la gestión, el análisis y la reutilización. Cada uno de esos procesos, contiene un

flujo de trabajo definido por varias tareas que el usuario debe desarrollar para completarlo.

Cada uno de estos procesos, define flujos de trabajo específicos, aunque entre ellos comparten muchas de las actividades. Una de las más importantes metodologías de la Ingeniería de Ontologías es Methontology [71], que describe las actividades principales para el proceso de creación de ontologías se muestran en el esquema de la Figura 1.2. Esta secuencia de actividades comienza con la especificación de los requerimientos del sistema, y después continúa la abstracción del dominio de aplicación, la creación de un modelo conceptual, la formalización del modelo conceptual, una probable reutilización de modelos ontológicos existentes, y finalmente la propia implementación de la ontología en un lenguaje formal.

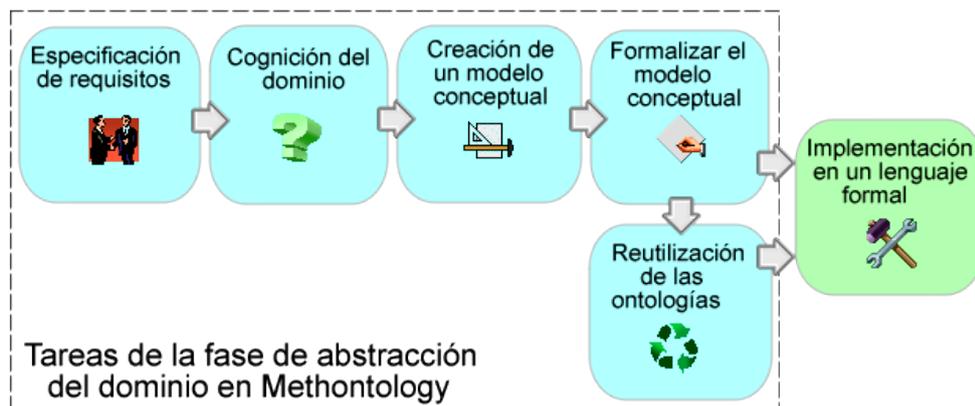


Figura 1.2: Principales tareas definidas en Methontology, para el proceso de creación de ontologías.

La especificación de los requerimientos generalmente involucra entrevistas con los usuarios finales del sistema, en las que se hace una descripción completa del funcionamiento del sistema a diseñar. Existen algunos estándares internacionales que definen las tareas concretas de la fase de especificación de los requerimientos, como el estándar IEEE 830 [198].

La segunda tarea en el esquema de la Figura 1.2, es la cognición del dominio de aplicación por parte del diseñador del modelo ontológico. Esta tarea implica principalmente, que la persona encargada de diseñar la ontología, obtenga conocimiento sobre el dominio de aplicación, ya que generalmente no es un experto en el tema. Una vez que el diseñador del modelo ha obtenido el conocimiento necesario sobre el dominio de aplicación, la tercera fase implica la conceptualización de este conocimiento, mediante la creación de uno o vari-

os modelos conceptuales. La siguiente tarea conlleva la formalización de estos modelos conceptuales creados. La siguiente tarea involucra la reutilización de ontologías; esta tarea es opcional, ya que se tiene la certeza de que exista un modelo ontológico que pueda utilizarse como solución para este dominio de aplicación en específico. Finalmente, la última tarea corresponde a la implementación de la ontología en un lenguaje formal.

Los flujos de trabajo y las actividades difieren entre las diversas propuestas de metodologías, pero en términos generales, hay cinco actividades importantes para la creación de un modelo de datos basado en ontologías, que se muestran en la Figura 1.3. El proceso de creación comienza con una fase de abstracción del dominio del problema, seguido por la definición de una taxonomía de conceptos, después se definen las propiedades intrínsecas, así como las relaciones entre los conceptos, para finalizar con la definición de las instancias que forman la base de conocimiento. El proceso de creación de ontologías, junto con estas actividades, se describe en detalle más adelante en la sección 5.1.1.

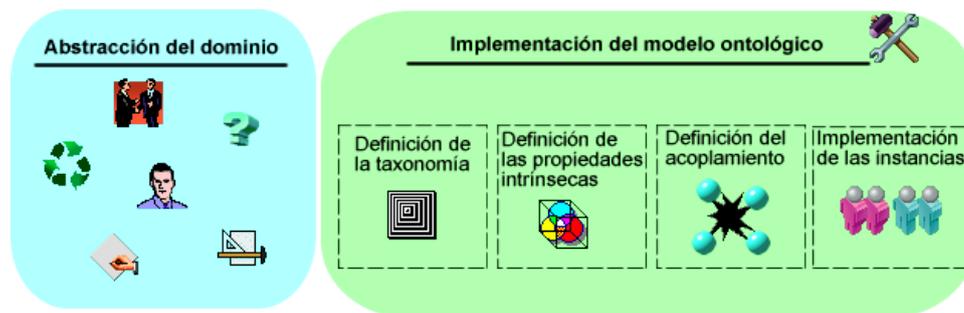


Figura 1.3: El proceso de creación de ontologías está definido por cinco tareas principales, divididas en dos grandes fases: una fase de abstracción del dominio, y otra fase para la implementación del modelo ontológico, que incluye la definición de la taxonomía de conceptos, la definición de las propiedades intrínsecas a las clases, la definición de las propiedades objeto que representan las relaciones, y la definición de las instancias de las clases, que forman la base de conocimiento.

Sea cual sea la metodología empleada, lo cierto es que el usuario tiene que desarrollar una fase de implementación de la ontología. Para llevar a cabo esta fase, las metodologías definen diferentes actividades que se llevan a cabo mediante el uso de ciertas herramientas, que permiten a los usuarios completar estas actividades. A lo largo de más de una década han surgido continuamente diversas herramientas, algunas de ellas comerciales y otras de libre distribución, entre las que algunas son de código abierto para que la propia comunidad investigadora sea la encargada de actualizarlas. Muchas de estas herramientas

han caído en desuso, no han sido actualizadas, e incluso algunas de ellas ya no están disponibles, por lo que actualmente se pueden considerar simplemente como una referencia bibliográfica.

Cuando se habla de representación del conocimiento, y más específicamente, de modelos ontológicos creados por el usuario, se pretende que las herramientas que permiten crear estos modelos se adapten a las necesidades y características de los usuarios. Una de las características que facilita este proceso, es el uso de representaciones visuales que apoyen todos los procesos básicos en el ciclo de vida de los modelos ontológicos, como la creación, gestión, análisis y reutilización. Estas representaciones visuales facilitan el proceso de representación del conocimiento, ya que utilizan la capacidad cognitiva de los usuarios a través de la vista, uno de los sentidos que permite más fácilmente asimilar información y obtener conocimiento.

Los métodos visuales se han utilizado para el aprendizaje a todos los niveles, desde los niños en las escuelas, hasta las conferencias más importantes alrededor del mundo, y pueden abarcar desde un simple dibujo hasta una complicada representación visual de una compleja reacción química. Estas técnicas han evolucionado hoy en día, gracias al avance de la tecnología y en especial al uso de ordenadores, hasta llegar a convertirse en complejas representaciones visuales capaces de procesar y visualizar grandes cantidades de información de una forma visualmente eficiente.

Es justo ahora, precisamente con el uso de ordenadores, que se pueden desarrollar complejas estructuras visuales, que permiten no solo representar grandes cantidades de información, sino que también están provistas con capacidades de interacción con el usuario, de tal forma que este puede manipular la información desplegada. Esta capacidad de interacción permite pasar de un modelo visual estático, a uno completamente dinámico, en el cual la información representada visualmente puede cambiar de acuerdo a las exigencias del usuario [39, 40].

El área de la representación del conocimiento no es ajena al uso de estos métodos visuales, que comenzaron con la simple representación en papel de mapas conceptuales, y que en la actualidad se ha visto enriquecida con diversas herramientas, tanto de visualización como de modelado [52]. Es importante destacar la diferencia entre una herramienta de visualización de ontologías y una herramienta de modelado de ontologías. La primera se limita simplemente a representar visualmente una ontología, sin opciones de edición o modificación, mientras que la segunda permite el desarrollo de diversas actividades para las fases de creación, gestión o incluso reutilización, directamente sobre el modelo visual.

1.2. Problemática

En términos generales, el desarrollo de un sistema informático conlleva la necesidad de modelar un cierto dominio de aplicación. Este desarrollo requiere un proceso interno de modelado de los datos del sistema. En otras palabras, la forma en que se representan, almacenan y acceden a los datos del sistema. Este modelo de datos representa un aspecto crucial para el buen desarrollo del sistema.

El proceso para crear un modelo de datos no es trivial, requiere la participación del diseñador como punto central, que es el que define el propio modelo, así como una serie de herramientas que le permiten llevar a cabo todo el proceso de modelado. Un ejemplo de estos modelos de datos es, el modelo entidad-relación para bases de datos. Análogamente, las ontologías representan un modelo de datos semántico para los sistemas de representación de conocimiento, que comparte la misma complejidad en cuanto a su diseño y construcción.

En el contexto de este trabajo, se adopta la definición de proceso como *el conjunto de actividades y eventos coordinados que se realizan o suceden bajo ciertas circunstancias con un fin determinado* [140]. Los procesos básicos para el desarrollo de ontologías descritos en el esquema de la Figura 1.1, se han definido en la mayoría de las metodologías existentes. Sin embargo, al no haber una metodología estándar, existe una diversidad de procesos, flujos de trabajo y actividades, en que únicamente, dependiendo de la experiencia y el nivel de conocimiento del diseñador de las ontologías, se puede tomar una decisión acerca de cuáles de todos estos llevar a cabo.

Noy [210] propone una guía para el desarrollo de ontologías, enfocada especialmente para desarrolladores noveles, que cubre las tareas más importantes para el proceso de creación de una ontología. A pesar de representar un buen punto de inicio, hay aspectos importantes que no se incluyen, como por ejemplo los mecanismos que permitan generar modelos ontológicos reutilizables, o la forma de realizar un análisis para determinar si un modelo existente puede ser reutilizado en el sistema.

El resultado de la falta de procesos adecuados, es la creación de ontologías con baja calidad, difíciles de actualizar [174], reutilizar e incluso con inconsistencias. Estas ontologías tienen que someterse a diferentes procesos de evaluación [211] para detectar problemas y poder corregirlos. Uno de los problemas principales en los modelos ontológicos, son las inconsistencias terminológicas. Ovchinnikova et al. [216] proponen un algoritmo para corregir las inconsistencias definidas por el usuario, especialmente la inconsistencia lógica, aunque

también se presentan otro tipo de inconsistencias, especialmente como resultado de operaciones como la alineación de ontologías [65].

Por otro lado, Guarino [118] destaca la importancia en el diseño de ontologías reutilizables; pero esto únicamente puede lograrse, con procesos bien definidos de Ingeniería del Conocimiento, así como con herramientas robustas que soporten estos procesos. No solo es importante considerar el concepto de reutilización, sino también el propio mantenimiento de las ontologías creadas.

Arpírez et al. [15] han resumido la principal problemática para el desarrollo de ontologías. El primer problema descrito es la falta de una correspondencia entre las metodologías existentes y las herramientas para construir ontologías. El segundo problema que se menciona es que las herramientas existentes únicamente dan soporte al diseño e implementación de ontologías, pero no soportan todas las actividades del ciclo de vida. En base a esta problemática, Arpírez et al. [14] propusieron un marco de trabajo llamado WebODE, una plataforma que cubre y da soporte a la mayoría de las actividades involucradas en el proceso de desarrollo de ontologías (conceptualización, razonamiento, intercambio, etc.). Este editor de ontologías ha quedado sin soporte desde el 2006, y no se encuentra disponible en la actualidad.

La calidad de las ontologías es un aspecto crucial, que puede verse afectado severamente por la problemática descrita previamente. Guarino y Welty [120, 121] propusieron una metodología llamada OntoClean para analizar las ontologías desde un punto de vista lógico, para evaluar la calidad de las ontologías, y poder detectar y corregir problemas que vienen desde la fase de diseño. Sin embargo, esta solución está enfocada a ontologías ya existentes, pero no incide directamente sobre los procesos de creación, gestión o reutilización.

Cuando un usuario pretende definir un modelo de datos semántico basado en ontologías, se enfrenta con el dilema de ¿por dónde empezar?, ¿qué tareas desarrollar?, ¿cuál es la secuencia de esas tareas?, ¿se pueden reutilizar modelos existentes?, ¿qué herramienta utilizar para implementar el modelo?, ¿cómo desarrollar ontologías de calidad?.

La Figura 1.3 muestra las principales actividades definidas para el proceso de creación de una ontología. La primera fase que corresponde a la fase de abstracción del dominio, está definida completamente por el diseñador del modelo, que se apoya en los métodos y las metodologías existentes, y que en base a estos y a su conocimiento, realiza un planteamiento de solución. Si bien es verdad que existen una infinidad de soluciones al problema de modelado de un dominio, también es verdad que hay soluciones que generan ontologías con una mucho mayor calidad que otras. El problema del diseño de ontologías con una

baja calidad, es atribuible al diseñador del sistema, y es causado directamente por fallos en los mecanismos empleados por este para llevar a cabo el diseño.

Una vez realizada la fase de abstracción del dominio, la siguiente fase que se muestra en el esquema de la Figura 1.3 es la fase de implementación del modelo diseñado. Es en esta segunda fase, en la que se presenta otra problemática que repercute directamente en el proceso de la creación de ontologías, esta problemática está ligada a las herramientas de edición y modelado.

Los diseñadores requieren el uso de diversas herramientas para modelar las ontologías, y para la implementación de estos modelos (última tarea definida en la Figura 1.2). Uno de los principales problemas con las actuales herramientas para la creación y la gestión de modelos de representación de conocimiento es la falta de procesos adecuados para estas herramientas (como se ha mencionado previamente).

En el caso específico de la implementación de los modelos para la representación del conocimiento, las herramientas actuales únicamente se limitan a ofrecer un conjunto de visualizaciones o interfaces de usuario que no son adecuadas para las metodologías existentes. Sin un flujo de trabajo claramente definido, ni herramientas robustas de soporte, los procesos de reutilización, creación o gestión de modelos de conocimiento pueden llevarse a cabo de forma ineficiente y se vuelven extremadamente complejos para el diseñador.

El problema principal con las actuales herramientas visuales, es que no permiten el desarrollo eficiente de los procesos básicos de la Ingeniería de Ontologías (mostrados en la Figura 1.1). Este área de investigación (definida formalmente en la sección 2.3), proporciona las directrices del conjunto de actividades para los procesos involucrados en el ciclo de vida de las ontologías. La principal causa es la complejidad derivada del uso de visualizaciones poco robustas, demasiado simples o con una interacción pobre con el usuario. Estas visualizaciones y la interacción con el usuario, resultan completamente ineficientes cuando los modelos ontológicos van creciendo en tamaño.

Este problema se ilustra claramente en la Figura 1.4, que muestra una herramienta durante la fase de implementación de un modelo ontológico. En este caso, la herramienta no tiene la capacidad de representar el modelo de forma clara, sino que, por el contrario, el resultado es una vista saturada que primero, impide entender exactamente el modelo que se construye, y segundo, vuelve compleja la fase de implementación.

Una representación visual, tiene como principal finalidad permitir al usuario adquirir un cierto conocimiento a partir de los símbolos, colores y formas que percibe, y este hecho debería ser directo e intuitivo. Cuando se habla de on-

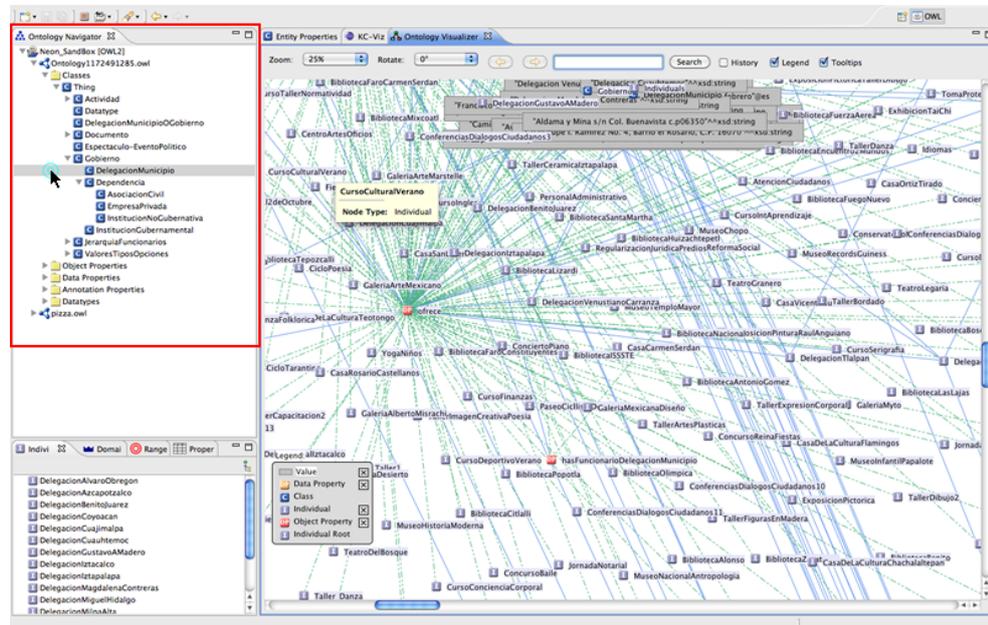


Figura 1.4: La fase de implementación se vuelve extremadamente compleja, debido al uso de herramientas poco robustas. La saturación visual de elementos complica en demasía el entendimiento del dominio que se modela.

tologías, la representación visual debería claramente transmitir al usuario el sistema que se ha modelado o que se está modelando. Por ejemplo, dado que las ontologías contienen un esquema definido por una taxonomía de conceptos, la interacción debería permitir al usuario seleccionar un concepto e inmediatamente conocer su jerarquía y sus relaciones, o navegar entre sus instancias, etc. Esta interacción es fundamental para que el diseñador o el usuario según sea el caso, puedan claramente obtener o modelar el conocimiento que se pretende.

La herramienta mostrada en la Figura 1.4, también falla en cuanto a la interacción con el usuario. El recuadro en color rojo resaltado en la esquina superior izquierda, muestra la taxonomía de los conceptos, en donde uno de los elementos se ha seleccionado. Esta selección muestra la taxonomía de la clase, pero también debería permitir al usuario o al diseñador, identificar las relaciones con otras clases o las instancias en caso de que existan. Esta situación no se presenta, ya que la interacción no es lo suficientemente buena como podría ser, por lo que el único mecanismo para que el usuario o el diseñador pueda conocer las relaciones o las instancias, es adentrarse en una representación visual saturada de elementos.

Como se ha descrito, las actuales propuestas fallan principalmente en dos

aspectos: primero, por el uso de técnicas de visualización muy básicas, que fácilmente se ven rebasadas en su objetivo de visualizar y modelar eficientemente ontologías y bases de conocimiento y, segundo, debido a que la interacción de estas herramientas tampoco es lo suficientemente robusta para permitir al usuario un análisis o modelado intuitivo y sencillo. Esta situación se ha ilustrado con una de las herramientas mostrada en la Figura 1.4, pero se describe en detalle en la sección 2.3.4 con el resto de las herramientas.

Las ontologías representan un aspecto mucho más complejo que la representación de una simple taxonomía de conceptos y un conjunto de relaciones entre clases. Es justo en este aspecto, en la propia naturaleza de las ontologías, donde radica la problemática de su visualización y modelado de forma eficiente. Si bien es cierto que no existe un método que parezca ser el apropiado para representar visualmente las ontologías en general, existen soluciones específicas que pueden ajustarse a cada uno de los problemas de forma individual, desde una perspectiva diferente a la que han seguido la totalidad de las soluciones implementadas en la actualidad.

En resumen, la problemática en la creación de ontologías de calidad tiene dos vertientes. Por un lado, la diversidad de propuestas metodológicas sin que exista una que resulte claramente dominante, que abarque todos los procesos necesarios, y permita construir modelos ontológicos de calidad. Esta situación incide directamente en los diseños de las ontologías. Esta problemática está directamente relacionada con el usuario que es el que se encarga de diseñar los modelos ontológicos y en donde, como resultado, se generan diseños de ontologías con una baja calidad.

La segunda vertiente es la falta de correspondencia entre las metodologías y las herramientas, así como el hecho de que las herramientas sean poco robustas, lo que dificulta en demasía el desarrollo de los procesos de la Ingeniería de Ontologías. Esta problemática es atribuible directamente a las herramientas existentes, ya que, como se ha descrito en párrafos anteriores, se limitan al uso de técnicas simples que fácilmente son llevadas al límite de sus capacidades o rebasadas, y tampoco implementan mecanismos de interacción eficientes.

Cuando estas dos vertientes se conjugan, los resultados son claramente negativos, ya que se presentan modelos ontológicos de baja calidad, y también se presenta un alto grado de complejidad para desarrollar los procesos de análisis, implementación y reutilización.

1.3. Objetivos

El trabajo de esta tesis básicamente propone el planteamiento de soluciones a la problemática de la representación visual y el modelado de datos basado en ontologías. Estas soluciones deben ser planteadas desde una perspectiva diferente a las actuales propuestas, con el objetivo de hacer eficientes los diversos procesos involucrados en la Ingeniería de Ontologías. Más específicamente, se requiere la propuesta de diversos flujos de trabajo y actividades para el desarrollo de los modelos de datos basados en ontologías, y su ciclo de vida.

El primer objetivo general es la definición de los flujos de trabajo para desarrollar los procesos de creación, gestión o mantenimiento, análisis y reutilización de representaciones basadas en conocimiento, más específicamente, de ontologías. Existen algunas metodologías propuestas, que han definido una diversidad de estos procesos, flujos de trabajo y las actividades dentro de estos últimos. Esto simplifica considerablemente el trabajo en esta vertiente, ya que existe un camino recorrido previamente por otros autores.

El segundo objetivo comprende la implementación de una herramienta que soporte el desarrollo de los procesos definidos, que sea suficientemente robusta en cuanto a las visualizaciones y la interacción con el usuario, para poder modelar bases de conocimiento semántico de forma eficiente. Esta vertiente representa un aspecto fundamental a conseguir en este trabajo de tesis, ya que la problemática existente con las actuales herramientas, obliga a pensar en soluciones nuevas y más eficientes.

Como base de la investigación, se plantea como principal hipótesis, la siguiente:

¿Es posible diseñar procesos eficientes y robustos para la creación, gestión, análisis y reutilización de bases de conocimiento, que faciliten al diseñador la exploración, análisis y modelado de datos?

Para plantear soluciones a esta hipótesis, la investigación se basa en el enfoque que la Analítica Visual define para la visualización, exploración y análisis de datos, ya que este enfoque basado en explotar la capacidad cognitiva humana, junto a los potentes métodos de procesamiento de datos, ha permitido el desarrollo de diversas herramientas y soluciones exitosas en otros contextos.

De acuerdo a la hipótesis y a los objetivos generales descritos anteriormente, a continuación se enumeran los objetivos específicos de este trabajo de tesis:

1. La definición de los flujos de trabajo en los procesos de creación, gestión o mantenimiento, análisis y reutilización de las bases de conocimiento, bajo el esquema de la exploración, análisis y modelado de datos definido

en el área de Analítica Visual.

2. La definición de las técnicas visuales y de interacción que soporten cada una de las tareas en los flujos de trabajo definidos.
3. La integración de las visualizaciones en una herramienta que soporte completamente los procesos antes descritos.
4. El desarrollo del/los caso(s) de estudio, que permitan evaluar los procesos y los productos resultantes.

1.4. Metodología de investigación

La investigación comprende dos grandes enfoques metodológicos: el primero corresponde a una revisión sistemática de la literatura [131, 171], para poder establecer el estado del arte al comienzo del desarrollo de la tesis; el segundo se basa en una serie de ciclos investigación-acción [73], a medida que se va desarrollando el trabajo de tesis, con el objetivo de solucionar problemas específicos.

Una revisión sistemática de la literatura representa generalmente la fase inicial en cualquier proyecto de investigación y desarrollo [29]. Es la forma en la que los investigadores pueden obtener el conocimiento existente en cuanto a una cierta línea de investigación. El primer paso corresponde a la búsqueda en la literatura de artículos relevantes para poder detectar soluciones a problemas similares o para poder analizar los procedimientos llevados a cabo por otros investigadores.

Para el caso específico de este trabajo de tesis, la revisión sistemática comprende las siguientes áreas:

- Los sistemas de representación de conocimiento, ontologías y bases de conocimiento. Específicamente, las metodologías para el desarrollo de las bases de conocimiento, las métricas, así como las herramientas para la creación, gestión, análisis y modelado.
- Procesos de análisis y soluciones basadas en Analítica Visual a diversos problemas similares.
- Soluciones basadas en el área de Visualización de Información (*Infovis*) a problemas similares.

Una vez elaborada la revisión sistemática, así como el análisis de la información obtenida, la siguiente fase corresponde al planteamiento e implementación de las soluciones específicas a la problemática descrita. Primero, la elaboración de las diversas propuestas para cada una de las actividades de los flujos de trabajo en los procesos de creación, gestión, análisis y reutilización de los modelos ontológicos, bajo el enfoque definido para el análisis de datos en el área de Analítica Visual. Segundo, la implementación de las técnicas visuales más adecuadas de acuerdo a estas actividades, así como el diseño de la interacción con el usuario. Tercero, la implementación de todas estas visualizaciones en una herramienta, que soporte el desarrollo de los procesos antes mencionados.

Durante la fase de implementación de las técnicas visuales, se desarrollan de manera paralela una serie de ciclos investigación-acción, muy específicos sobre el proceso y las soluciones más adecuadas para su desarrollo. Estos ciclos complementan la revisión sistemática, y representan una revisión muy específica y puntual de la literatura sobre ciertos aspectos del estado actual del desarrollo.

Finalmente, una vez implementadas las técnicas visuales, la interacción con los usuarios y la implementación de la herramienta final, la última fase corresponde a la elaboración de los casos de estudio que permitan evaluar por un lado, la usabilidad de las actividades y flujos de trabajo, y por otro lado, la funcionalidad de la herramienta final.

1.5. Líneas de investigación de la tesis

La presente tesis ha sido desarrollada dentro del marco de las líneas de investigación de la Ingeniería del Conocimiento, Visualización de Información y Analítica Visual, dentro de dos grupos de investigación de la Universidad de Salamanca, por un lado, el *Grupo de Investigación en Interacción y eLearning* (GRIAL)¹, y por otro lado el *Grupo de Visualización de la Universidad de Salamanca* (VisUSAL)².

Este trabajo de tesis se relaciona con otros proyectos del grupo de investigación GRIAL; uno de ellos se basa en el uso de Analítica Visual en el área de *eLearning* [54, 105]. Otros trabajos de investigación en el área de Analítica Visual dentro del grupo VisUSAL, en el área de fármacos [82] y en el área de evolución del software [113].

Los proyectos de investigación en los que se ha ubicado la tesis son el

¹<http://grial.usal.es/>

²<http://vis.usal.es/~visusal/grupo/>

proyecto *Turimov* y el proyecto *Ulises* en colaboración con la empresa Interra³. Estos dos proyectos están relacionados con el desarrollo de una plataforma para teléfonos móviles, en la que los turistas pueden elaborar recorridos turísticos en la región de Castilla y León. Esta plataforma se basa en el uso de ontologías con conceptos relacionados con el turismo y bases de conocimiento con instancias específicas de la región de Castilla y León. Otros proyectos dentro de los cuales esta tesis ha colaborado y/o recibido apoyo, son el proyecto del Gobierno de Castilla y León GR47 y los proyectos del Ministerio de Ciencia e Innovación FI2010-16234 y TIN2010-21695-C02-01.

1.6. Contenido

El presente trabajo se ha dividido en ocho capítulos, cada uno de los cuales se resume a continuación.

En este primer capítulo se ha descrito la problemática involucrada en el desarrollo de la Ingeniería de Ontologías, específicamente aquella relacionada con los procesos definidos para el desarrollo del ciclo de vida de las ontologías, así como la problemática de las actuales herramientas para la edición de ontologías. También se han definido los objetivos del presente trabajo de tesis, basados en los planteamientos resultado del análisis de la problemática discutida. Se ha descrito la metodología utilizada durante el desarrollo de esta tesis, y finalmente las líneas de investigación con las que se asocia.

El segundo capítulo está dedicado a describir de forma más específica las ontologías, las bases de conocimiento y su funcionalidad. También proporciona una descripción de las metodologías y los lenguajes utilizados para la representación de ontologías. Finalmente, el capítulo concluye con un resumen de las principales herramientas de visualización y edición actualmente existentes.

El tercer capítulo está dedicado a describir las áreas de Visualización de Información y Analítica Visual, dos áreas de investigación con mucho auge actualmente, que se han tomado como base para la implementación de esta propuesta de tesis.

El cuarto capítulo hace una discusión del estado del arte de esta investigación, se comienza por analizar las más importantes metodologías para el proceso de creación de las ontologías, para aprovechar sus respectivas ventajas. La segunda parte del capítulo se enfoca al análisis y discusión de las principales herramientas para la gestión de ontologías, y se describe la principal problemática con ellas.

³<http://www.interra.es/>

El quinto capítulo define completamente la propuesta objeto de esta tesis, dividido en dos secciones principales que abordan las diversas soluciones implementadas. La primera parte, describe los flujos de trabajo en los procesos básicos de la Ingeniería de Ontologías: creación, gestión, análisis y reutilización. La segunda parte describe los mecanismos de soporte a las fases de implementación y análisis de ontologías.

El sexto capítulo describe la evaluación de funcionalidad sobre los procesos definidos, y sobre la herramienta que los implementa. Esta evaluación se desarrolló con un grupo de usuarios a los que se pidió crear un modelo ontológico desde el principio según las actividades definidas en los flujos de trabajo descritos en el capítulo 5.

El séptimo capítulo describe el desarrollo de un caso de estudio de un sistema basado en ontologías. Este caso de estudio corresponde al modelado de los datos de un sistema de recomendación aplicado al dominio del turismo.

El octavo capítulo discute los resultados obtenidos y las conclusiones de este trabajo de tesis y, finalmente se agregan los apéndices correspondientes.

Capítulo 2

Ontologías

Este capítulo se enfoca en la descripción de las ontologías desde el punto de vista de la representación del conocimiento, desde el que se citan las definiciones más ampliamente difundidas dentro del ámbito de la informática. Así mismo, se describe la estructura de las ontologías y su proceso de creación.

La segunda sección describe algunas de las aplicaciones de las ontologías, con especial atención a su uso en la Web Semántica, uno de los más importantes campos de aplicación. La tercera sección describe lo que es la Ingeniería de Ontologías, en donde también se analizan las principales propuestas metodológicas. Después, se resumen los lenguajes para su representación, se describen las principales métricas, así como las más importantes herramientas para la edición y modelado de ontologías, y, para terminar este capítulo, se discuten las conclusiones.

2.1. ¿Qué es una Ontología?

El término ontología se relaciona principalmente con la filosofía y la metafísica y trata de la descripción del ser en general a través de sus propiedades trascendentales. Este término se ha adoptado en el área de informática, y se han dado algunas definiciones más específicas para este área. La más difundida en la bibliografía es la definición de Gruber [116], que establece que en el contexto de las ciencias de la información y la informática, *una ontología define un conjunto de primitivas con las cuales se modela un dominio de conocimiento o discurso, en el que, dichas primitivas son típicamente clases, atributos y relaciones entre las clases*. En este contexto, las clases son representaciones de conceptos en el modelo de datos semántico, en tanto que las primitivas pueden

incluir también información acerca del significado y restricciones lógicamente consistentes.

Otra definición describe a una ontología como *una descripción formal y explícita de conceptos en un dominio de discurso (clases), propiedades de los conceptos que describen características y atributos (slots) y restricciones sobre las propiedades (facets)* [210]. Se dice que una ontología es una especificación formal y explícita de una conceptualización compartida. El término *formal*, se refiere a que una máquina pueda ser capaz de interpretar su significado, mientras que el término *explícita* se refiere a que los conceptos y las restricciones estén explícitamente definidos.

De forma muy general, las ontologías están formadas principalmente por definiciones de conceptos básicos en el dominio de aplicación y las relaciones entre estos; más específicamente, la estructura de las ontologías está formada por los siguientes elementos:

- **Conceptos:** Son las ideas básicas que se intentan formalizar. Los conceptos pueden ser clases de objetos, métodos, planes, estrategias, procesos de razonamiento, etc. La forma más común de definir los conceptos es mediante clases, que no son más que un molde en el cual se definen las propiedades y el estado interno que tendrán los objetos.
- **Propiedades:** Las propiedades son las características intrínsecas a cada concepto, que permiten discernir claramente entre uno y otro.
- **Relaciones:** Representan la interacción y enlace entre los conceptos del dominio y especifican las dependencias entre unos y otros. Una relación está definida por un dominio, formado por una o más clases a las que se les aplica la relación directamente, mientras que el rango está definido por una o más clases que pueden ser aplicadas a la relación.
- **Funciones:** Son un tipo concreto de relación donde se identifica un elemento mediante el cálculo de una función que considera varios elementos de la ontología.
- **Instancias:** Representan objetos determinados de una clase, son las implementaciones específicas de un concepto, con valores concretos a sus propiedades, que permiten establecer unicidad entre todos ellos.
- **Axiomas:** Son teoremas que se declaran sobre relaciones que deben cumplir los elementos de la ontología. Por ejemplo: “Si A y B son instancias de la clase C, entonces A no puede ser subclase de B”.

Las relaciones, vistas desde una perspectiva de la lógica matemática, establecen un vínculo formado por dos partes, el dominio y el rango. El dominio de una propiedad limita las instancias a las cuales la propiedad puede aplicarse, mientras que el rango limita las instancias que la propiedad puede tener como su valor.

Las relaciones, así como su dominio y rango, se pueden enriquecer con diversas características y restricciones, que definen y limitan la cardinalidad, el tipo de valores requeridos o características propias de la lógica. La cardinalidad limita el número de valores que una clase puede tener con respecto a una propiedad, esto implica que se puede definir un número mínimo, máximo o un número fijo de valores para una cierta propiedad en una determinada clase. Así mismo, se puede determinar no solo el número, sino también el tipo de valores que la propiedad puede tener con respecto a una clase concreta. Básicamente, se puede definir que todos los valores pertenezcan a una clase en específico o que, por el contrario, puedan pertenecer a diferentes clases, esto limita las instancias posibles en el rango de la propiedad.

Por otro lado, también pueden definirse ciertas características lógicas sobre las propiedades. La primera característica es la simetría, que indica que para cualesquiera dos valores (x, y) en el dominio y en el rango de la propiedad respectivamente, se cumple que (y, x) también satisfacen la misma propiedad. La segunda característica es la transitividad, que establece que si un par (x, y) es una instancia de una propiedad transitiva P , y también el par (y, z) es instancia de la misma propiedad transitiva, entonces (x, z) también es una instancia de la misma propiedad P .

Otra característica es la posibilidad de definir propiedades inversas. Una propiedad P definida como inversa de Q implica que si X está relacionada con Y a través de P , entonces Y estará relacionada con X a través de Q . Una propiedad también puede restringirse a tener un único valor para cada instancia, esta característica define a una propiedad funcional.

Todos los componentes en una ontología (conceptos, relaciones, propiedades, etc.), se representan e identifican mediante el uso de diferentes URI, un identificador uniforme de recurso definido en el RFC 3986. Básicamente, un URI es un identificador formado por una cadena de caracteres, que identifica un recurso inequívocamente. Una de las características de los URI es la uniformidad, que permite que diferentes tipos de identificadores de recursos se usen en el mismo contexto, aun cuando los mecanismos utilizados para acceder a esos recursos pueden ser diferentes.

Un aspecto fundamental en el uso de ontologías, es la capacidad de estas para inferir o deducir conocimiento. Esta capacidad se fundamenta principal-

mente en el uso de la lógica descriptiva asociada, que se basa en representaciones lógicas de conocimiento y el razonamiento sobre dicho conocimiento. Esta lógica descriptiva se define a través de restricciones y reglas de inferencia, que permiten deducir conocimiento a partir del modelo ontológico.

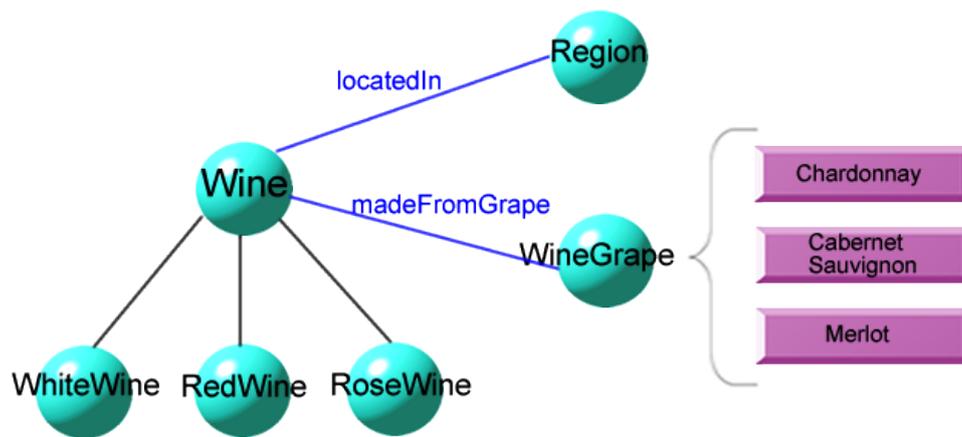


Figura 2.1: El esquema de algunos de los principales conceptos de la ontología *wine.owl*, sus relaciones y las instancias de la clase *WineGrape*, que define los tipos de uvas existentes.

La Figura 2.1, muestra un pequeño esquema de los principales conceptos en la ontología *wine.owl*¹). Esta ontología que se basa en el dominio de los vinos, es uno de los ejemplos más comúnmente usados para describir un modelo ontológico. En este esquema, se ilustran algunos de los principales conceptos del dominio: *Wine*, *WineGrape*, *Region*, *WhiteWine*, *RedWine* y *RoseWine*. Las clases *WhiteWine*, *RedWine* y *RoseWine* son subclases de *Wine* puesto que son una especialización de la clase.

En cuanto a las relaciones entre conceptos, destaca *madeFromGrape*, que relaciona a la clase *Wine* en el dominio con la clase *WineGrape* en el rango; la interpretación lógica de la relación sería decir que *el vino está hecho de algún tipo de uva*. Otra de las relaciones entre conceptos es *locatedIn*, que relaciona a la clase *Wine* en el dominio con la clase *Region* en el rango. La interpretación lógica sería decir que *el vino se localiza o proviene de alguna región específica*. Las instancias de una ontología corresponden a valores específicos de las diversas clases, por ejemplo las instancias *Chardonnay*, *CabernetSauvignon* y *Merlot* corresponden a algunos tipos específicos de uvas de la clase *WineGrape*.

¹<http://krono.act.uji.es/Links/ontologies/wine.owl/view>

2.2. Aplicaciones de las Ontologías

Las ontologías constituyen la forma más difundida para la representación y administración del conocimiento, y uno de sus principales objetivos es el de compartir este conocimiento entre personas o agentes de forma clara y sin ambigüedades [263, 25, 115]. Esta funcionalidad de interoperabilidad entre sistemas heterogéneos e incluso bases de datos, se ofrece principalmente a través de diversos servicios web, aunque existen diversas áreas en las que las ontologías se usan para proveer este tipo de servicios de interoperabilidad, por ejemplo en portales web [149] que proporcionan información específica de algún tema o dominio de interés, como pueden ser el clima o noticias.

Un ejemplo de estas plataformas es ODESeW² (*Semantic Web Portal based on WebODE platform*), que administra un portal de conocimiento para intranets y extranets. Esta aplicación se encuentra diseñada en la parte superior de la plataforma WebODE, un marco de trabajo para desarrollar Ingeniería de Ontologías [15, 110]. En términos generales, los portales de conocimiento presentan vistas estructuradas de la Web de acuerdo a lo que usualmente es un catálogo de conocimiento, que almacena descripciones acerca de los recursos disponibles para los miembros de la comunidad y es más compleja y flexible que las bases de datos comunes.

Otra aplicación de las ontologías es la reutilización del conocimiento que representan, lo que facilita que diversas aplicaciones que necesiten hacer uso de dicho conocimiento, puedan recibirlo de otras aplicaciones que lo hayan compartido previamente y, a su vez, compartirlo con nuevas aplicaciones [263]. La reutilización del conocimiento se relaciona en términos generales, pero no exclusivamente, con la especificación de la Web Semántica. En este contexto, las ontologías forman parte de los estándares W3C y se utilizan para especificar vocabularios conceptuales estandarizados para poder intercambiar datos entre diferentes sistemas, así como para proveer servicios para resolver consultas o publicar bases de conocimiento reutilizables.

Posiblemente la Web Semántica sea el campo de aplicación más importante para el uso de ontologías. La Web Semántica surge a partir de la publicación de un artículo en el año 2001 en la revista *Scientific American*, por parte de Tim Berners-Lee [26], creador de la World Wide Web. A partir de entonces, se han venido definiendo una serie de modificaciones que han permitido evolucionar a partir del concepto original, siempre con el objetivo de mejorarlo. No obstante, la visión original de proveer de una estructura y un contenido semántico a las páginas web, permanece como la principal filosofía detrás de la Web Semántica.

²<http://mayor2.dia.fi.upm.es/oeg-upm/index.php/es/downloads/74-odesew>

Se puede pensar como una extensión de la Web, en la que la información se proporciona con un significado bien definido, lo que permite la comunicación y el intercambio de información entre diversos equipos de cómputo de forma efectiva. El beneficio detrás de esta comunicación efectiva, conlleva el uso de la lógica matemática, que debe ser suficientemente potente como para describir propiedades complejas de objetos y, por otro lado, permitir el uso de reglas para hacer inferencia.

La Web Semántica permite a las aplicaciones decidir entre las necesidades de los usuarios y la información disponible de los recursos, debido a que el contenido web está representado de una forma completamente procesable por las máquinas, que se encargan de mantener dicha información, realizar búsquedas precisas y razonamientos lógicos [196]. Incluso, la Web Semántica se ha combinado con algunas otras áreas como la minería de datos [205], en una nueva área de investigación que se denomina *Semantic Web Mining*.

Aparte del uso de las Ontologías en la Web Semántica, existen otras diversas áreas de aplicación. Una de estas áreas es la clasificación de elementos multimedia, en donde un ejemplo es la ontología denominada COMM (*Core Ontology for Multimedia*)³. Básicamente, este tipo de aplicaciones consisten en proveer información sobre diversas colecciones de imágenes, audio o vídeo, que permiten hacer una clasificación de los recursos de acuerdo a diversos criterios [104, 28].

Otro tipo de aplicaciones que utilizan ontologías comprenden la administración de sitios web corporativos (especialmente cuando estos contienen una gran cantidad de documentos), páginas web o archivos en diferentes formatos. En esta situación, todos estos documentos se indexan para su posterior recuperación o consulta [266].

También se suelen utilizar modelos ontológicos para la representación de diversos conceptos especializados en las áreas de *Ingeniería de Software* [127, 267], *e-learning* [240], Geografía [76, 77], Biología [75, 243], Biomedicina [191], entre otras áreas.

En términos muy generales, las ontologías han sido ampliamente utilizadas en tres grandes áreas: la Ingeniería del Conocimiento, la Inteligencia Artificial y las Ciencias de la Computación. En términos muy específicos, principalmente en aplicaciones relacionadas con la administración del conocimiento, procesamiento del lenguaje natural, comercio electrónico, recuperación de información, integración de bases de datos, bioinformática, educación, y como se ha mencionado, son cruciales para el desarrollo de la Web Semántica [111].

³<http://comm.semanticweb.org/>

2.3. Ingeniería de Ontologías

Una vez descritas tanto las definiciones como las aplicaciones de las ontologías, esta sección se enfoca en describir el proceso de desarrollo de las ontologías. Existe un área de investigación específica para este proceso de desarrollo, denominada Ingeniería de Ontologías [110, 111]. La Ingeniería de Ontologías puede considerarse como una metodología de investigación que proporciona el diseño racional de una base de conocimiento [197]. También permite la conceptualización de un dominio de interés, las restricciones semánticas de los conceptos, junto con teorías y tecnologías que permiten la definición de conocimiento. Una de las principales finalidades es el procesamiento de este conocimiento por aplicaciones del mundo real.

Por otro lado, la Ingeniería de Ontologías proporciona las directrices para el conjunto de actividades propias de los procesos que cubren el ciclo de vida de las ontologías, así como también los diversos aspectos involucrados en estos procesos, como son las metodologías, los lenguajes y las métricas. Cada uno de estos aspectos se describen en detalle en las secciones 2.3.1, 2.3.2 y 2.3.3, respectivamente.

2.3.1. Metodologías para la creación de Ontologías

Las metodologías surgen por la necesidad de hacer más eficientes los procesos para la creación de las ontologías. Corcho et al. [49] hacen una recopilación y un análisis de las principales metodologías, las cuales se resumen en los siguientes párrafos. Uschold y King [263] son algunos de los investigadores pioneros en este área. Ellos proponen una metodología llamada *Knowledge Engineering Methodology* [264], como resultado de su propia experiencia en la construcción de la ontología *Enterprise* [265]. Esta metodología considera el problema de la categorización o la captura, esto es, la identificación de los conceptos clave y las relaciones en el dominio de interés. Después, Uschold y King describen una segunda fase de codificación, esto es, la representación del problema conceptual, en un lenguaje formal. También plantean el esquema de reutilización de modelos existentes, y la documentación de los modelos resultantes [262].

Esta metodología representa en gran medida, una de las bases para las posteriores propuestas metodológicas. La Figura 2.2 muestra las seis etapas definidas por Uschold y King para su propuesta de metodología: definición del dominio, propósito y el alcance de la ontología, conceptualización del conocimiento del dominio de aplicación, reutilización de ontologías existentes, especificación formal de la ontología, población de la ontología con instancias y finalmente, la evaluación y documentación de la ontología.



Figura 2.2: Las seis etapas definidas en la metodología *Knowledge Engineering Methodology* [124].

Aunque en la actualidad no hay ninguna metodología que sea adoptada como estándar [49], existen diversas propuestas. Algunas están especializadas en ciertos aspectos como la adquisición, la validación [182] o la reutilización [241], aunque la gran mayoría se enfocan en la creación [71]. Las actividades a desarrollar difieren entre las diversas propuestas, sin embargo, la gran mayoría de ellas incluye las actividades que se describen a continuación.

El proceso de creación de una ontología ilustrado en la Figura 1.3, comienza con una fase llamada de abstracción del dominio del problema, para poder determinar aspectos básicos como el uso, análisis y alcance del dominio que cubrirá la ontología, los tipos de preguntas que debe contestar y quién usará y será el responsable de mantener la ontología [224].

Después de la fase de abstracción del problema, se desarrolla una fase de implementación, que comienza con la identificación de los términos relevantes que deben aparecer en la ontología, que se transforman en conceptos y, posteriormente, en clases. Un aspecto crucial durante esta fase es la definición de la taxonomía de la ontología. Esta taxonomía generalmente se desarrolla mediante un enfoque descendente, en el que se comienza con los conceptos más generales y se van definiendo los conceptos más específicos.

Una vez definida la taxonomía del modelo ontológico, la siguiente fase es la definición de las propiedades intrínsecas de las clases, así como las relaciones

entre ellas. Estas relaciones especifican la forma en la que las diversas clases interactúan entre sí, mediante el establecimiento de factores de dependencia.

Finalmente, una vez que se han definido claramente tanto la taxonomía de conceptos como las relaciones entre estos, la parte final consiste en la creación de instancias de clases, que representan objetos de alguna clase con valores específicos en las propiedades, que distinguen a cada elemento de los demás.

Como se ha comentado previamente, hay una falta de herramientas que soporten por completo los procesos en las metodologías propuestas. Únicamente Methontology [71] y On-To-Knowledge [68] tienen algún tipo de herramienta de soporte a las tareas dentro de sus respectivos procesos. Ontometric [187] tiene una implementación de un servidor web para la evaluación de las ontologías, pero no para soporte a procesos de creación o gestión de ontologías.

Methontology [110] es una de las primeras y más importantes metodologías, e incluye una herramienta llamada WebODE, que soporta las tareas definidas para los procesos. Principalmente se enfoca en el desarrollo de ontologías a nivel de conocimiento, y propone un acercamiento relacionado con el tradicional proceso en cascada para el desarrollo de *software* [230], aplicado al ciclo de vida del desarrollo de una ontología, definido en [72]. Esta metodología tiene sus raíces en las principales actividades identificadas en el proceso de desarrollo de *software* y metodologías de la Ingeniería del Conocimiento. Incluye la identificación del proceso de desarrollo de ontologías, así como un ciclo de vida basado en prototipos evolutivos y diversas técnicas particulares para llevar a cabo cada actividad. En cada prototipo, en cada iteración, se pueden agregar, cambiar o eliminar términos, propiedades, etc. Para cada prototipo, esta metodología propone iniciar con la especificación y, simultáneamente con esta fase, inicia la actividad de adquisición del conocimiento.

Methontology define básicamente cuatro pasos para la construcción de ontologías. El primer paso es la especificación del alcance y la granularidad; el segundo paso consiste en la conceptualización, fase en la que incluso pueden generarse diagramas tipo UML (*Unified Modeling Language*)⁴ [173, 55]. La tercera fase consiste en la implementación de la ontología, en la cual se pasa de la conceptualización a la especificación de la ontología en un lenguaje como RDF (*Resource Description Framework*)⁵ u OWL (*Web Ontology Language*)⁶, para finalmente terminar con la evaluación de la ontología.

Methontology se basa en el modelo iterativo de *Ingeniería de Software* [142], en el que se desarrollan diversos prototipos o versiones, hasta lograr

⁴<http://www.uml.org/>

⁵<http://www.w3.org/RDF/>

⁶<http://www.w3.org/TR/owl-features/>

completar el modelo ontológico. Otro aspecto que considera esta metodología es la evaluación del conocimiento, y los errores más comunes en el momento de implementar los modelos ontológicos [106]. Arpírez et al. [15] han propuesto WebODE, un marco de trabajo para desarrollar Ingeniería de Ontologías que se basa en el uso de Methontology, y soporta las actividades propuestas en esta.

On-To-Knowledge [68] es un proyecto europeo sobre una herramienta de administración de conocimiento basada en contenidos. Como resultado de la investigación, se propone una metodología para aplicaciones de administración de conocimiento basadas en ontologías. La metodología propone lo que llama bloques de construcción y sus relaciones, para generar y mantener soluciones de administración de conocimiento basadas en ontologías, con un enfoque especial en los procesos y metaprosesos del conocimiento. La metodología On-To-Knowledge se centra en la identificación, ilustración e instanciación del proceso y del metaproseso del conocimiento. Los bloques de construcción están formados por aspectos relevantes de administración del conocimiento y por diversas ontologías. Esta metodología está asociada a la herramienta OntoEdit, que soporta las tareas definidas en On-To-Knowledge.

La propuesta DOGMA (*Development of Ontology Guided Methodology Approach*) es un marco de trabajo para desarrollar la Ingeniería de Ontologías muy formalmente [146, 241]. La filosofía fundamental en la metodología DOGMA es la reutilización de las ontologías, ya que se consideran como recursos escalables y compartidos que permiten reutilizar el conocimiento [148]. La reutilización de las ontologías se propone en base a la independencia de las aplicaciones, con las ontologías definidas en un nivel conceptual, o dicho de otra forma, un nivel de abstracción alto. Esta independencia del dominio de aplicación, se logra con un concepto denominado *ontología base*, que representa el nivel más abstracto posible sobre un dominio, lo que la metodología denomina como *axiomatización del dominio*. Esta ontología base es muy genérica y define conceptos generales a un nivel de abstracción alto, que pueden aplicarse en diversos dominios, y que a partir de esta, pueden generarse ontologías más específicas para ciertas aplicaciones. Todo esto, con la finalidad de hacer las ontologías mas reutilizables entre los diferentes dominios de aplicación que puedan compartir estos modelos.

El proyecto DOGMA incluye la definición de un lenguaje de marcado conceptual llamado ORM-ML [144], que permite representar diagramas conceptuales ORM (*Object Role Modeling*) [125, 126], para reutilizar modelos conceptuales de datos basados en este lenguaje. También cuenta con una herramienta prototipo que soporta las actividades definidas, denominada DogmaModeler⁷

⁷<http://www.jarrar.info/Dogmamodeler/index.htm>

[143]. DogmaModeler [145] es una herramienta de modelado de ontologías bajo el lenguaje propietario ORM-ML. El uso de la herramienta DogmaModeler para modelar ontologías, presenta el inconveniente del lenguaje propietario ORM-ML, el cual tiene que ser traducido a otros lenguajes como OWL [134]. Sin embargo, la filosofía detrás de la metodología DOGMA representa una buena opción para generar diseños altamente reutilizables.

Sugumaran y Storey [245] proponen una arquitectura para un sistema de administración de ontologías, así como una metodología para la creación de modelos ontológicos. Esta metodología comienza con la identificación de todos los conceptos básicos del dominio de aplicación, después la identificación de las relaciones, para después finalmente, definir las restricciones. La arquitectura contiene un repositorio de ontologías para fomentar la reutilización. La plataforma ha sido implementada, aunque no contiene funcionalidad para el desarrollo de los procesos de creación, gestión, mantenimiento y reutilización de ontologías.

Las ontologías requieren metodologías, pero también otro tipo de herramientas y servicios que garanticen su calidad. Por ejemplo, se necesita revisar si todos los conceptos son coherentes y que las clases puedan tener instancias (*meaningfulness*). También se requiere revisar que sean correctas, es decir que los conceptos estén correctamente definidos (*correctness*). De igual forma, se necesita revisar que sean mínimamente redundantes (*minimally redundant*). Algunas metodologías han sido propuestas específicamente enfocadas a la evaluación de modelos ontológicos.

Para evaluar si una ontología es correcta o no, Vrandeic y Sure [271] proponen realizar un proceso de normalización sobre su estructura, para corregir la falta de valores sobre los elementos de la ontología. Este proceso consiste en cinco pasos: el primero implica nombrar a todas las clases anónimas (clases que no tienen un nombre definido), nombrar a los individuos anónimos, clasificar jerárquicamente y unificar los nombres, propagar todas las instancias en las clases más profundas en la jerarquía y, finalmente, normalizar las propiedades objeto. Este proceso de normalización está orientado a mejorar el comportamiento de las ontologías, así como corregir algunos posibles errores.

Por otro lado, OntoClean es otra metodología para la evaluación de las ontologías, cuya filosofía propone asignar algunas metapropiedades a las entidades de la ontología, para proveerlas con un significado lógico y semántico [120, 121]. Esas metapropiedades adoptadas de la filosofía, son: la esencia, la rigidez, la unidad, la identidad y la dependencia. Una propiedad de una entidad es esencial a la entidad, si y sólo si esa propiedad es verdadera en la entidad para cada mundo posible. Por otro lado, se dice que una propiedad

es rígida si se cumple que es esencial para todas sus posibles instancias; una instancia de una propiedad rígida no puede dejar de ser una instancia de esa propiedad en un mundo diferente. La identidad se refiere al problema de la capacidad de reconocer entidades individuales en el mundo, que son las mismas. Finalmente, la unidad se refiere a la capacidad de reconocer a todas las partes que forman una entrada individual. Todos estos conceptos y su aplicación en la metodología permiten analizar y detectar relaciones que no son lógicamente consistentes.

Cuando se requiere incorporar una ontología existente a un proyecto nuevo, se tiene que evaluar de alguna forma, que la ontología candidata sea idónea. Lozano-Tello y Gómez-Pérez [187] han propuesto Ontometric, un marco de trabajo para medir la idoneidad de las ontologías, principalmente en los procesos de reutilización. Esta propuesta se basa en la definición de una taxonomía de ciento sesenta características, llamada *multilevel framework of characteristics*. Este marco de trabajo provee los aspectos principales para poder escoger y comparar las ontologías existentes, además que se utiliza como base para la construcción de nuevos modelos, principalmente, bajo el principio de la reutilización. Ontometric cuenta con una implementación de una interfaz web para el desarrollo de la evaluación.

<i>Metodología</i>	<i>Soporte</i>	<i>Reutilización</i>	<i>Ciclo de vida</i>	<i>Evaluación</i>
Ushold	no	sí	no	no
Methontology	WebODE	sí	sí	sí
OnToKnowledge	OntoEdit	sí	sí	no
OntoClean	no	no	no	sí
Vrandecic	no	no	no	sí
Sugumaran	sí	sí	no	no
Ohgren	no	sí	sí	no
DOGMA	DogmaModeler	sí	no	no
Ontometric	Ontometric	sí	no	sí

Tabla: 2.1: Resumen de las principales metodologías propuestas, donde se muestran los aspectos más importantes: a) si tienen algún tipo de soporte por parte de alguna herramienta; b) si fomentan el principio de reutilización; c) si abarcan el ciclo de vida completo de las ontologías y d) si están enfocadas en la evaluación de las ontologías.

La Tabla 2.1 resume las metodologías descritas, en donde se muestra que hay pocas metodologías que tienen una herramienta que las soporte, ya que únicamente Methontology, OnToKnowledge, DOGMA y Ontometric, tienen alguna implementación que las soporte.

Sea cual sea la metodología utilizada, lo cierto es que se requiere un lenguaje de implementación de las ontologías. La evolución de los lenguajes para la implementación de ontologías comienza en la década de los años noventa. La sección 2.3.2 hace una recopilación de la evolución de los lenguajes para la representación de ontologías.

2.3.2. Representación de ontologías

El comienzo del paradigma para la representación del conocimiento, incluía lenguajes para la representación de ontologías basados en lógica de primer orden como CycL [181], Ontolingua [66], OCML [201], OKBC [42] y FLogic [110].

CycL es un lenguaje para la representación de conocimiento que utiliza una sintaxis propia. Este lenguaje declarativo, se basa en el uso de predicados de primer orden e incluye un motor de inferencia. Surge como parte del proyecto Cyc, una gran base de conocimiento que se ha utilizada en diversas áreas, como en el procesamiento del lenguaje natural [21] y en la recuperación de textos [229], entre otras.

Ontolingua es un lenguaje con sintaxis propia, sumamente expresivo, que permite la representación de conceptos, taxonomías de conceptos, relaciones entre estos, funciones, axiomas, instancias y procedimientos. El proyecto incluye un servidor también denominado Ontolingua, que contiene una librería de ontologías escrita en el propio lenguaje. Tanto el servidor Ontolingua como el proyecto Cyc, representan dos propuestas con características muy similares. Ambas propuestas implementan lenguajes propios (Ontolingua y CycL), así como un conjunto de ontologías públicamente disponibles⁸, para la construcción de nuevos modelos ontológicos, principalmente a través del concepto de reutilización.

Uno de los lenguajes más tradicionales es *OCML (Operational Conceptual Modelling Language)*⁹, un lenguaje de modelado conceptual, que surge como una evolución de Ontolingua. Utiliza reglas de producción y proporciona clasificaciones automáticas que soportan la construcción de modelos de conocimiento a través de diversos tipos de conceptos. *OCML* se enfoca en la lógica en lugar de la implementación, de esta forma provee mecanismos que permiten la especificación y operación de funciones, relaciones, clases, instancias y reglas en lugar de matrices o tablas de *hash*. También incluye mecanismos para definir ontologías y métodos para la solución de problemas que representan las

⁸ *OpenCyc* es la versión con código abierto de la tecnología Cyc.

⁹ <http://kmi.open.ac.uk/projects/ocml/>

principales tecnologías desarrolladas en el área de modelado del conocimiento. *OCML* tiene soporte a través de una serie de bibliotecas de modelos que son reutilizables, lo que representa una fuente de recursos para la comunidad de modelado del conocimiento.

Por otro lado, también surgió OKBC (*Open Knowledge Base Connectivity*), un protocolo que permite acceder a las bases de conocimiento almacenadas en diferentes sistemas de representación de conocimiento, principalmente para fomentar la reutilización.

Después, el surgimiento de Internet propició la aparición de lenguajes formales, que explotan las características de la Web (*web-based ontology languages*). Estos lenguajes tienen una sintaxis basada en HTML y en XML, y, su propósito principal no es el desarrollo de ontologías, sino la presentación y el intercambio de datos [110]. Dentro de esta categoría de lenguajes destacan SHOE [188, 130], XOL [156] y RDF¹⁰, como los primeros lenguajes de “marcado”[109].

SHOE se basa en el uso de etiquetas similares a las de HTML. Permite la representación de conceptos, taxonomías, relaciones, instancias y reglas para realizar deducciones lógicas. Después del surgimiento de XML, SHOE fué modificado para utilizar XML en lugar de HTML.

XOL es un lenguaje creado para permitir el intercambio de definiciones ontológicas entre diversas partes. Originalmente surgió como una evolución del protocolo OKBC, utiliza XML como el lenguaje para la definición de la sintaxis, y permite codificar información del esquema de la ontología (clases, taxonomía y propiedades).

Con el surgimiento de la Web y los lenguajes de marcado como HTML, los lenguajes para la representación de ontologías evolucionaron en este sentido. Uno de estos lenguajes es RDF, que surge a finales de los años noventa como un modelo estándar para el intercambio de datos en la Web, desarrollado por el consorcio W3C. RDF es un lenguaje de propósito general para representar información en la Web. Se basa en el uso de “tripletas”, formadas por dos conceptos y una relación entre estos dos conceptos (sujeto, predicado y el objeto). Esto permite la definición de datos estructurados y semi-estructurados, que se publican e intercambian entre diferentes aplicaciones.

RDF evoluciona posteriormente a una extensión denominada RDF-Schema, también desarrollado por el consorcio W3C. RDF-Schema¹¹, que añade una jerarquía a la definición anterior de RDF, introduce el concepto de clase para

¹⁰<http://www.w3.org/RDF/>

¹¹<http://www.w3.org/TR/rdf-schema/>

sustituir al de recurso, y agrega una jerarquía y restricciones sobre las propiedades. XML se utiliza como el lenguaje para describir la parte sintáctica, mientras que RDF se usa para definir las propiedades semánticas.

Después, y como parte de la misma evolución, surgen otras dos propuestas de nuevos lenguajes desarrollados como extensiones de RDF-Schema: por un lado, el lenguaje *OIL* [69, 70], desarrollado por un grupo de investigadores europeos, algunos de ellos colaboradores en el proyecto On-To-Knowledge [68]; y, por otro lado, el lenguaje *DAML-ONT* [195], desarrollado por investigadores en Estados Unidos, como parte del programa DARPA DAML.

Una vez creados estos dos lenguajes separadamente, un esfuerzo por unificarlos resultó en el lenguaje denominado DAML+OIL [194]. Este lenguaje surge como una capa superior a RDF-Schema, utiliza la misma sintaxis, y las mismas primitivas ontológicas: subclases, rango, dominio, cardinalidad, transitividad, etc. DAML+OIL es equivalente al lenguaje de lógica de descripción expresiva, que incluye una semántica bien definida, propiedades formales, algoritmos de razonamiento y formas simples de tipos de datos.

Posteriormente el consorcio W3C publica la recomendación de un nuevo lenguaje para la definición de ontologías denominado OWL. OWL surge como un estándar¹² en el año 2004, y actualmente se considera como la mejor opción para la definición de ontologías para la Web Semántica y los sistemas basados en conocimiento. Es compatible con la arquitectura de la World Wide Web en general y de la Web Semántica en particular. Entre las capacidades de OWL se pueden mencionar la posibilidad de poder distribuirse entre diversos sistemas, la escalabilidad a las necesidades web y la compatibilidad con los estándares web para accesibilidad e internacionalización.

OWL también está basado en RDF y RDF-Schema, pero agrega más funcionalidad y vocabulario para describir propiedades y clases; por ejemplo, incluye los conceptos de clases disjuntas o equivalentes, cardinalidad (número de posibles valores para un atributo), las clases enumeradas, y también enriquece los tipos de propiedades, ya que agrega características como simetría y transitividad. OWL tiene tres versiones de acuerdo a la expresividad lógica necesaria: OWL Lite, OWL DL y OWL Full¹³.

OWL Lite soporta la definición de una taxonomía ontológica, así como una serie de restricciones básicas, mientras que OWL DL soporta una mayor expresividad relacionada con la lógica descriptiva, enfocada a cubrir la integridad desde el punto de vista computacional. Esto implica que todos los procesos puedan desarrollarse computacionalmente y que esos procesos se realicen en

¹²<http://www.w3.org/2004/OWL/>

¹³<http://www.w3.org/TR/2004/REC-owl-guide-20040210/#OwlVarieties>

tiempos finitos. Finalmente, la tercera versión OWL Full, ofrece la máxima expresividad posible y toda la libertad sintáctica de RDF, pero sin ninguna garantía computacional de integridad. En resumen, cada sublenguaje es una extensión de su predecesor, tanto en lo que puede ser legalmente expresado como en lo que puede ser válidamente concluido.

Diversos artículos [51, 109, 122] hacen una comparación y recopilan el proceso evolutivo de los lenguajes para ontologías. La Tabla 2.2 muestra la clasificación más general de los lenguajes para representar ontologías, de acuerdo a [110, 111].

La clasificación divide en tres tipos los lenguajes: el primer tipo son los lenguajes tradicionales que surgen para la representación del conocimiento, y que permiten realizar deducciones lógicas e inferencia; el segundo tipo lo representan los estándares web, y por último, el tercer tipo que clasifica a todos aquellos lenguajes que se basan en el uso de la Web, a diferencia de los tradicionales que básicamente son para aplicaciones de escritorio.

SPARQL es una recomendación del consorcio W3C como un lenguaje de consultas sobre RDF ¹⁴. Es un lenguaje que no se enfoca en la definición del esquema de las ontologías, sino que es un lenguaje de consulta sobre la base de conocimiento. Está basado en el clásico lenguaje SQL para consultas sobre bases de datos, pero se define sobre las bases de conocimiento.

Tradicionales	Recomendaciones	Basados en Web
Ontolingua	XML	XOL
OKBC	RDF	SHOE
OCML	RDF-Schema	OIL
FLogic	OWL	DAML
LOOM		

Tabla: 2.2: Clasificación más general de los lenguajes para la representación de ontologías. Se definen tres grandes áreas, por un lado los lenguajes tradicionales, por otro lado los estándares web y las recomendaciones y, finalmente, los lenguajes basados en la Web.

2.3.3. Métricas ontológicas

El tercero de los elementos de la Ingeniería de Ontologías, lo representan las métricas sobre los modelos ontológicos. Las métricas tienen como principal objetivo, permitir la evaluación de ciertos aspectos en las ontologías [64], como

¹⁴<http://www.w3.org/TR/rdf-sparql-query/>

el acoplamiento, la cohesión o simplemente las métricas de esquema, enfocadas al número de clases, propiedades, instancias, etc.

Sin lugar a dudas, OntoQA es la principal propuesta en lo que se refiere a métricas ontológicas [253, 254]. Esta propuesta comprende algunas métricas basadas en el esquema de las ontologías, para medir la riqueza de las relaciones, los atributos y la herencia. Otras categorías están enfocadas en las clases y miden aspectos como la cohesión, la conectividad, la herencia o la riqueza de una clase, y finalmente, otra categoría se enfoca en las propiedades.

OntoQA establece tres categorías principales de métricas: las métricas del esquema, las métricas de las instancias y de las clases. El primero de los grupos, está formado por diversas métricas que permiten medir la riqueza de las relaciones, los atributos y la herencia del esquema de la ontología. Estas métricas son la riqueza de las relaciones de una clase, la importancia y plenitud de una clase, la herencia, así como la conectividad y la legibilidad.

Por otro lado, la segunda categoría de métricas evalúa la colocación de las instancias de datos dentro de la ontología, y también permite evaluar el uso efectivo de la ontología, para representar el conocimiento modelado. El promedio de población es la métrica principal en este grupo, y calcula la distribución promedio de instancias en todas las clases, mientras que la cohesión se refiere al grado de separación entre las instancias, dadas las relaciones entre estas. La métrica del tercer grupo es la riqueza de las clases, que indica el porcentaje de clases con instancias en la base de conocimiento, con respecto a las clases definidas en la ontología.

Orme et al. [215] proponen un conjunto de métricas de acoplamiento para sistemas basados en ontologías OWL. Estas métricas son: el *número de clases externas* (NEC), las *referencias a las clases externas* (REC) y la métrica de las *referencias incluídas* (RI). Este trabajo define una nueva propuesta con un nuevo tipo de medición de acoplamiento para el desarrollo de sistemas, que se basa en los datos de la ontología y su estructura.

La primera métrica (NEC), representa el número de clases externas distintas, que han sido definidas fuera de la ontología, pero que se utilizan para definir nuevas clases y propiedades en la ontología. La segunda métrica (REC), se refiere al número de referencias a clases externas en la ontología. A diferencia de NEC, que es una medida directa del número de clases en la ontología, REC es una medida directa del número de *fanouts* (diferentes jerarquías de clases con raíces externas). Finalmente, RI es una medida directa del número de elementos incluídos en la ontología.

Otra propuesta, complementaria a las anteriores, es un conjunto de métricas

de cohesión para medir las relaciones modulares de las ontologías OWL [281]. Estas métricas son el *número de clases raíz* (NoR), el *número de clases hoja* (NoL) y la *profundidad promedio de los nodos hoja en los árboles de herencia* (ADIT-LN). La métrica NoR se define como el número total de clases raíz, explícitamente definidas en la ontología. La métrica NoL se define como el número de clases hoja explícitamente definidas en la ontología. Finalmente, la métrica ADIT-LN se define como la suma de las profundidades de todos los caminos, dividido por el número total de caminos en la ontología.

Yinglong et al. [282] proponen un conjunto de métricas de cohesión que miden la relación modular de las ontologías en el contexto cambiante de la Web. Estas métricas se han definido tomando en cuenta el principio de cohesión del paradigma orientado a objetos, pero adaptado a las ontologías. Los autores se concentran en medir inconsistencias en las ontologías y considerar la semántica ontológica en lugar de la estructura. Estas métricas son el *número de particiones de ontologías* (NOP), el *número de subconjuntos mínimamente inconsistentes* (NMIS) y el *valor promedio de inconsistencias* (AVAI). La métrica NOP se define como el número de particiones semánticas de una base de conocimiento. La métrica NMIS se define como el número de subconjuntos mínimamente inconsistentes en una base de conocimiento. Finalmente, la métrica AVAI se define como el radio de la suma de los valores del impacto de las inconsistencias de todos los axiomas y aserciones a la cardinalidad de la base de conocimiento.

*Ontology Metrics*¹⁵ es una herramienta que calcula diversas métricas de esquema de las ontologías. Básicamente es una plataforma web que valida el código de las ontologías OWL en línea, y despliega estadísticas de la ontología así como la expresividad del lenguaje utilizado (OWL DL, OWL LITE u OWL FULL). Esta herramienta calcula principalmente contadores que indican el total de clases, el total de propiedades, de instancias y axiomas lógicos. Una vista de esta herramienta se muestra en la Figura 2.3 (b).

Yang et al. [284] proponen un conjunto de métricas definidas desde un punto de vista diferente, basadas en la evolución de las ontologías. Estas métricas, examinan principalmente la cantidad, el radio y la correlatividad de los conceptos y las relaciones, para evaluar las ontologías desde el punto de vista de la complejidad y su evolución. Estas métricas se dividen en dos grandes grupos: las métricas primitivas y las métricas de complejidad. Las métricas primitivas incluyen el *número total de conceptos o clases* (TNOC), el *número total de relaciones* (TNOR) y el *número total de caminos* (TNOP), en donde un camino puede tomarse desde un concepto específico en particular, hasta el concepto más general en la ontología. La primera de las métricas de complejidad defini-

¹⁵<http://owl.cs.manchester.ac.uk/metrics/>

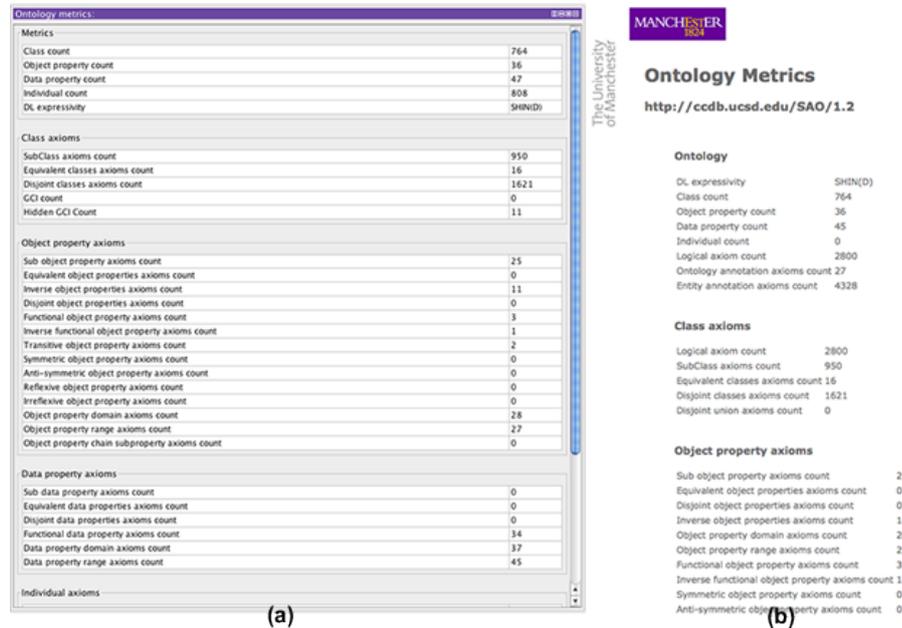


Figura 2.3: (a) Muestra la representación de las métricas con Protégé, mientras que la figura (b) muestra las métricas con la herramienta *Ontology Metrics*. Ambas herramientas se limitan a desplegar valores numéricos globales a todo el modelo ontológico, por ejemplo el número total de clases, de propiedades, de instancias, etc.

da, es el *número de relaciones promedio por concepto*, que se calcula dividiendo el valor de TNOR por TNOC. La segunda de las métricas de complejidad es el *número de caminos promedio por concepto*, que se calcula dividiendo TNOP por TNOC.

Algunas de las herramientas para el desarrollo de ontologías incluyen métricas basadas en el esquema, básicamente contadores (valores totales) de subclases, propiedades, instancias, axiomas, etc. La Figura 2.3 (a) muestra una representación de métricas en Protégé y la Figura 2.3 (b) muestra las métricas con la herramienta *Ontology Metrics*¹⁶, una herramienta en línea que despliega un conjunto de métricas sobre un modelo ontológico. Ambas representaciones tienen en común que se limitan a desplegar contadores generales de toda la ontología.

La tabla 2.3 establece un cuadro comparativo de las métricas descritas en esta sección.

¹⁶<http://owl.cs.manchester.ac.uk/metrics/>

Métrica	Estructura	Clasifica	Cohesión	Acoplamiento
OntoQA	estructura	sí	sí	no
Orme	estructura	no	sí	sí
Yinglong	semántica	no	sí	no
Ontology Metrics	estructura	no	no	no
Yang	estructura	no	no	no

Tabla: 2.3: Resumen comparativo de las métricas ontológicas descritas. Se indica si las métricas se basan en la estructura o si son semánticas, si permiten clasificar a las ontologías, si calculan valores de cohesión y acoplamiento.

2.3.4. Herramientas para la gestión de ontologías

Aparte de las metodologías, los lenguajes y las métricas, otra categoría importante para el desarrollo de la Ingeniería de Ontologías, corresponde a las herramientas de desarrollo. Existen diversas herramientas enfocadas a la creación, edición, visualización y mantenimiento de ontologías [251].

Hoy en día, con el poder de cómputo que está disponible para los usuarios, y con las novedosas propuestas en diversas áreas de investigación, las herramientas para el desarrollo de ontologías pueden realmente ser bastante robustas y permitir a los usuarios desarrollar de forma eficiente los procesos de la Ingeniería de Ontologías.

Algunas de estas son comerciales, aunque generalmente ofrecen versiones libres con una funcionalidad reducida o simplemente permiten su uso de forma libre durante un cierto periodo de prueba. Entre estas herramientas comerciales destacan SemanticWorks¹⁷, OntoStudio¹⁸, TopBraidComposer¹⁹, o IODT²⁰, aunque esta última no tiene soporte ni actualizaciones desde hace varios años. Las principales características de este primer grupo de herramientas, se describen en los siguientes párrafos.

La primera de estas herramientas es SemanticWorks, que se muestra en la Figura 2.4. Es una herramienta comercial diseñada para editar documentos RDF y crear ontologías OWL mediante el uso de una vista gráfica de diseño. La herramienta permite revisar la sintaxis y la semántica de las ontologías, gracias a un modelador gráfico basado en el enfoque de mapas conceptuales. Sin embargo, su principal desventaja radica en la redundancia de información, ya que como se muestra en la Figura 2.4, esta solución es poco eficiente, ya

¹⁷<http://www.altova.com/semanticworks.html>

¹⁸<http://www.ontoprise.de/en/products/ontostudio/>

¹⁹<http://www.topquadrant.com/index.html>

²⁰<http://www.alphaworks.ibm.com/tech/semanticstk>

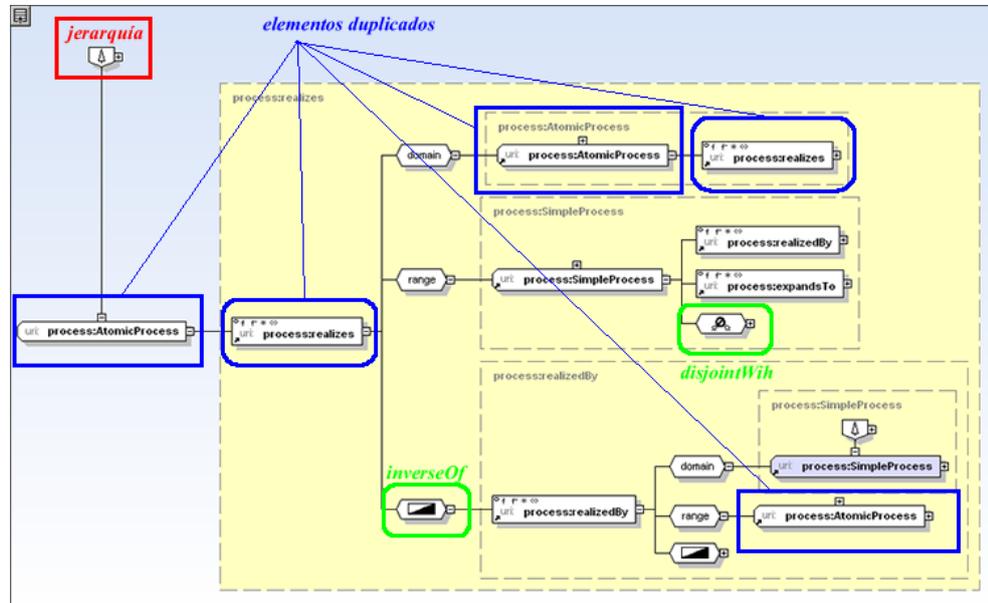


Figura 2.4: Una vista de la herramienta Semantic Works. Utiliza representaciones basadas en mapas conceptuales. Define sus propia simbología para la representación de las propiedades, los conceptos, la jerarquía, la unión de clases, etc.

que duplica muchos de los elementos visuales mostrados. Esta duplicidad de elementos provoca, en primer lugar, visualizaciones saturadas de información y, en segundo lugar, que el usuario pierda fácilmente el flujo de navegación a través de la herramienta.

Otra de estas herramientas comerciales es OntoStudio, una opción bastante completa para modelar ontologías [276, 277]. Básicamente cubre los procesos de creación y mantenimiento, mediante el uso de una herramienta de mapeo gráfico. También proporciona un soporte bastante robusto de conexiones con bases de datos, y funcionalidad de inferencia.

TopBraid Composer, que se ilustra en la Figura 2.5, es otra herramienta comercial. Básicamente, es un ambiente de desarrollo gráfico para modelar datos, ontologías y en general una gama de aplicaciones para el desarrollo de la Web Semántica. También cuenta con capacidad de realizar inferencia, tiene soporte para SPARQL y múltiples fuentes de datos y lenguajes como OWL, RDF, bases de datos, archivos UML creados con IBM Rational²¹, archivos de esquema XML y hojas de cálculo. Es una herramienta muy completa para el

²¹<http://www.ibm.com/developerworks/rational/>

desarrollo de la Ingeniería de Ontologías y cuenta con una versión libre con funcionalidad reducida.

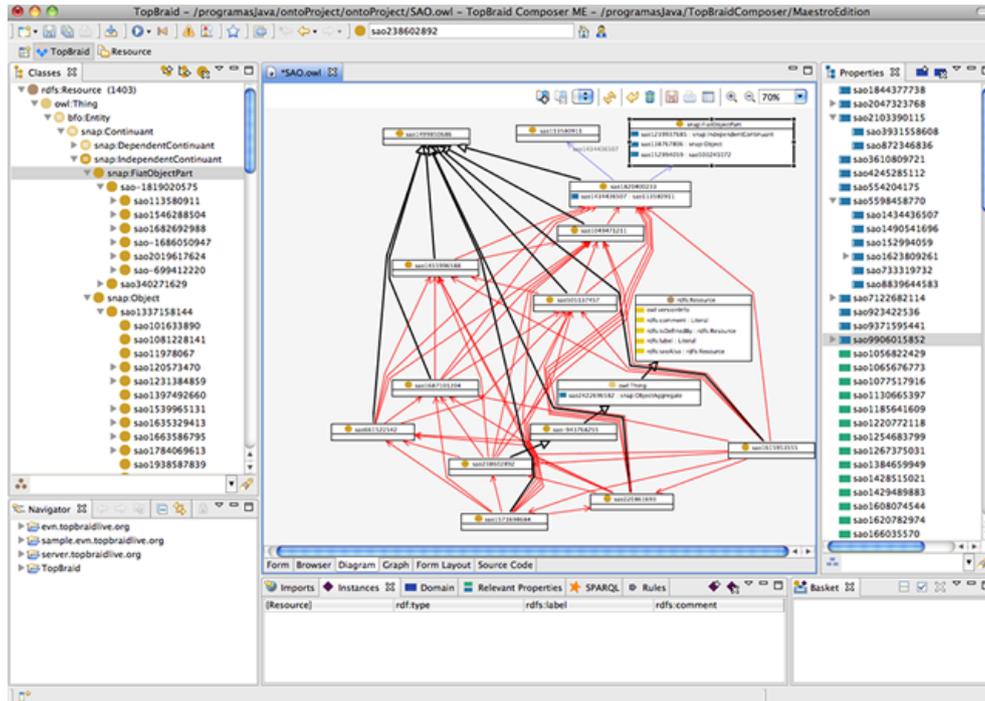


Figura 2.5: Una vista de la herramienta Top Braid Composer. Utiliza el mismo diseño de interfaz gráfica de usuario que Protégé. Se muestra del lado izquierdo la jerarquía de clases, al centro las relaciones entre las clases, y del lado derecho se muestran las propiedades objeto y de datos.

IBM también ha propuesto una herramienta denominada IODT (*Integrated Ontology Development Toolkit*), que representa un metamodelo para la definición de nuevas ontologías llamado EODM y un repositorio para almacenarlas llamado SOR. EODM²² es un proyecto de código abierto basado en Eclipse, para el desarrollo de la Ingeniería de Ontologías, administración y visualización. El entorno de modelado está basado en IBM Rational para el desarrollo de aplicaciones de *software*, con lo que, las visualizaciones del esquema ontológico están completamente basadas en una representación UML.

De este primer grupo de herramientas comerciales, destacan tres enfoques principalmente. El primero de ellos es el uso de mapas conceptuales para la visualización de modelos ontológicos, propuesto por Semantic Works, con la desventaja previamente comentada. El segundo enfoque es el uso de la cono-

²²<http://www.eclipse.org/modeling/mdt/?project=eodm\#eodm>

cida técnica visual de grafos dirigidos [128] para la representación de modelos, una solución común en la teoría de grafos [33, 278]. Esta solución de grafos dirigidos lo utilizan tanto TopBraid Composer como OntoStudio, prácticamente de una forma idéntica. Finalmente, el tercer enfoque se basa en el uso de representaciones basadas en los diagramas de clase UML para modelar ontologías. Esta solución la utilizan tanto TopBraid Composer como IODT.

Un segundo grupo de herramientas en esta clasificación, cubre todas aquellas que son de libre distribución. En este grupo de herramientas, destacan básicamente dos tipos de propuestas. La primera abarca todas aquellas que se basan en el uso del navegador web, entre las que se encuentran Ontolingua [66], OwlSight [239], WebODE [15], SWOOP [273] y HOZO [247, 176]. El resto de las herramientas se ejecutan como cualquier otra aplicación de escritorio, aunque pueden estar dotadas de capacidades para interactuar a través de la web. Entre estas herramientas destacan principalmente NeOn Toolkit [123], OntoEdit [249], KAON [36], Protégé [103], OntoSphere [35] y OntoTrack [183, 184]. En los párrafos siguientes se describen ambos grupos de herramientas.

Ontolingua²³ es una de las herramientas que se basan en el uso del navegador web. Se puede describir como un proyecto que proporciona un ambiente colaborativo distribuido, para crear, editar y modificar ontologías, disponible desde un entorno web. El servidor Ontolingua contiene una librería de ontologías compartidas por los usuarios, que fomenta muy bien el principio de reutilización de ontologías. Es un proyecto que se inició varios años atrás, limitado en sus capacidades y que ha quedado sin actualizaciones recientes.

Otra herramienta similar a Ontolingua es OwlSight²⁴, un navegador de ontologías OWL que se ejecuta desde un navegador web. Permite cargar ontologías desde ubicaciones web, despliega sus elementos y, posiblemente, su mayor ventaja es el uso de un razonador OWL llamado Pellet [239]. No utiliza ningún tipo de visualizaciones ni permite la edición ni creación de ontologías, simplemente se limita a representar las ontologías y permitir la navegación entre sus elementos.

WebODE [15] es una herramienta para modelar conocimiento mediante el uso de ontologías. Es una plataforma completa para el desarrollo de la Ingeniería de Ontologías que provee una variedad de servicios relacionados. Además, cubre y da soporte a la mayoría de las actividades involucradas en el proceso de desarrollo definidas en Methontology, una metodología descrita en la sección 2.3.1. Las ontologías en WebODE se almacenan en una base de datos relacional y proporciona una librería de funciones bien definida, orientada a servicios, que

²³<http://www.ksl.stanford.edu/software/ontolingua/>

²⁴<http://pellet.owldl.com/ontology-browser/>

facilita la integración con otros sistemas.

SWOOP²⁵ es un proyecto de código abierto con desarrolladores en todo el mundo [153]. El diseño es una aplicación basada en texto, que permite la edición de las ontologías, pero sin ofrecer interacción visual. Contiene una barra de direcciones para introducir directamente la dirección URI de la ontología, soporta múltiples ontologías, y está basada en el paradigma modelo-vista-controlador (*MVC*) [80]. También utiliza un razonador para revisar errores e inconsistencias, tiene múltiples vistas de la ontología, de las entidades y la vista jerárquica.

Finalmente, HOZO [214] es un editor de ontologías que provee una interfaz gráfica que utiliza árboles de jerarquías y grafos. Se basa en la idea de definir roles específicos y contextos que permiten clasificar a estos roles, formados por un conjunto de clases con tareas similares. HOZO utiliza un lenguaje propio para la definición de los modelos ontológicos, pero permite exportarlos a otros lenguajes como RDF-(s) y OWL principalmente. Este editor tiene una interfaz gráfica basada en el uso de listas para representar la jerarquía, y de árboles para mostrar los elementos de la clase seleccionada. La visualización es sencilla, basada en texto, y el flujo de navegación de la herramienta se desarrolla mediante vínculos que relacionan a los conceptos entre sí, de forma similar a los hipervínculos en HTML.

Este grupo de herramientas que se basan en el uso del navegador web son principalmente editores, que proveen poca o nula funcionalidad de visualización e interacción con el usuario. Se basan principalmente en el uso de formas en páginas HTML, o aplicaciones que se ejecutan dentro del entorno del navegador web como la tecnología de las *applets* [32]. Prácticamente, la única representación visual la implementa el editor SWOOP, ya que provee una visualización basada en el uso de la técnica conocida como *crop circles* [221], que básicamente son circunferencias internas anidadas dentro de otras circunferencias, para representar una relación de propiedad o como en este caso, una relación jerárquica.

Por otro lado, la Unión Europea ha visto el potencial del desarrollo de la Web Semántica y ha financiado diversos proyectos de investigación en el tema. Posiblemente el proyecto NeOn²⁶ sea el más importante al respecto. Particularmente, el objetivo principal era el de mejorar la capacidad para administrar múltiples ontologías que existen en un contexto específico y que se crean colaborativamente, y que además, pueden llegar a ser altamente dinámicas y en constante evolución.

Como resultado de este proyecto de investigación han surgido diversas he-

²⁵<http://www.mindswap.org/2004/SWOOP/>

²⁶<http://www.neon-project.org/>

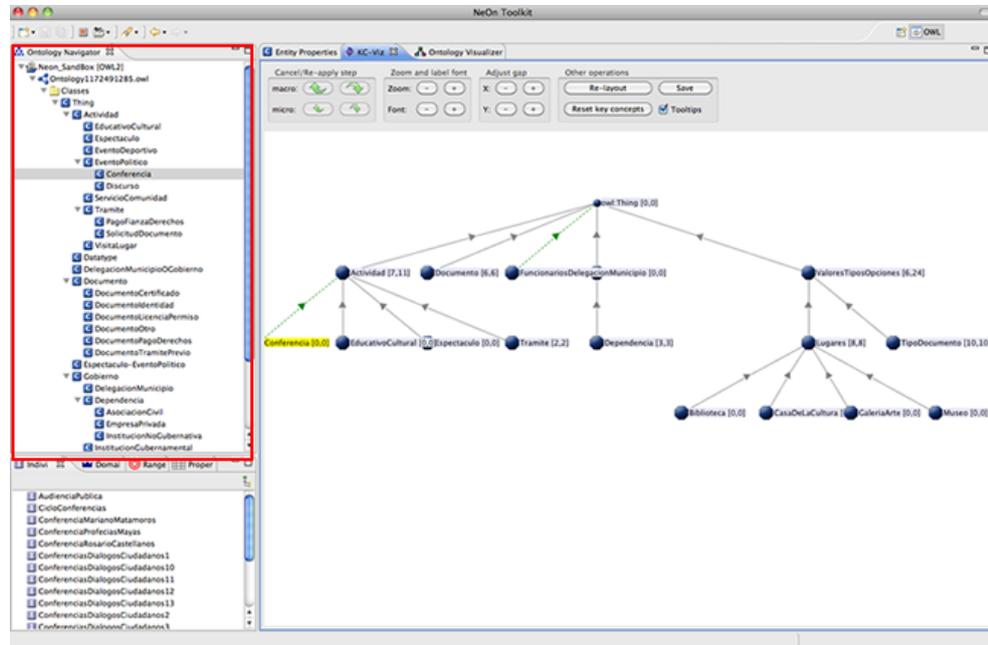


Figura 2.6: La herramienta Neon Toolkit muestra la jerarquía de los conceptos (en un recuadro en color rojo del lado izquierdo), mientras que del lado derecho despliega las visualizaciones de la ontología con las propiedades.

herramientas y aplicaciones basadas en el uso de ontologías [252], entre las que destacan una metodología [112], un servidor de alineación de ontologías y visualización [217] y, uno de los productos más importantes, un ambiente para el desarrollo de la Ingeniería de Ontologías llamado NeOn Toolkit [123]. Esta plataforma es bastante robusta y actualmente está abierta al desarrollo de *plugins* de Eclipse, que pueden integrarse al *toolkit*. La Figura 2.6 muestra una vista general del entorno de desarrollo NeOn Toolkit, resaltada en el recuadro de color rojo ubicado en la esquina superior izquierda, se representa la jerarquía de los conceptos. Al centro de la visualización, también se representa la jerarquía usando una vista diferente.

OntoEdit [249] es otro editor que se basa en una metodología para el desarrollo de ontologías, que hace uso de la inferencia. Básicamente, es un entorno flexible basado en una arquitectura de *plugins* que provee funcionalidad para editar ontologías. Se enfoca principalmente en tres aspectos principales del desarrollo, que son, la especificación de los requisitos, el refinamiento y la evaluación.

OntoEdit utiliza como motor de inferencia Ontobroker [67], una plataforma

que procesa fuentes de información en HTML, XML y RDF. Ontobroker incluye cuatro principales componentes: un motor de consultas, agentes de información, un motor de inferencia y un administrador de base de datos. El motor de consultas recibe las consultas y las contesta de acuerdo a los contenidos en su base de datos, que se actualizan gracias a los agentes que recolectan diversos contenidos de la Web. Otro componente crucial es el motor de inferencia, que utiliza las ontologías y el conocimiento almacenado para derivar nuevo conocimiento implícitamente; y finalmente, el administrador de la base de datos es el encargado de almacenar toda la información recopilada.

Karlsruhe Ontology and Semantic Web Tool Suite (KAON o también conocido como OntoServer) [36], surgió como la herramienta sucesora de OntoEdit [269]. KAON²⁷ es una plataforma con funcionalidad para el desarrollo de tecnologías de Web Semántica, principalmente en escenarios de comercio electrónico. La plataforma está dividida en tres capas: la capa superior es de aplicación y servicio a clientes, la capa intermedia se encarga de proveer acceso a las ontologías y, finalmente, la tercera capa se encarga de proporcionar acceso físico a los repositorios de datos como archivos de texto o bases de datos.

Sin duda alguna, como lo demuestra el análisis sobre el uso de herramientas de desarrollo que hicieron Khondoker y Mueller [170], la herramienta para edición de ontologías más ampliamente difundida entre la comunidad científica es Protégé [103], un ambiente para el desarrollo de sistemas basados en conocimiento que ha venido evolucionando desde hace más de una década. Comenzó como una pequeña aplicación diseñada para el dominio de la medicina, pero en la actualidad ha evolucionado hacia un conjunto de herramientas de propósito más general, de distribución libre y con código abierto.

El editor Protégé-OWL [172] permite a los usuarios cargar y guardar ontologías OWL y RDF, editar y visualizar clases, propiedades y reglas SWRL (*Semantic Web Rule Language*) [138, 137]. También permite definir características lógicas de las clases como expresiones OWL, ejecutar razonadores tales como los clasificadores de descripción lógica y la edición de individuos para el contexto de la Web Semántica. Así mismo, contiene funcionalidad para visualizar los modelos ontológicos [160, 2], con diversos *plugins* como Jambalaya [11, 185], GraphViz [60] y TGVizTab [3], que se muestra en la Figura 2.7.

OntoSphere [35] es una herramienta que utiliza tecnología tridimensional para la visualización de ontologías. Tanto la representación visual como la interacción, resultan muy intuitivas para el usuario. Esta herramienta no ofrece funcionalidad de edición ni modelado de ontologías, únicamente permite la representación visual.

²⁷<http://kaon.semanticweb.org/>

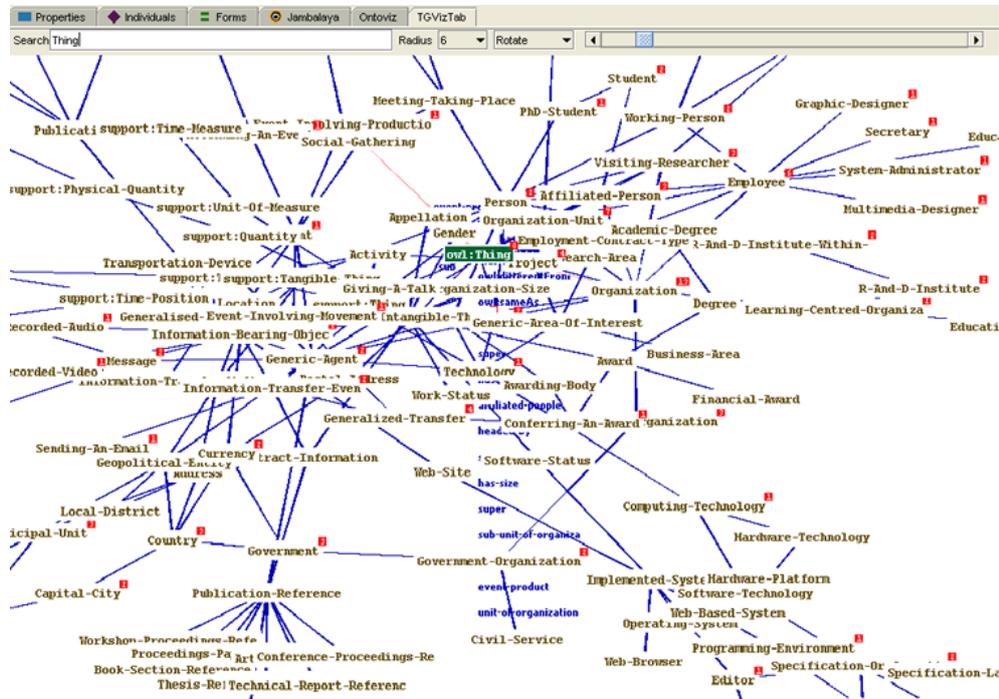


Figura 2.7: Una vista de la herramienta *TgVizTab* de Protégé. La visualización se enfoca en la representación de los conceptos y las relaciones entre estos.

Liebig y Noppens [183, 184] han propuesto *OntoTrack*, una herramienta para visualizar y editar ontologías “en una simple vista”, según describen estos autores. Esta propuesta muestra toda la ontología en una sola vista, en donde las relaciones tanto jerárquicas como de acoplamiento, se representan con líneas que representan aristas que unen a los nodos. Esta propuesta tiene funcionalidad para la edición de ontologías OWL Lite.

Existen algunas otras herramientas específicas para ciertos aspectos, por ejemplo, *OILED* [22] es una propuesta inicialmente desarrollada como un editor de ontologías OIL, para después evolucionar como un editor de ontologías DAML+OIL. *OILED* utiliza el motor de inferencia *FaCT* [139], para proveer características de clasificación y la revisión de consistencias. Otra herramienta de visualización de ontologías enfocada explícitamente en la taxonomía, es *TViz* [59]. Esta herramienta es muy específica, ya que únicamente cubre el aspecto referente a la visualización de taxonomías, y no tiene soporte para el resto de actividades involucradas en la Ingeniería de Ontologías.

Dmitrieva y Verbeek [57, 56] proponen algunos métodos de visualización de ontologías, bajo el esquema de una estructura de grafos con diferentes ge-

ometrías. Su propuesta combina vistas de grafos en dos y en tres dimensiones, con geometrías euclidianas²⁸, hiperbólicas²⁹ [166] y esféricas³⁰. Esta propuesta incluye técnicas de *zoom* semántico y árboles de jerarquías. Esta herramienta no posee funcionalidad para edición ni modelado de ontologías, sino que se limita exclusivamente a la visualización.

Algunas de las herramientas están provistas con funcionalidad específica para fomentar los procesos de reutilización de ontologías.

AKTiveRank es otra herramienta enfocada principalmente en los procesos de reutilización, que define ciertos mecanismos para clasificar ontologías relacionadas con algún concepto específico [4, 5]. La herramienta procesa como entradas diversas palabras clave introducidas por el usuario. Después, la herramienta busca las ontologías relacionadas con esos conceptos, para lo que utiliza el motor semántico Swoogle³¹, y una vez recuperadas esas ontologías, se clasifican y se listan en orden de relación con las palabras clave.

Una vez descritas las más importantes herramientas para desarrollar la Ingeniería de Ontologías, la tabla 2.4, hace un resumen comparativo de las principales características de las herramientas descritas en esta sección. Por otro lado, la sección 4.4 se enfoca en describir la problemática específica de estas herramientas.

²⁸Basadas en planos bi-dimensionales y espacios tridimensionales.

²⁹Basadas en planos denominados hiperbólicos, que se definen mediante el uso de cierto tipo de curvas.

³⁰Basadas en la superficie bi-dimensional de una esfera.

³¹<http://swoogle.umbc.edu/>

Herramienta	Visual	Capacidades	Distribución
SemanticWorks	sí	modelado con mapas conceptuales, consultas con SPARQL	comercial con una versión de prueba
OntoStudio	sí	editor basado en grafos con soporte para bases de datos y OWL, RIF, RDF(S) y ObjectLogic	comercial, con una versión de prueba
TopBraid Composer	sí	modelado con grafos y UML, consultas con SPARQL	comercial, con una versión libre
IODT	sí	modelado basado en UML	comercial sin actualizaciones recientes
Ontolingua	no	editor basado en web	libre
OwlSight	no	navegador basado en web	libre
WebODE	no	editor basado en web	no disponible
SWOOP	no	editor basado en texto, utiliza un razonador para la revisión de inconsistencias	libre
HOZO	no	editor basado en web, implementa una vista de árboles y listas jerárquicas	libre
NeOn Toolkit	sí	modelado basado en grafos	libre
OntoEdit	sí	editor de ontologías	no disponible
OntoTrack	sí	editor de ontologías OWL Lite	libre pero no actualizado
KAON	no	editor de ontologías	libre
Protégé	sí	edición de ontologías, consultas con SPARQL	libre
OntoSphere	sí	visualización tridimensional	libre
AKTiveRank	no	servidor de clasificación de ontologías	libre
Dmitrieva	sí	visualización de ontologías basada en grafos, con diferentes geometrías	no disponible

Tabla: 2.4: Resume las características más importantes de las principales herramientas para el desarrollo de la Ingeniería de Ontologías.

2.4. Conclusiones

Las ontologías son una de las representaciones más comunes de modelos de datos semánticos para la representación de conocimiento. En general, las ontologías se han utilizado en tres grandes áreas principales: la Ingeniería del Conocimiento, la Inteligencia Artificial y las Ciencias de la Computación. Entre las aplicaciones principales de los modelos ontológicos, destacan la Web Semántica y las aplicaciones de representación del conocimiento.

En términos generales, la estructura interna de una ontología comprende una taxonomía de conceptos que representan el dominio del problema, un conjunto de propiedades intrínsecas que definen las características de estos conceptos, un conjunto de relaciones entre los conceptos definido como acoplamiento y un conjunto de axiomas lógicos que permitan inferir cierto conocimiento a partir del modelo ontológico. Una base de conocimiento, es una ontología que ha sido poblada con instancias que representan valores específicos para los conceptos definidos.

Existe un área de investigación específica que se encarga de los procesos involucrados en la gestión del ciclo de vida de las ontologías. Este área denominada Ingeniería de Ontologías, define todos los flujos de trabajo y las actividades dentro de estos flujos, para gestionar el ciclo de vida de las ontologías. Los aspectos involucrados en esta gestión son principalmente las metodologías, los lenguajes para implementar las ontologías y las métricas, una estimación para evaluar ontologías.

Entre los procesos existentes en la Ingeniería de Ontologías para la gestión del ciclo de vida de los modelos ontológicos, sobresalen los procesos de creación, gestión, análisis y reutilización (mostrados en la Figura 1.1), aunque se considera que el proceso más importante es el proceso de creación, ya que permite construir un modelo ontológico prácticamente desde cero. A pesar de que las diversas metodologías definen diferentes tareas, hay dos grandes fases, con cinco tareas básicas definidas para el proceso de creación, que son comunes a la gran mayoría de todas ellas. Primero, una fase de abstracción del dominio del problema, que también incluye el planteamiento de la solución. Esta primera fase, es seguida por una segunda fase denominada de implementación, que comienza con la definición de la taxonomía de los conceptos, después la definición de las propiedades intrínsecas y las relaciones entre las clases, para finalizar con la definición de las instancias de las clases, que construyen la base de conocimiento.

La fase de abstracción del dominio del problema, involucra directamente al diseñador del modelo ontológico, que primero, tiene que entender el dominio

que se pretende modelar, y segundo, plantear la solución del modelo. Esta solución tiene que implementarse en un lenguaje formal procesable por los ordenadores (RDF, RDF-S, OWL, etc.). Esta fase de implementación, requiere el uso de herramientas que permitan al diseñador, desarrollar el proceso de modelado, o dicho de otra forma, la construcción del modelo de datos ontológico.

Finalmente, un aspecto importante en cuanto a las métricas, es que la gran mayoría de ellas se enfoca en la estructura del modelo ontológico, como se muestra en la Tabla 2.3. Hasta ahora ha sido poco cubierto el aspecto semántico de las ontologías con las métricas existentes, por lo que representa un campo de exploración que puede resultar interesante.

Capítulo 3

Visualización de Información y Analítica Visual

El lenguaje visual es una de las formas más antiguas de representación del conocimiento, y antecede al tradicional lenguaje escrito, por cerca de 25.000 años. Las representaciones visuales son efectivas, debido a que utilizan la capacidad y el poder de abstracción del sistema visual humano. En diversas áreas de Ingeniería se hace uso de representaciones visuales de información, y en los casos específicos de Ingeniería de *Software* e Ingeniería de Ontologías, estas representaciones se basan en el uso de diagramas. Los diagramas pueden transmitir información más concisa y precisa que el lenguaje ordinario, además, la información representada visualmente es más fácilmente recordada [200]. Un ejemplo de que la memoria visual es más potente, es el hecho de que es más fácil recordar un organigrama con fotografías de los trabajadores, que una relación textual de puestos de trabajo con los nombres de los mismos trabajadores.

El lenguaje visual se ha definido como aquel lenguaje que facilita la comunicación humana a través de imágenes o gestos determinados [62]. Esta definición, aunque breve, es muy precisa y transmite adecuadamente el objetivo principal del lenguaje visual: la transmisión de conocimiento o información. Esta capacidad de transmisión de información por medio de elementos visuales ha sido aplicada a diferentes áreas, y el área de la Representación del Conocimiento también se beneficia de esta.

Este capítulo describe dos áreas de investigación basadas en el uso de representaciones visuales de conocimiento: Visualización de Información (*Infovis*) y Analítica Visual (*Visual Analytics*). La primera sección proporciona las definiciones más ampliamente difundidas de *Infovis*, mientras que la segunda sección describe las principales propuestas desarrolladas en el ámbito de la

Ingeniería de Ontologías. La tercera sección proporciona una descripción muy específica de la problemática con las actuales herramientas de representación de ontologías, de acuerdo a la clasificación establecida. La cuarta sección describe la Analítica Visual y las diversas técnicas usadas para el análisis de la información; finalmente la quinta sección contiene las conclusiones del capítulo.

3.1. Visualización de Información

Hoy en día con el uso de los ordenadores ha crecido exponencialmente la capacidad para procesar grandes cantidades de información de una manera más rápida y eficiente. Sin embargo, el ser humano requiere del uso de ciertas herramientas o técnicas que le ayuden a entender de forma más clara y precisa la información representada. Principalmente, se trata de aquellas técnicas y herramientas que utilizan el sistema visual humano, o lo que es lo mismo, se basan en el uso del lenguaje visual.

El área de investigación que se enfoca en el estudio de estas técnicas o herramientas se conoce como Visualización de Información (*Infovis*). Básicamente, *Infovis* se basa en el uso de los ordenadores para procesar y transformar la información (generalmente datos abstractos) en elementos visuales que permitan a los usuarios comprender dicha información, mediante su sistema visual.

Dos de los principales investigadores en el área, Stuart Card y John Mackinlay [39] definen la Visualización de Información como “el uso de representaciones visuales de datos abstractos, interactivas y basadas en el uso del ordenador, para amplificar el proceso cognitivo”. Card y Mackinlay describen *Infovis* como un conjunto de tecnologías que utilizan la computación visual para amplificar el proceso cognitivo a través de la abstracción de la información [40]. Su perspectiva incluye, por un lado, el objetivo principal del área de investigación, que es ayudar en el proceso cognitivo y, por otro lado, enfatiza que el medio para alcanzar tal objetivo es a través del uso de representaciones interactivas.

La interacción con el usuario representa un aspecto trascendental en las visualizaciones de datos. En *Infovis*, la interacción está dada por interfaces gráficas avanzadas, que permiten acceder, explorar y visualizar la información, independientemente de los elementos gráficos comunes de interacción denominados *widgets*, o controles de la interfaz gráfica del usuario (Graphic User Interface *GUI*) [235]. Es importante distinguir claramente entre las interfaces gráficas y los elementos gráficos denominados *widgets*. Por un lado, las interfaces gráficas son representaciones visuales de información en ocasiones complejas, que permiten la interacción con el usuario. Esta interacción, puede hacerse a través del uso de este tipo de controles *widgets*, o a través de mecanismos

distintos.

Otra definición de Visualización de Información más reciente establece que “la Visualización de Información intenta trazar eficientemente datos de variables en dimensiones visuales para crear representaciones gráficas” [102]. En resumen, se puede afirmar que *la Visualización de Información tiene como su principal objetivo, reducir la complejidad para examinar y entender la información, mediante el diseño de técnicas apropiadas para la representación visual de los datos* [46].

Los sistemas de cómputo basados en visualización de información, están mejor enfocados para tareas de exploración, especialmente sobre grandes conjuntos de datos de información. Frecuentemente, los usuarios de un sistema de *Infovis* no tienen una pregunta específica en mente, sino que están interesados en el análisis de los datos, para aprender más sobre ellos o descubrir nueva información [168].

3.1.1. Proceso de visualización de datos

Un aspecto fundamental en *Infovis* es el conjunto de los datos que se visualizan. En términos generales, se puede decir que existen dos formas fundamentales de datos: por un lado los datos que representan valores simples y, por otro lado, las estructuras de datos [27]. Los datos de entrada en un sistema generalmente provienen de la salida de sensores, dispositivos, o representan valores reales del dominio que se pretende modelar. Estos datos mayormente son de tipo numérico o cadenas de caracteres, que representan los principales tipos de datos simples existentes. La gran mayoría de las veces, estos datos simples tienen que transformarse a ciertas estructuras de datos más robustas, que permitan su análisis, representación y acceso de forma más eficiente. Este proceso de tratamiento y transformación de los datos originales, representa el punto inicial en un proceso de visualización y análisis de datos.

El tratamiento de los datos es diverso, y está relacionado directamente con la solución del modelo del dominio que se pretenda. Por ejemplo, para los sistemas de bases de datos, el proceso de transformación consiste en la generación de estructuras de datos más robustas denominadas tablas, que se llenan con los datos simples de entrada al sistema.

Otra propuesta para el tratamiento de los datos en otros ámbitos como las ontologías, consiste en dividirlos en entidades y relaciones, donde las entidades representan los objetos que se desean visualizar, mientras que las relaciones definen las estructuras y patrones que relacionan a las entidades entre sí [274]. En ocasiones estas relaciones son explícitas, mientras que, por el contrario,

en otras hay que descubrirlas, y es este aspecto el principal propósito de la visualización. Tanto las entidades como las relaciones poseen atributos que representan cualidades que les permiten distinguirse unívocamente entre los diversos elementos.

Aparte de la transformación de los datos, otra característica importante del proceso de visualización de datos es la expresividad visual, definida como el número de variables visuales usadas en una visualización para que sea eficiente [274]. Estas variables son: la ubicación en el plano dimensional ($[x,y]$ para planos bidimensionales), la posición (horizontal, vertical o la profundidad), la forma (rectángular, circular, etc.), el tamaño, el brillo o la opacidad, la orientación, la textura y los colores.

Un aspecto fundamental es el uso del color; primeramente, el uso de colores ayuda a una persona a distinguir visualmente objetos de su entorno, ya que el cerebro puede claramente discernir entre los elementos, de esta forma, la fase de la detección de los colores es el primer punto de abstracción del cerebro. Una vez hecha esta distinción de elementos a través de los colores, el cerebro procesa otras variables visuales como son la forma, el tamaño o la textura, para finalmente establecer una abstracción visual de los elementos. Una vez hecho este proceso de abstracción visual, el siguiente paso del cerebro es el reconocimiento e interpretación de los elementos visuales. Es aquí en donde el ciclo completo de visualización de información permite deducir conocimiento a partir de los elementos visualizados.

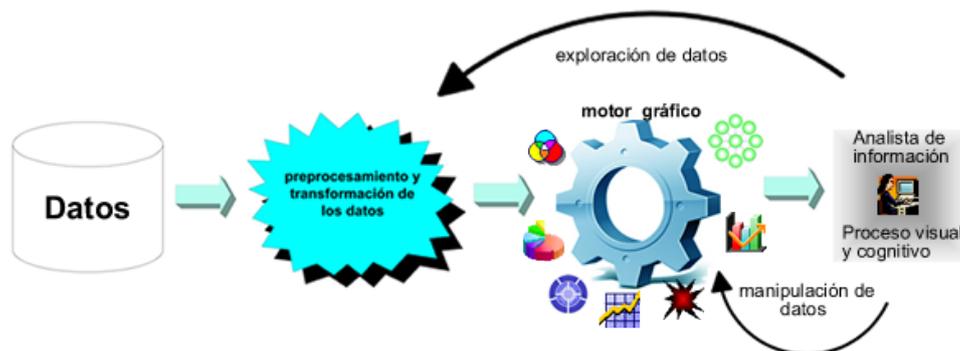


Figura 3.1: Diagrama esquemático del proceso de visualización definido en [274]. Este proceso define cuatro etapas básicas: el almacenamiento de los datos, el preprocesamiento y transformación de los datos, el motor gráfico y finalmente el análisis humano a través de un proceso visual y cognitivo.

El proceso completo de visualización de datos incluye cuatro etapas bási-

cas, combinadas con un número de ciclos de retroalimentación, que se ilustran en la Figura 3.1, y han sido definidos en [274]. Este proceso comienza con el almacenamiento y el tratamiento de los propios datos que se quieren visualizar. Después se desarrolla un preprocesamiento para transformarlos en estructuras de datos más robustas, que permitan hacer más eficiente su acceso, almacenamiento y manipulación por parte de los sistemas de cómputo. La tercera etapa la constituyen los algoritmos gráficos que producen una imagen de salida en la pantalla, que representa visualmente los datos procesados previamente. Finalmente, la última etapa corresponde a la percepción humana y al sistema cognitivo, en donde el usuario representa el factor fundamental del proceso, ya que es él, en esta última parte, el que genera el conocimiento.

Dentro de este proceso se pueden presentar diversos ciclos iterativos entre las diferentes etapas. Principalmente, como resultado de la interacción del usuario, ya sea mediante la exploración o la manipulación de los datos. En el caso de la exploración de un nuevo banco de datos, se requiere de nueva cuenta realizar las fases de preprocesamiento, transformación y finalmente el despliegue de las nuevas visualizaciones. Por el contrario, cuando se trata de una simple manipulación de los mismos datos por parte del usuario, el motor de gráficos se encarga de recalcular los valores y refrescar las visualizaciones. Estos casos son típicamente interacciones del usuario para realizar filtrado de datos, selección de elementos, ocultar o mostrar elementos previamente ocultos, el cálculo de diferentes distribuciones visuales de los elementos en pantalla, entre otros.

3.1.2. Modelado de datos

Una de las principales aplicaciones de la visualización de información, es el modelado de datos, ya que parte del proceso de visualización consiste en la construcción y la interpretación de diversos diagramas. Estos diagramas generalmente están formados por símbolos, y el significado de estos símbolos permite la comunicación de la información entre las diferentes personas que tienen acceso a los modelos.

Como ejemplo del uso de diagramas para modelar datos se puede mencionar el diagrama Entidad-Relación para el modelado de bases de datos. El objetivo del uso de diagramas, es representar por un lado, la forma en que se almacenan los datos, las estructuras creadas, el tipo de los datos y, por otro lado, también se provee información sobre la forma de extraer estos datos desde sus repositorios. Detrás de la filosofía de este tipo de modelos, está el enfoque en el cual el mundo real consta de una colección de objetos básicos que se denominan entidades, las propiedades que definen estas entidades, y las relaciones entre

las entidades [47, 19]. La Figura 3.2 muestra un clásico diagrama de un modelo Entidad-Relación, que representa el esquema de una base de datos mediante la representación de las entidades persistentes, sus propiedades y las asociaciones entre estas.

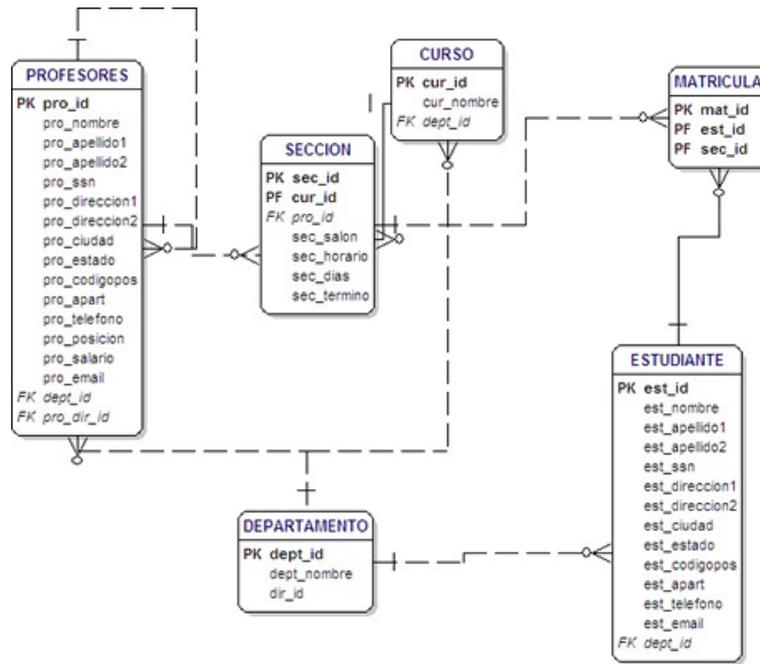


Figura 3.2: Un modelo de entidad-relación para una base de datos en un contexto académico. Se muestran las entidades persistentes del sistema, y las relaciones existentes entre estas.

Estos diagramas están diseñados para poder extraer información a través de un proceso de interpretación, en el que el usuario utiliza primero su capacidad visual y cognitiva para recorrer el diagrama e identificar los elementos, y después, elaborar un juicio sobre el significado de cada uno de ellos por separado y finalmente en conjunto. Este proceso de obtención de conocimiento a través de la interpretación de un diagrama o visualización, se realiza de forma más fácil y, sobretodo, más rápida, a través del uso de esquemas visuales, en comparación con el caso en el que todo este conocimiento hubiera sido descrito en lenguaje escrito o de alguna otra forma.

A medida que la cantidad de información aumenta, los diagramas o esquemas se vuelven más complejos de interpretar, y se vuelve necesario el uso de más elementos que permitan al usuario modelar u obtener el conocimiento que se pretende. Entre estos elementos, se cuentan el incremento en el número y la diversidad de las variables visuales, así como las técnicas de visualización y el

uso de visualizaciones dinámicas, que proporcionen al usuario diversos mecanismos de interacción. Es justo en estos escenarios, en los que la capacidad de los ordenadores es un componente fundamental para el modelado de grandes cantidades de datos, ya sea en una base de datos o en un modelo ontológico.

Aparte del modelo Entidad-Relación, existen otros tipos de modelos de datos como las bases de conocimiento en los sistemas basados en la representación del conocimiento. En este caso específico, las ontologías representan el modelo de los datos para una base de conocimiento, ya que definen las estructuras de los datos, los tipos, las relaciones, etc. El proceso de creación de un modelo ontológico, es muy similar al proceso de creación de un modelo relacional o de entidad-relación, e incluso puede compartir o reutilizar algunas de las soluciones de estos dos modelos. El modelado conceptual, como en el caso de las ontologías, se refiere a la actividad que se lleva a cabo durante el proceso de desarrollo de sistemas de información, y en las subsecuentes fases de mantenimiento, con la finalidad de construir una representación semántica de un dominio del mundo real [275].

3.2. Analítica Visual

El éxito en el uso de representaciones visuales para desarrollar el proceso cognitivo ha propiciado el surgimiento de un área científica basada en el uso de interfaces visuales para detectar nuevo conocimiento a partir de un conjunto de datos. Este área de investigación, relativamente nueva, se denomina Analítica Visual y se define como *la ciencia del razonamiento analítico soportado por interfaces visuales altamente interactivas* [259].

La Analítica Visual es un campo multidisciplinar que comprende el estudio del desarrollo de diversas técnicas de razonamiento analítico, representaciones visuales e interacción, así como una serie de transformaciones de datos [259]. Es un área en la que el usuario destaca como punto central del proceso de análisis, apoyado por diversas herramientas de cómputo para desarrollar todas las tareas necesarias.

La Analítica Visual va más allá de la simple representación visual de información, ya que pretende ofrecer una gama de herramientas con las que el usuario analiza la información y, en base a dicho análisis, se puede primero, obtener conocimiento a partir del modelo de datos y, segundo, tomar decisiones o ejecutar diversas acciones. Así mismo, combina técnicas de análisis automatizado con visualizaciones interactivas para un efectivo entendimiento, razonamiento y toma de decisiones, sobre la base de conjuntos de datos muy grandes y complejos [165].

Entre los principales objetivos de la Analítica Visual, destacan la elaboración de herramientas y técnicas capaces de sintetizar información, para poder obtener conocimiento a partir de un conjunto de datos, que generalmente son grandes, dinámicos y ambiguos. Por otro lado, también facilita el razonamiento analítico a través del desarrollo de aplicaciones de *software* que maximicen la capacidad humana para percibir, entender y razonar acerca de conjuntos de datos complejos y dinámicos [259].

Al igual que en *Infovis*, todas las visualizaciones y la interacción proveen los mecanismos que permiten a un usuario poder analizar y obtener conocimiento a partir de grandes volúmenes de información. El cerebro humano puede comprender información compleja, recibida a través de los diversos canales visuales, y la Analítica Visual utiliza esta habilidad para facilitar el proceso de razonamiento analítico. Este proceso es directamente afectado por la calidad en la representación de los datos de las visualizaciones, para lo cual se requiere de una eficiente administración y transformación de los datos, de acuerdo al enfoque descrito en la Figura 3.1.

Los datos representan la parte crucial de las visualizaciones, y estos deben transformarse en representaciones visuales apropiadas para lograr un análisis correcto y adecuado. Estas representaciones visuales deben ser acorde al tipo, a la fuente de obtención y, más importante aún, a lo que semánticamente representa esta información. Los tipos de datos corresponden a la propia naturaleza de la información, por ejemplo: valores numéricos, cadenas de caracteres, valores lógicos (falso o verdadero) o estructuras de datos más complejas. Las fuentes de obtención de los datos también son diversas, y se incluyen datos que pueden ser obtenidos en tiempo real, como lecturas de sensores, o información previamente almacenada como datos estructurados provenientes de una base de datos relacional, archivos de imágenes, audio, vídeo o archivos de texto [164, 167].

Para poder visualizar y analizar información, los datos deben transformarse de su estado original a una forma o representación que resulte fácilmente manipulable computacionalmente. Estas representaciones de datos son formas estructuradas que permiten la manipulación de los datos por medio de los ordenadores y frecuentemente resultan más complejas que los datos originales. Una transformación de datos se define como un procedimiento matemático que convierte datos en diferentes representaciones que pueden proveer un mayor conocimiento a un analista [259].

La Analítica Visual se aplica en diversas áreas de conocimiento, principalmente aquellas en las que tienen que analizarse y procesarse grandes cantidades de información como en Bioinformática [18], Geografía [10], Paleoceanografía

[258] o Medicina [261]. También la industria está sacando ventaja en el área de bases de datos [236], la industria de desarrollo de *software* [141, 256], en los sistemas de manufactura [163] o en la industria farmacéutica [82, 231].

3.2.1. Exploración visual de los datos

Las representaciones visuales y las tecnologías de interacción, proveen los mecanismos que permiten al usuario visualizar y entender grandes volúmenes de información de una sola vez, ya que la mente humana puede entender información compleja recibida a través de los canales visuales. La Analítica Visual se basa en esta capacidad, para facilitar el proceso de razonamiento analítico [259]. Este razonamiento tiene como principal objetivo, el descubrimiento y la adquisición de conocimiento, a través del uso de diversas técnicas de visualización e interacción con el usuario [165].

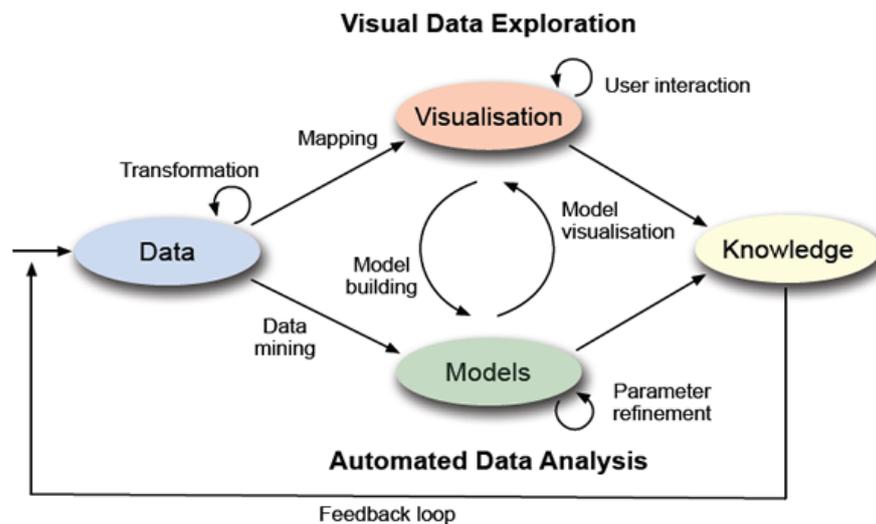


Figura 3.3: El proceso de exploración visual de los datos bajo el enfoque de la Analítica Visual definido en [165]. Destacan principalmente las fases de transformación de los datos en estructuras más complejas, el modelado de las estructuras, la representación visual y finalmente la fase de análisis para la obtención de conocimiento.

El proceso de exploración visual de los datos bajo el enfoque de la Analítica Visual se muestra en la Figura 3.3. Este proceso, definido en [165], describe las diferentes fases involucradas en el análisis de información bajo un enfoque de Analítica Visual. Primero comienza con la transformación de los datos, general-

mente en estructuras de datos más complejas que los simples datos básicos de entrada. Estas estructuras permiten la generación de ciertos modelos de datos, y a partir de estos modelos o las propias estructuras de datos, se alimentan las visualizaciones que permiten analizar y, finalmente, obtener conocimiento del conjunto de datos. Generalmente este conocimiento obtenido apoya la toma de decisiones.

Este proceso, presente en todas las propuestas de Analítica Visual, incluye también la interacción con el usuario como el principal mecanismo mediante el cual el usuario manipula las estructuras o los modelos de datos previamente creados. Esta interacción también permite la navegación del usuario sobre las diferentes visualizaciones.

3.2.2. Técnicas de Interacción Visual

La interacción del usuario con las visualizaciones representa un aspecto fundamental para el desarrollo del análisis de la información. Esta interacción se desarrolla a través del uso de diferentes técnicas o herramientas que permiten utilizar al máximo las capacidades de las representaciones visuales. A continuación, se describen algunas de las técnicas más importantes para la interacción con los usuarios, en el ámbito de la Analítica Visual.

3.2.2.1. Vistas enlazadas

Una de estas técnicas surge por la necesidad de combinar diferentes vistas y enlazarlas entre sí, de forma que los cambios o modificaciones hechos en cualquiera de las partes se ven reflejados en el resto de las vistas, lo que permite una mejor exploración de la información [209]. Esta técnica, denominada vistas enlazadas o coordinadas, permite a los usuarios comparar rápida y dinámicamente los datos visualizados en ventanas o vistas diferentes, y así poder analizar grandes cantidades de datos mediante la correlación entre diferentes tipos de representación de los mismos datos [237].

La coordinación o el enlace mantiene a las vistas sincronizadas durante la interacción, lo que permite la sincronización de la información entre las vistas y, por consiguiente, ayuda a la navegación de la información [207, 208]. Hoy en día prácticamente todas las herramientas de Analítica Visual hacen uso de vistas enlazadas (en mayor o menor medida), dado que el poder presentar diferentes perspectivas de análisis facilita enormemente el proceso de cognición y análisis de los datos visualizados.

3.2.2.2. Foco+Contexto

Otra de las técnicas que forman parte importante en el campo de la Analítica Visual es el *Foco+Contexto* [129]. Se utiliza para la integración de una vista general y una vista detallada de forma que en la misma visualización se integran ambas vistas. En su versión más frecuente, la técnica de *Foco+Contexto* se refiere a una distorsión desigual del espacio de visualización, tal que el espacio dedicado a cierto subconjunto de datos (datos enfocados) es mayor. Al mismo tiempo, el resto de la visualización se comprime para seguir mostrando el resto de los datos como contexto para que sirva de orientación al usuario [175, 242].

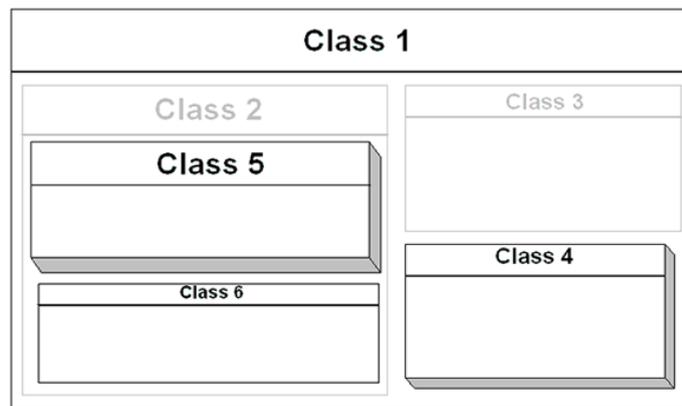


Figura 3.4: El *Foco+Contexto* puede lograrse aplicando una técnica de difuminado. En este caso, algunas clases se difuminan para restarles importancia al quedar en un segundo plano (Class2 y Class3), otras se dejan en su mismo contexto (Class1 y Class6), mientras que otras incluso se resaltan para asignarles un grado de importancia mucho mayor (Class4 y Class5).

Otra posibilidad para realizar el *Foco+Contexto* es el uso de otra técnica llamada difuminado (*blurring*), que consiste en desvanecer la tonalidad del color de ciertos elementos visuales. Esto ocasiona que ciertos elementos resalten sobre aquellos desvanecidos; esta técnica resulta muy útil cuando se tienen visualizaciones saturadas de elementos visuales. El uso del difuminado es una de las opciones para desarrollar *Foco+Contexto* sin tener que realizar distorsiones de tamaño de los elementos visuales, ya que únicamente se difuminan todos aquellos elementos visuales a los cuales se les quiere retirar el foco.

El resultado como se muestra en la Figura 3.4, es uno o más elementos en un primer plano a la vista, mientras que el resto de los elementos permanecen en un segundo plano apenas visibles, o incluso en un tercer plano. El cere-

bro interpreta esta representación como que los elementos en el primer plano adquieren mucha más importancia que los elementos en un segundo, tercero o enésimo plano.

3.2.2.3. Zoom Semántico

Otra de las técnicas ampliamente utilizada es el *zoom semántico* (*semantic zoom*), una técnica gráfica para balancear el contexto con el detalle, que, a diferencia del *zoom* físico que únicamente se limita a cambiar el tamaño del elemento visual desplegado, el *zoom semántico* cambia el tipo y el significado de la información desplegada [199]. La principal ventaja de esta técnica radica en que se puede mantener el contexto de la vista general, mientras que se ofrece una vista muy detallada de alguno de los elementos visuales [132].

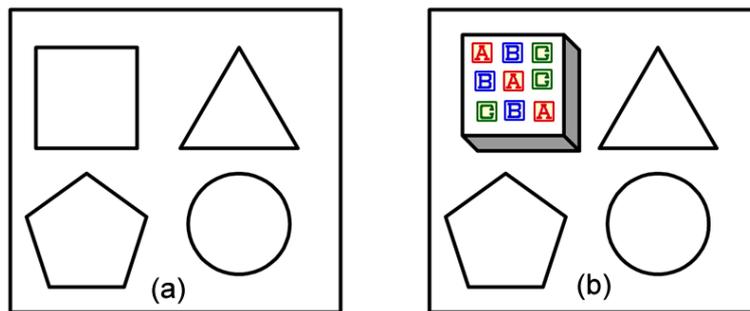


Figura 3.5: El *zoom semántico* cambia el tipo y la información desplegada, generalmente se utiliza para mostrar los detalles de un cierto elemento que ha sido seleccionado. También, puede o no, incluir un *zoom* físico para aumentar el tamaño del elemento seleccionado.

Esta técnica se ilustra en las Figuras 3.5 (a) y 3.5 (b). En la figura del lado izquierdo se representa el estado original de la visualización, mientras que en el lado derecho se ha aplicado una técnica de *zoom semántico* sobre uno de los elementos.

El efecto más general y ampliamente utilizado de aplicar esta técnica, es mostrar los detalles de un cierto elemento seleccionado, sin tener que mostrar los detalles de todos los elementos al mismo tiempo.

El uso de esta técnica tiene la ventaja de que las visualizaciones no se saturan con elementos que en un cierto instante de tiempo, el usuario no tiene interés en conocer. El uso de técnicas de este tipo, en las que la información o los detalles de los elementos se muestran u ocultan en respuesta a la demanda de los usuarios, permite al usuario centrar su atención en un elemento específico

a la vez.

El *zoom semántico* es una técnica que intenta evitar la sobresaturación de elementos en las visualizaciones. Es fundamental mantener las visualizaciones lo más claras posibles, para evitar que la saturación de elementos impida al usuario tener una clara percepción de la información. El uso de técnicas que despliegan la información con diferentes niveles de detalle, de acuerdo a la demanda del usuario, presentan una solución al problema de la saturación, un problema recurrente en visualizaciones con grandes cantidades de información.

3.2.2.4. Distribución de los elementos

Dentro del esquema de análisis visual, la distribución física de los elementos visuales sobre la pantalla (*layout*) juega un papel fundamental para la comprensión de los datos representados. Como se ha mencionado en la sección 3.1.1 al definir la expresividad visual, la ubicación de los elementos sobre el espacio dimensional, juega un papel fundamental para la comprensión y el análisis de las visualizaciones.

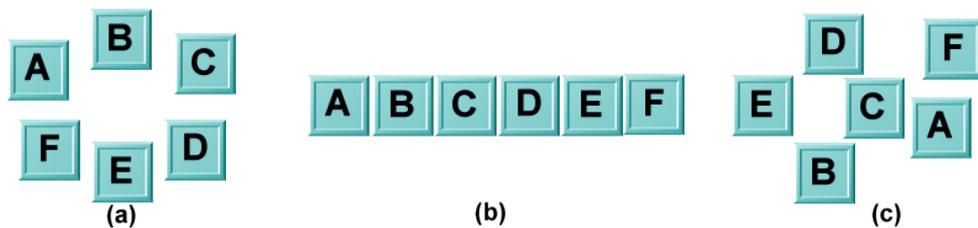


Figura 3.6: La distribución física de los elementos sobre el espacio dimensional es una de las variables de la expresividad visual. La Figura (a) distribuye los elementos de forma circular, y los ordena en el mismo sentido en el que giran las manecillas del reloj. La Figura (b) distribuye los elementos de forma lineal, y los ordena en sentido de izquierda a derecha, mientras que la Figura (c) distribuye los elementos de forma completamente aleatoria.

Las Figuras 3.6 (a), 3.6 (b) y 3.6(c), muestran tres distribuciones diferentes de seis elementos. La figura del lado izquierdo distribuye los elementos de forma circular, y los ordena en el sentido en el que giran las manecillas del reloj. La figura al centro distribuye los elementos de forma lineal, y los ordena en el sentido de izquierda a derecha. Por otro lado, la figura del lado derecho no tiene una distribución definida de los elementos, a los que coloca de forma aleatoria sobre el espacio dimensional.

La comparación de las tres figuras permite destacar la importancia que

tiene la distribución física de los elementos visuales. Es evidente que para las Figuras 3.6 (a) y 3.6 (b), el cerebro tiene la capacidad de, primero, reconocer que los elementos están organizados y ordenados siguiendo algún esquema definido y, segundo, construir un mapa mental que fácilmente puede recordar e incluso volver a reproducir. Estas capacidades se complican demasiado para la Figura 3.6 (c), ya que al no haber una distribución clara de los elementos, el cerebro humano no puede o le resulta mucho más complicado, construir un mapa mental de la representación visual.

En términos generales, siempre es deseable, en la medida de lo posible, que la representación de los elementos visuales tenga una distribución física sobre el espacio dimensional, que ayude al usuario a mejorar la percepción visual y el proceso de abstracción de la información mostrada.

3.2.3. Consultas Visuales

Las representaciones visuales, además de permitir analizar un conjunto de datos, también pueden utilizarse para realizar consultas sobre los mismos datos. El término *Visual Querying* en el área de las bases de datos, surgió por la necesidad de permitir que usuarios no expertos en SQL pudieran realizar consultas a una base de datos SQL [186]. Los sistemas que implementan la técnica de *Visual Querying*, permiten consultar las bases de datos mediante representaciones completamente visuales, de forma que facilita enormemente este proceso.

Este concepto ha sido extendido a diversas áreas, tales como los repositorios de *software* [227], el reconocimiento de patrones en imágenes [53] y las ontologías OWL mediante una propuesta denominada OntoVQL (*Ontology Visual Query Language*) [63]. Este lenguaje realiza consultas sobre un modelo ontológico, diseñadas mediante una interfaz gráfica basada en diagramas de conjuntos *Venn-Euler*, en donde los nodos representan conceptos y los operadores lógicos se representan como los conjuntos que los agrupan. El lenguaje se basa en el uso de la *Description Logic* [139] y utiliza una semántica formal basada en lógica, que puede traducirse en predicados lógicos de primer orden para inferir conocimiento a partir de la base de conocimiento establecida.

Para que esta técnica pueda desarrollarse con éxito, es necesario de nueva cuenta, que las visualizaciones sean adecuadas, intuitivas, y que permitan al usuario interactuar con el conjunto de datos para solicitar como resultado aquellos valores que respondan a una cierta consulta específica. En este caso, el proceso de exploración de los datos dentro del flujo que se muestra en la Figura 3.1, se repite constantemente mientras el usuario continúe realizando

consultas.

Un ejemplo de una aplicación basada en este enfoque, es la herramienta OZONE (*Zoomable Ontology Navigator*) [246], que permite realizar consultas a modelos ontológicos DAML. Estas consultas visuales se realizan a través de la navegación de los elementos en la ontología, y la interacción es bastante robusta. La principal desventaja de esta propuesta, es que está definida para el lenguaje DAML, que hoy en día no es la mejor opción para los modelos ontológicos, y poco a poco, incluso ha ido cayendo en desuso.

3.3. Conclusiones

Este capítulo ha hecho una recopilación de las más importantes definiciones de Visualización de Información, y se ha resaltado la aplicación de estas áreas a la representación del conocimiento y muy específicamente a las ontologías. En términos generales, el uso de técnicas de visualización de información provee a los usuarios de herramientas para la representación, análisis y obtención de conocimiento a partir de un conjunto de datos simples.

La filosofía detrás del planteamiento de soluciones basadas en Visualización de Información, es aprovechar al máximo las capacidades cognitivas humanas para obtener conocimiento a partir de una representación visual. Este proceso de análisis, debe apoyarse con la implementación de diversas técnicas que permitan al usuario interactuar con las representaciones visuales. En la medida en la que las técnicas de visualización e interacción sean suficientemente robustas, el proceso de abstracción por parte del usuario se desarrollará de forma sencilla y eficiente, incluso para grandes o complejas representaciones de datos.

Por otro lado, también se han descrito las actividades y su flujo dentro del proceso completo de visualización de información, que comienza con la transformación de los datos de entrada en el sistema, y termina con el análisis y la interacción por parte del usuario para la obtención de conocimiento. El resultado final son diversas representaciones visuales que permiten al usuario el análisis, modelado y representación del conjunto de datos del dominio de aplicación.

La Analítica Visual representa un proceso evolutivo a partir del área de la visualización de información, en donde el enfoque principal es que el proceso de exploración visual de los datos se centra en el usuario, quién hace uso de las herramientas disponibles para el análisis de los datos. Todas las técnicas se desarrollan teniendo en consideración que es la capacidad cognitiva y visual del usuario, la que permitirá el análisis y la obtención del conocimiento sobre los

datos del sistema. En este caso, los ordenadores implementan estas técnicas para soporte del usuario, quién se beneficia de este poder de cómputo para el análisis principalmente, de grandes cantidades de datos, y muchas veces también complejas.

Es justo en este aspecto, donde radica la importancia en el diseño de técnicas o herramientas basadas en Analítica Visual. Su principal objetivo es proporcionar a los usuarios con mecanismos robustos, que les permitan desarrollar los procesos de análisis, razonamiento, obtención de conocimiento y finalmente, la toma de decisiones.

Este enfoque de Analítica Visual, utiliza diversas técnicas o mecanismos, que hacen más eficiente su desarrollo. Estas técnicas generales, pueden usarse en cualquier dominio de aplicación, ya que son propuestas genéricas que resuelven diversa problemática. Las técnicas que se han descrito en este capítulo (vistas enlazadas, *Foco+Contexto*, *zoom* semántico y las consultas visuales), se basan principalmente en la interacción con el usuario, como el principal mecanismo para su desarrollo. Es precisamente el usuario, el objetivo principal para la definición de estos mecanismos, que están enfocados completamente a la interpretación humana, al igual que todas las técnicas que puedan definirse para las áreas de Visualización de Información y Analítica Visual.

Capítulo 4

Estado del arte

El modelado de datos es un proceso que puede resultar bastante complejo y requiere que se definan muy específicamente los flujos de trabajo con las actividades para construir el modelo de datos. Este capítulo se enfoca en los modelos de datos semánticos que se basan en el uso de ontologías. El proceso de construcción de una ontología es equivalente al desarrollo de un proceso de modelado de datos, ya que estas representan una definición de la estructura, los tipos, las relaciones e incluso, de la forma en la que se almacenan y accede a los datos en un sistema basado en representación de conocimiento semántico.

Como se ha descrito en la sección 2.3, la Ingeniería de Ontologías proporciona las directrices del conjunto de actividades de los procesos que cubren el ciclo de vida de las ontologías. Los principales procesos son la creación, gestión o mantenimiento, análisis y reutilización de las ontologías. Cada uno de estos procesos, tiene asociado un cierto flujo de tareas, de acuerdo a las metodologías propuestas, así como una serie de herramientas que permiten desarrollar estas tareas.

Este capítulo se enfoca en la discusión de cada uno de estos procesos. Se discuten las principales metodologías propuestas para su desarrollo, así como las herramientas que soportan estos procesos, y se hace especial hincapié en la problemática que estas presentan.

La estructura del capítulo comienza con una discusión del proceso de creación de ontologías, para posteriormente revisar los procesos de análisis, gestión y reutilización de ontologías, la problemática de las herramientas de visualización y modelado de ontologías, y finalmente, el capítulo termina con las conclusiones.

4.1. Creación de modelos ontológicos

El proceso de creación de ontologías se basa principalmente en el desarrollo de un conjunto de actividades, definidas en diversas metodologías que se han propuesto. De hecho, la gran mayoría de las metodologías propuestas, se enfocan principalmente en el proceso de creación. Uschold y King [263, 264] proponen una de las primeras metodologías, basada específicamente en la creación de modelos ontológicos. Estos autores describen algunas de las tareas más importantes involucradas en el proceso de creación, descritas en el esquema de la Figura 2.2.

Gruninger [117, 23] propone otra de las primeras metodologías denominada TOVE, como resultado de la experiencia en la creación de la ontología con el mismo nombre. Esta metodología se basa en el uso de lenguajes de lógica de primer orden, los cuales en la actualidad, ya han evolucionado a otros lenguajes más robustos.

Otra de las metodologías que han servido como base para posteriores propuestas es Methontology [71, 110]. Se trata de una propuesta bastante completa, ya que cubre todo el ciclo de vida de las ontologías [72]. Ohgren y Sandkuhl [213] han propuesto una metodología basada en el análisis de otras existentes (Uschold [262], Methontology [110] y Sugumaran [245]), en donde se proponen ciertas mejoras a estas metodologías. Básicamente, se enfocan en contextos de sistemas a pequeña y mediana escala, en los que proponen la reutilización de fragmentos de otras ontologías. Esta metodología también considera el ciclo de vida completo de las ontologías y la reutilización como aspectos cruciales a considerar siempre.

On-To-Knowledge [68] es una herramienta que define una metodología que se basa en lo que denomina bloques de construcción. Estos bloques contienen información para la administración del conocimiento a través de las ontologías. Como se ha mencionado previamente, las tareas definidas en On-To-Knowledge, son soportadas por la herramienta OntoEdit.

Las propuestas metodológicas que se pueden considerar más robustas, son la metodología DOGMA [241, 146, 148], que se centra principalmente en la usabilidad y reutilización de los modelos ontológicos, la propuesta de Uschold, y Methontology, que establece claramente los rasgos más característicos para la construcción de las ontologías. La gran mayoría de las otras propuestas, toman como base principalmente las metodologías de Uschold y Methontology, y han incorporado algunas adecuaciones que mejoran en cuanto a estas.

Diversas aplicaciones han utilizado estas metodologías para la creación de sus modelos ontológicos. Por ejemplo, Corcho et al. [50] describen la construc-

ción de una ontología en el dominio legal mediante el uso de Methontology. Este proceso de construcción se ha basado en la reutilización de una ontología existente, y sobre la que se han realizado diversas modificaciones. En otro ejemplo, [124] describe la construcción de un modelo ontológico en el dominio de la genética mediante el uso de la metodología *Knowledge Engineering Methodology*, y posteriormente, se utiliza el mismo dominio para ejemplificar el uso de la metodología DOGMA, en donde se define una ontología base, y una capa de validación, en un enfoque denominado doble articulación [143].

Hozo: Ontology Viewer

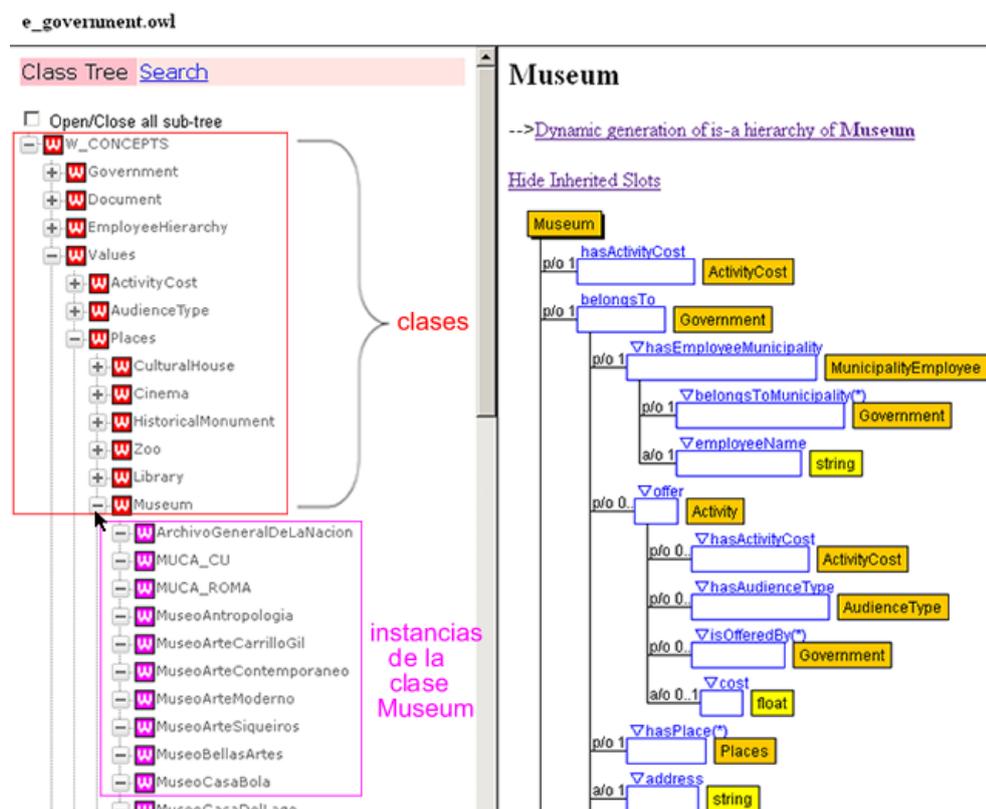


Figura 4.1: El editor de ontologías HOZO utiliza listas para desplegar la jerarquía (mostrada en el lado izquierdo de la visualización). En el lado derecho se muestran las propiedades de la clase, la jerarquía de superclases, así como las instancias de la clase seleccionada.

Aparte de las metodologías, el proceso de creación involucra el uso de lenguajes para la implementación de los modelos ontológicos. La gran mayoría de las metodologías proponen tareas muy generales, e independientes del

lenguaje de implementación. Zhdanova y Keller [283] describen los requisitos de un proyecto específico, y la forma en que estos autores seleccionaron un lenguaje de implementación para sus modelos ontológicos.

Gómez-Pérez y Corcho [109] hacen una exhaustiva comparación de los lenguajes de implementación de ontologías¹, para ayudar a los desarrolladores a encontrar los lenguajes que más se ajusten a sus necesidades, y ejemplifican su propuesta con la toma de decisiones para un proyecto específico.

Uno de los factores más importantes a considerar cuando se trata de escoger el lenguaje más apropiado para la implementación de modelos ontológicos, es la evolución de la propia ontología, que se vuelve dependiente del lenguaje seleccionado. Algunas propuestas han surgido en este sentido, como [16], que propone una solución mediante la identificación de los elementos cambiantes en la fase evolutiva de una ontología OWL. Maedche et al. [189] proponen un ambiente basado en ontologías llamado Ontologging, enfocado en la evolución de las ontologías. Esta propuesta se basa en la definición de una meta-ontología que permite la representación y el análisis, para desarrollar cambios ontológicos de una forma sistemática y consistente.

Aparte de la selección de la metodología y del lenguaje de implementación, el diseñador tiene que tomar una decisión acerca de la herramienta para implementar el modelo ontológico. Las herramientas para la creación de ontologías se pueden clasificar en dos grandes áreas: los editores y las herramientas de modelado visual. Suresh et al. [251] hacen un estudio comparativo de diferentes editores de ontologías, entre los que destacan principalmente OntoTrack, Protégé, HOZO y OWLEditor.

La Figura 4.1 muestra una vista de la herramienta HOZO [247, 176], muy similar en cuanto a su diseño a otras herramientas basadas en web como SWOOP, WebODE, Ontolingua y OwlSight descritas en la sección 2.3.4. HOZO se basa en la utilización de listas para la representación de la jerarquía de clases. En la Figura 4.1 se ha resaltado una parte de la taxonomía de clases, con un recuadro en color rojo, mientras que las instancias de la clase seleccionada se han resaltado en un recuadro en color púrpura. Esta representación visual es incorrecta, ya que al mezclar las clases con las instancias, la percepción es que las instancias pertenecen a la jerarquía de conceptos (relaciones *is-a*), algo que no es correcto. Al contrario, las instancias son valores específicos de las clases, pero no relaciones de jerarquía. Esta representación mezcla dos tipos de relaciones diferentes, que representa exactamente de la misma forma, lo que desde un punto de vista semántico no es correcto. Es en el uso del color en el

¹con la excepción del lenguaje OWL que no había sido definido para el momento de la publicación del artículo

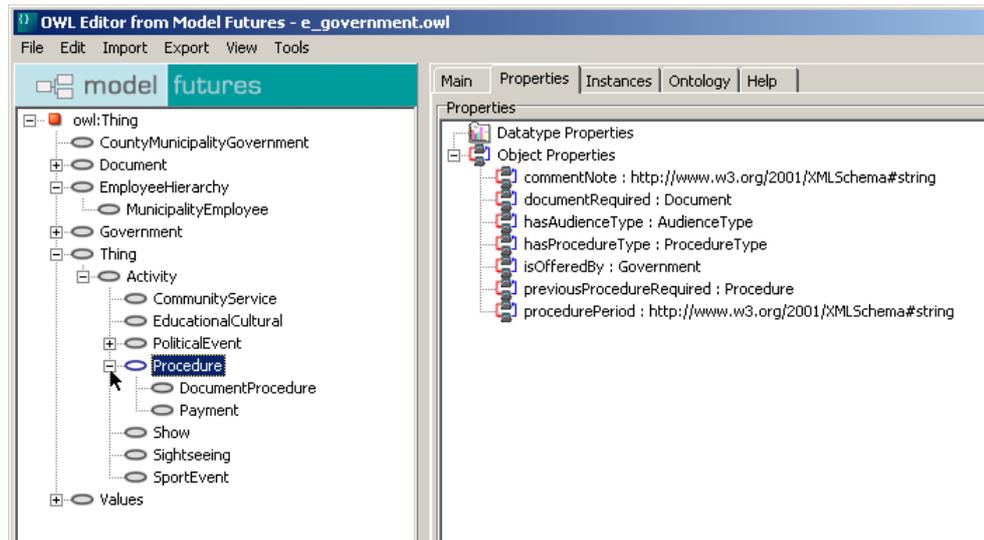


Figura 4.2: La herramienta OwlEditor es una aplicación de escritorio que permite editar ontologías OWL en formato textual, sin implementar visualizaciones ni mecanismos de interacción robustos.

elemento, donde se puede identificar si se trata de una clase o de una instancia de la clase.

El lado derecho de la Figura 4.1 muestra la representación de las propiedades de la clase seleccionada. Esta representación también se basa en el uso de listas, en consecuencia, la representación es muy limitada, ya que únicamente puede representar las propiedades del elemento seleccionado (solamente uno a la vez). Este tipo de distribución de elementos basada en listas, no es la más adecuada para la representación de relaciones, como se detalla más adelante en la sección 4.4.

En términos generales, las herramientas basadas en el uso de navegadores web, no poseen una funcionalidad robusta para la visualización de las ontologías. Existen otro tipo de editores que son aplicaciones de escritorio muy similares entre sí, como SWOOP [273], DL-workbench [161, 162] y OwlEditor [20]. Esta última herramienta que se muestra en la Figura 4.2, no tiene visualizaciones implementadas, tampoco ha sido actualizada y su estado al parecer continúa siendo de prueba. Por otra parte, DL-Workbench se basa en Description Logics y, más específicamente, en el lenguaje DAML+OIL para aplicaciones de Web Semántica [219].

Por otro lado, las herramientas para el modelado visual de ontologías implementan mecanismos más robustos de interacción visual, que permiten la

creación de ontologías de una mejor manera. Entre estas herramientas se encuentran principalmente OntoTrack, TopBraid Composer, OntoStudio, NeOn Toolkit y Semantic Works, todas estas descritas en la sección 2.3.4.

OntoTrack [183, 184] es una herramienta de modelado de ontologías cuya principal filosofía representa su mayor desventaja. El enfoque que sigue es la representación de todos los elementos en una misma visualización, lo que origina que los modelos se saturen de información fácilmente. Una situación similar de saturación de elementos visuales sucede con las herramientas TopBraid Composer, OntoStudio y NeOn Toolkit. Esta problemática se describe en detalle en la sección 4.4.

4.2. Análisis de ontologías

El proceso de análisis de ontologías desde un enfoque basado en el usuario se desarrolla mediante el uso de dos mecanismos principales: por un lado, un conjunto de métricas definidas para la evaluación de las ontologías y, por otro lado, el más importante, mediante el uso de herramientas de análisis que permitan al usuario desarrollar un proceso cognitivo sobre los modelos ontológicos. Existen dos enfoques cuando se habla de análisis ontológico. El primero se refiere al análisis de un modelo ontológico existente, generalmente con el objetivo de reutilizar dicho modelo, mientras que el segundo enfoque se refiere al uso de un modelo ontológico para analizar un conjunto de datos de algún dominio específico como la Bioinformática [169] o el análisis de un modelo de datos en una base de datos [272].

Cuando se realiza un proceso de análisis sobre una ontología, generalmente se pretende realizar una evaluación de la misma, con la finalidad de tomar una decisión al respecto [120]. Guarino y Welty [121] proponen OntoClean, una metodología para analizar y evaluar un modelo ontológico, desde una perspectiva conceptual filosófica. El resultado es la evaluación de cada uno de los conceptos en el modelo ontológico, lo que define una taxonomía de conceptos mejor definida, a través de la reorganización de los conceptos y la eliminación en lo posible de la herencia múltiple.

Otro tipo de evaluación enfocada en la taxonomía, es el que propone Gómez-Pérez [106, 108], en el cual se pretenden detectar cierto tipo de errores producidos en el diseño del modelo ontológico, como errores de referencias circulares, errores de partición de clases, errores de redundancia, gramaticales, semánticos y de estado incompleto.

Vrandecic [270] propone un marco de trabajo para medir la calidad de

una ontología, y detectar errores y omisiones. Este marco de trabajo es muy completo ya que permite analizar diversos aspectos de la ontología, como la sintaxis, la estructura, la semántica, la representación y el contexto.

Supekar et al. [248] proponen una metodología para un análisis cuantitativo y cualitativo de las ontologías. Estos autores proponen evaluar la calidad del conocimiento en el modelo ontológico, a través de la definición de una serie de características inherentes a la ontología.

Otro mecanismo para el análisis ontológico aparte de las metodologías y los marcos de trabajo, son las métricas, que en general, son valores que permiten estimar o medir alguna característica de un sistema de información. Las métricas han tenido una gran proyección, principalmente en el área de Ingeniería de Software, y posteriormente se han adecuado también para la Ingeniería de Ontologías.

Dos aspectos fundamentales para medir en un sistema de *software*, son el acoplamiento y la cohesión, y en este sentido, existen varias propuestas de métricas [7, 212, 280]. Chidamber y Kemerer [45] proponen unas métricas para analizar el acoplamiento interno en sistemas de *software*. Estas métricas denominadas CBO (*Coupling Between Objects*), se aplican en el área de Ingeniería de Software, especialmente, en sistemas diseñados bajo el paradigma orientado a objetos [244]. Estas métricas de acoplamiento se han diseñado para medir la complejidad en el diseño de las clases de un sistema orientado a objetos. La base teórica que sustenta el acoplamiento, es la noción de que un objeto está acoplado a otro, si alguno de ellos actúa sobre el otro [45]. Esta noción permanece para el caso de las ontologías, en donde una clase está acoplada con otra, si entre estas dos, existe una relación a través de una propiedad.

El análisis de este tipo de acoplamiento permite, en el caso de los sistemas orientados a objetos principalmente, evaluar la modularidad del diseño para fomentar la reutilización de componentes de *software*. En el caso de las ontologías sucede algo similar, ya que también se fomenta la reutilización de modelos ontológicos. Es importante saber exactamente para cada concepto que se pretende reutilizar con que otros conceptos se relaciona, y determinar si es necesario definir nuevas relaciones o no. Por otro lado, el análisis del acoplamiento interno de una ontología permite determinar aquellas clases que tienen más relaciones, y el tipo de acoplamiento también indica información semántica de la clase.

En el campo de las ontologías, hay algunas propuestas de métricas, principalmente para estimar el acoplamiento, la cohesión o las métricas del esquema de la ontología. Un resumen sobre las diversas propuestas de métricas, se provee en [99]. Básicamente, se han definido dos categorías principales para analizar

un modelo ontológico [254]: por un lado, las métricas del esquema o la estructura, y por otro lado, las métricas enfocadas en la semántica.

La gran mayoría de las métricas definidas, se basan en la estructura de las ontologías, como se ha ilustrado en la Tabla 2.3 en la sección 2.3.3. Estas métricas son principalmente OntoQA [253], que analiza el esquema de la ontología y las instancias de la base de conocimiento, las métricas de acoplamiento definidas por Orme et al. [215], las métricas de cohesión definidas por Yinglong et al. [282], o las métricas definidas por las herramientas *Ontology Metrics* o *Protégé* que se muestran en la Figura 2.3.

Orme et al. proponen un conjunto de métricas de acoplamiento para sistemas basados en ontologías OWL. Como se ha mencionado previamente, el acoplamiento entre las clases de una ontología está dado por las relaciones objeto que establecen un vínculo entre las clases. Estas métricas se basan principalmente en el acoplamiento externo a la ontología. Su objetivo principal es el de medir las relaciones de las clases en la ontología, con clases en otras ontologías que se han reutilizado.

Un aspecto que no cubren las métricas propuestas por Orme et al. es el acoplamiento interno en la propia ontología. Este acoplamiento es importante considerarlo, ya que proporciona una vista global de las clases, y la importancia de estas, desde la perspectiva de sus relaciones con el resto de clases. En una ontología, las clases que tienen más relaciones, suelen representar los conceptos más importantes dentro del dominio de aplicación. Sin embargo, desde el punto de vista semántico, una relación entre dos clases proporciona información no solo estadística, como lo contemplan las métricas de estructura, sino también semántica. Es en este aspecto, en el que las métricas de acoplamiento propuestas por Orme et al. son insuficientes, ya que el concepto general de acoplamiento y relación, va más allá de un simple valor numérico.

Algunas de las métricas y herramientas se basan en el enfoque de la definición de meta-ontologías, como *Cyc* [21], *Ontolingua* [66] y *Ontologging* [189]. Gangemi et al. [81] proponen un modelo formal para evaluar las ontologías, basado en una meta-ontología denominada O^2 . El análisis con esta meta-ontología permite identificar tres tipos diferentes de medidas de evaluación: por un lado, métricas de la estructura o el esquema de la ontología, métricas de funcionalidad relacionadas con el uso de la ontología y métricas relacionadas con el perfil de la ontología, principalmente, un conjunto de anotaciones con información para el usuario.

Tartir [255] recopila las más importantes propuestas para la evaluación de las ontologías, y elabora una clasificación basada en tres aspectos: el primero concierne con la evolución de las ontologías, el segundo se enfoca en las pro-

puestas de evaluación desde un punto de vista lógico, y el tercer enfoque se basa en el uso de métricas. En este último enfoque se clasifican la gran mayoría de las propuestas. Otras conclusiones son que la gran mayoría de las técnicas de evaluación se basan en el esquema ontológico, y que estas están orientadas completamente a su interpretación por parte del usuario, mayormente durante la fase de desarrollo.

Las métricas son valores numéricos que permiten evaluar los modelos ontológicos en ciertos aspectos. La representación de este tipo de valores numéricos se ha realizado principalmente mediante el uso de tablas estáticas, como lo muestra la Figura 2.3 de la sección 2.3.3. Sin embargo, estas tablas estáticas carecen de una interacción con el usuario, y simplemente se limitan a representar los valores numéricos.

Una propuesta genérica que agrega interacción con las tablas, denominada *table lens* [226, 150]. Esta técnica visual representa una tabla de valores, con funcionalidad adicional, que se utiliza principalmente para desplegar grandes cantidades de información, ya que ofrece ciertas ventajas sobre las tradicionales tablas estáticas. Entre las ventajas principales, se puede mencionar que implementan la técnica de *Foco+Contexto* para distorsionar filas o columnas específicas de las tablas. Esta distorsión permite aplicar el foco sobre ciertos elementos, sin perder el contexto global del resto, además de proporcionar una vista global que permite establecer correlaciones entre las dimensiones bajo análisis (columnas de la tabla).

El análisis y la interpretación de los resultados de estas métricas, se lleva a cabo principalmente mediante el uso de diversas herramientas. Existe una gama diversa de herramientas de análisis, algunas de las cuales se centran simplemente en calcular métricas como ODEval [233] o AktiveRank [5, 4]. Algunas otras tienen un soporte visual para el análisis, como Jambalaya [11, 185], TGVizTab [3], y los ambientes de desarrollo NeOn Toolkit, TopBraid Composer o Semantic Works.

ODEval es una herramienta centrada específicamente en la evaluación de las taxonomías de conceptos, con la finalidad de detectar un conjunto específico de problemas (descritos en [107, 108]), principalmente, la inconsistencia, los estados incompletos y la redundancia. AktiveRank [5, 4] propone un conjunto de métricas para la búsqueda de ontologías que se relacionen a un conjunto de términos introducidos por el usuario. Este análisis genera una clasificación ontológica, y permite tomar la decisión de aquellas ontologías que puedan reutilizarse dentro de algún dominio específico.

Por otro lado, las herramientas de visualización y modelado de ontologías permiten al usuario realizar un análisis de un modelo ontológico desde una

perspectiva diferente a la de las métricas. Las herramientas que se basan en el uso de métricas, desarrollan un proceso automatizado para determinar ciertos valores numéricos (la gran mayoría de las veces), mientras que las herramientas de visualización se basan en la interacción con el usuario. Esta interacción permite al usuario analizar el modelo ontológico desde su propia perspectiva y capacidad de análisis, lo que representa un enfoque completamente opuesto al proceso automatizado de las métricas.

Para que este proceso interactivo de análisis pueda desarrollarse de forma eficiente, se requiere que las herramientas provean mecanismos robustos de visualización e interacción.

El análisis de un modelo ontológico debería al menos cubrir las mismas tareas definidas para la fase de implementación, que se muestran en la Figura 1.3, por un lado, el análisis de la taxonomía de conceptos, el análisis de las propiedades intrínsecas de determinadas clases, el análisis de las relaciones entre ciertas clases y en caso de haber, el análisis de las instancias. Generalmente se comienza con el análisis de la taxonomía de conceptos en la ontología. Esta actividad requiere la búsqueda específica de ciertas clases en el modelo ontológico, y la inspección de sus propiedades intrínsecas.

La gran mayoría de las herramientas de visualización han implementado la misma solución que Protégé para la representación de las taxonomías. Esta solución es el uso de listas jerárquicas, como se muestra en la Figura 2.5 con TopBraid Composer, en la Figura 2.6 con NeOn Toolkit, en la Figura 4.1 con HOZO, en la Figura 4.2 con OwlEditor y en la Figura 4.3 con OntoStudio, en donde se ha resaltado en un recuadro azul del lado izquierdo. Otras soluciones se basan en el uso de diagramas de clase UML (TopBraid Composer), grafos (GraphViz, TgVizTab) o mapas conceptuales (Semantic Works).

El análisis de una taxonomía representada en una lista jerárquica no es una solución eficiente, ya que como se describe más adelante en la sección 4.4, provoca la pérdida del contexto global de la jerarquía de la clase. Algunas de las herramientas implementan un mecanismo de búsqueda para las clases en la taxonomía, lo que evita el tedioso proceso de realizar una búsqueda manual sobre la lista de clases.

Para el análisis de las propiedades intrínsecas de una clase, se pueden distinguir dos enfoques principalmente. El primero y más ampliamente difundido, se basa en el uso de listas jerárquicas como en el caso de HOZO (Figura 4.1), y OwlEditor (Figura 4.2). Otra opción que surge primeramente en Protégé y que adoptan la gran mayoría de las herramientas, es representar todas las propiedades del modelo ontológico mediante una gran lista jerárquica, como se ilustra en las Figuras 2.5 con TopBraid Composer y 2.6 con NeOn Toolkit. El

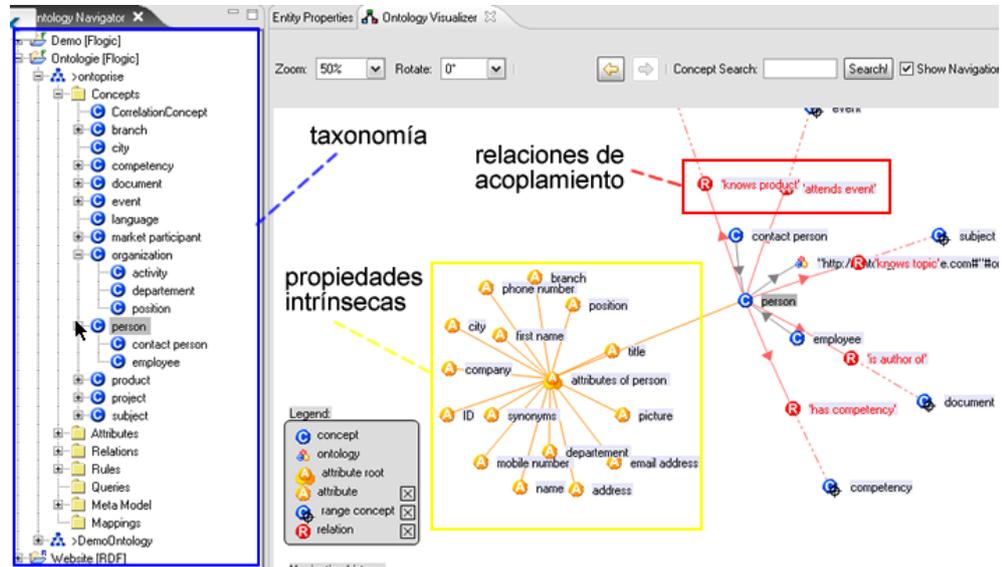


Figura 4.3: OntoStudio representa la taxonomía de conceptos mediante el uso de listas jerárquicas, como se ilustra en el recuadro definido en color azul. Por otro lado, las propiedades intrínsecas se representan mediante el uso de nodos en un grafo, como se resalta en el recuadro definido en color amarillo, mientras que las relaciones de acoplamiento se resaltan mediante el uso de nodos en color rojo.

segundo enfoque se basa en el uso de grafos como el caso de GraphViz, OntoSphere (para el caso de espacios tridimensionales), TgVizTab que se muestra en la Figura 2.7 y con OntoStudio, que se ilustra en la Figura 4.3, en donde se ha resaltado un recuadro en color amarillo que contiene estas propiedades intrínsecas.

El análisis de las relaciones entre las clases, se ha implementado principalmente mediante el uso de grafos, como en el caso de Jambalaya, OntoStudio (Figura 4.3), TopBraid Composer (Figura 2.5), NeOn Toolkit (Figura 2.6) y TgVizTab (Figura 2.7). La Figura 4.3 ilustra la representación de las propiedades de acoplamiento entre las clases, con un recuadro en color rojo sobre la visualización.

Finalmente, el análisis de las instancias de clases en una base de conocimiento, se ha abordado desde dos perspectivas básicamente. De nueva cuenta, el primer enfoque es el uso de listas jerárquicas (OwlEditor, HOZO, Protégé, TopBraid Composer), mientras que el segundo enfoque es el uso de grafos (Jambalaya, NeOn Toolkit).

La sección 4.4 elabora un análisis y una discusión muy específica sobre la problemática de las herramientas de visualización y de modelado.

4.3. Procesos de gestión y reutilización de ontologías

Uno de los hitos principales en el uso de modelos ontológicos es el principio de reutilización. Este concepto es importante ya que permite ahorrar esfuerzo en el diseño de un nuevo modelo ontológico. El concepto de reutilización se utiliza ampliamente en el área de Ingeniería de Software, en donde se ha definido como *el proceso de creación de sistemas de software, a partir de recursos existentes, en lugar de construir los sistemas de software desde el comienzo* [155]. En esta definición, los recursos existentes son bloques de construcción también llamados *componentes reutilizables de software*.

Otra definición más extensa de reutilización, establece que *la reutilización del software es la reaplicación de una variedad de tipos de conocimiento de un sistema a otro sistema similar, para reducir el esfuerzo de desarrollo y mantenimiento* [222].

La reutilización del *software* es un tema que ha sido ampliamente abordado desde hace décadas [222, 29, 155]. Se han propuesto algunas técnicas como los patrones de diseño [80], también algunas metodologías como el Proceso Unificado de Desarrollo de Software (*Unified Software Development Process*) [58], e incluso paradigmas como el Orientado a Objetos, con la finalidad de fomentar el desarrollo de sistemas altamente reutilizables [225, 140].

Muchos de los conceptos aplicados a la Ingeniería de Software, también pueden aplicarse a la Ingeniería de Ontologías, ya que, a final de cuentas, los sistemas basados en conocimiento, también requieren un desarrollo de *software*. Uno de estos conceptos que han permitido el desarrollo de sistemas altamente reutilizables, es principalmente el enfoque basado en el diseño de componentes. Estos componentes son bloques diseñados en un nivel de abstracción demasiado alto, que representan la base para otros componentes con funcionalidad más específica. Visser y Bench [268] describen un caso de estudio de reutilización de una ontología en un marco legal, bajo el enfoque de estos componentes reutilizables.

Bontas et al. [34] describen los costes y beneficios de la reutilización de ontologías para el dominio específico de la medicina, pero que pueden extenderse a otros dominios de forma muy similar. En términos generales, la reutilización debería disminuir costes en procesos de ingeniería, siempre y cuando, las tareas de búsqueda, análisis y evaluación de las ontologías no se vuelvan demasiado

costosas. Por otro lado, Simper [238] analiza la reutilización de las ontologías para el contexto específico de la Web Semántica.

La metodología DOGMA [143, 147] también se basa en el enfoque de componentes, con su propuesta de la definición de una metodología base, a partir de la cual se pueden definir submodelos ontológicos más específicos a la aplicación. En el caso de la Ingeniería de Ontologías, se puede establecer que los componentes reutilizables corresponden a modelos ontológicos muy generales, esto es, con un nivel de abstracción muy alto, que puedan compartirse entre diversas aplicaciones semánticas.

El proceso de reutilización en modelos ontológicos se realiza a través del diseño de varios submodelos. Comienza con aquellos que representan los niveles más altos de abstracción, hasta los de niveles más específicos a los dominios de aplicación. Los submodelos más abstractos pueden reutilizarse entre diferentes dominios de aplicación, tal y como se pretende en la metodología DOGMA. Es indispensable que el diseño de las soluciones abarque la implementación de varios submodelos ontológicos, en lugar de tener un único gran modelo muy específico del dominio de aplicación.



Figura 4.4: El proceso de reutilización de ontologías comienza con la fase de abstracción del dominio del problema. La segunda fase corresponde a una búsqueda de los modelos ontológicos que puedan reutilizarse. La tercera fase corresponde al análisis de los modelos ontológicos reutilizados, con el objetivo de hacer un diagnóstico sobre los cambios exactos que se requieren. Después, la última fase corresponde a la implementación del nuevo modelo ontológico.

La Figura 4.4 muestra las principales fases del proceso de reutilización de ontologías, definidas por las metodologías existentes. La primera fase corresponde al propio proceso de abstracción del dominio y el planteamiento de las soluciones. La segunda fase, corresponde a una búsqueda en diversos repositorios, de modelos ontológicos previamente creados en otros dominios de aplicación similares, que puedan reutilizarse. Esta búsqueda suele ser costosa en tiempo, no suele haber documentación de las ontologías, y otro problema es la gran diversidad de lenguajes de las ontologías. Engel et al. [61] proponen un méto-

do para facilitar esta tarea de búsqueda de ontologías, especialmente enfocado a la búsqueda de servicios web, así como un mecanismo de evaluación. Esta propuesta se basa en el análisis de los resultados de los motores de búsqueda, con respecto a las ontologías en el dominio de interés que se intenta reutilizar.

Esta búsqueda arroja como resultados uno o diversos modelos candidatos que pueden reutilizarse. Una vez que se tienen identificados estos modelos candidatos, es necesario desarrollar una fase de análisis (descrita en la sección 4.2), para evaluar que efectivamente son útiles y correctos para el modelo de datos, y para detectar aquellos conceptos que no han sido definidos. El objetivo principal de este análisis, es el de establecer claramente si los modelos candidatos detectados, realmente pueden utilizarse, así como detectar todos aquellos aspectos que haya que incluir (clases, propiedades, relaciones, etc.). Una vez terminado el proceso de análisis, se procede con la fase de implementación del modelo, de acuerdo a las soluciones planteadas en la fase de abstracción.

Aparte de la búsqueda, otro aspecto importante es la evaluación de las ontologías. Brank et al. [37] hacen un resumen de las diversas técnicas para la evaluación de ontologías. Vrandecic [270] propone un marco de trabajo con diversos criterios sobre ciertas características para medir la calidad de una ontología, principalmente para incrementar la disponibilidad de las ontologías (reutilización), y disminuir los costos de mantenimiento. La propuesta de Engel et al. [61], también propone un mecanismo para evaluar las ontologías a partir de un conjunto de palabras clave definidas, que pueden asociarse con las ontologías encontradas. Obrst [211] discute algunas estrategias y enfoques para la evaluación de las ontologías desde un punto de vista semántico. Principalmente propone el cálculo de la *distancia semántica* y la *similitud semántica*, dos medidas utilizadas en la teoría de grafos para interpretar el significado de la relación entre dos nodos.

Una de las metodologías más conocidas para la evaluación de ontologías es Ontoclean [120, 121]. Ontoclean se basa en la definición de ciertas metapropiedades que se asignan a las clases, con el objetivo de proporcionar un mecanismo para poder interpretar el valor semántico de los conceptos. Lozano-Tello y Gómez-Pérez [187] proponen un método para seleccionar la ontología más apropiada para reutilización. Esta metodología llamada Ontometric, se basa en tres conceptos para la selección de una ontología: el contenido, la profundidad máxima en la jerarquía y el camino más largo hacia una clase. Los autores de Ontometric establecen que una buena práctica es que las instancias de las clases deberían ubicarse en los niveles inferiores de la jerarquía, ya que los niveles superiores corresponden a conceptos más abstractos.

Cuenca et al. [114] proponen un conjunto de tareas relevantes para la re-

utilización modular de ontologías. Esta propuesta define diversos conceptos como *extensión conservativa*, *seguridad* y *módulo*, aplicados a las clases de una ontología representada en un lenguaje basado en lógica. Estos autores también proponen una serie de algoritmos para las tareas de modularización y segmentación de los modelos ontológicos.

Otros procesos asociados con la reutilización, son la mezcla, el mapeo y la alineación de ontologías. Por un lado, la mezcla de varias ontologías, termina con una sola ontología coherente, mientras que la alineación establece ligas entre los conceptos que son equivalentes [124]. El mapeo de ontologías se presenta en ambientes distribuidos como la Web Semántica, en donde es necesario hacer accesibles las ontologías a múltiples aplicaciones, por medio de una capa común con estas ontologías [152].

4.4. Problemática visual de las herramientas

Como se ha descrito en el capítulo 3, el uso del lenguaje visual para la representación de conocimiento y, especialmente de ontologías [43, 44], facilita los procesos de modelado, análisis y abstracción de conocimiento. Para abordar la discusión de la problemática en las herramientas desde el punto de vista de las visualizaciones, estas se agrupan de acuerdo a la clasificación hecha por Katifori et al. [159], en la que no se consideran los editores basados en texto, ni tampoco aquellos basados en el navegador web, ya que estas herramientas no proveen mecanismos de interacción visual.

Esta clasificación se divide en seis categorías, de acuerdo a las diferentes características de la presentación, interacción, técnicas, funcionalidad soportada o el número de dimensiones; sin embargo, la mayoría de las herramientas pueden pertenecer a más de una categoría. Estos seis grupos corresponden a: las listas jerárquicas, los grafos, las herramientas con técnicas de *zoom*, las que utilizan técnicas que cubren todo el espacio posible, las que utilizan la técnica de *Foco+Contexto* (descrita en la sección 3.2.2.2), y aquellas basadas en el uso de tecnología tridimensional. Otras categorías que pueden incluirse en esta clasificación, que no fueron consideradas, son por un lado, las herramientas que se basan en el uso de UML para visualizar y modelar ontologías, y las que se basan en el uso de mapas conceptuales. Al añadir estas dos últimas nuevas categorías, resultan un total de ocho categorías posibles, que se listan a continuación, así como las herramientas que pertenecen a cada una de ellas, las cuales han sido previamente descritas en la sección 2.3.4:

- Listas jerárquicas.- Las listas son representaciones enfocadas en repre-

sentar jerarquías. Esta técnica la utilizan la gran mayoría de las herramientas, como Protégé, OntoEdit, Kaon, OntoStudio, NeOn Toolkit, SWOOP, SemanticWorks, TopBraid Composer, HOZO y OWLSight.

- Grafos dirigidos.- Esta categoría representa a las ontologías como un conjunto de nodos interconectados por aristas. Estas aristas definen un sentido semántico (origen-destino), en el que la relación debería interpretarse. Las herramientas clasificadas en esta categoría son OntoViz, OntoSphere, TopBraid Composer, OntoStudio, Neon Toolkit, OntoTrack, Dmitrieva y Jambalaya.
- Técnicas de *zoom*.- Esta categoría contiene a todas las herramientas que implementan la funcionalidad de *zoom* para analizar los detalles de los elementos en la visualización. Las principales herramientas en esta categoría son Jambalaya, OntoTrack y SWOOP.
- Técnicas que cubren todo el espacio.- Esta categoría incluye todas aquellas técnicas que se basan en el concepto de utilizar todo el espacio posible de la pantalla, dividiéndolo entre todas las clases. Los *treemaps*, una técnica de visualización que se basa en la definición de las jerarquías a través de rectángulos anidados, se describe a detalle en la sección 5.2.3, representan la principal visualización en esta categoría, y Jambalaya la principal herramienta que los implementa.
- *Foco+Contexto*.- Esta categoría abarca todas las herramientas que implementan la técnica de *Foco+Contexto* [242, 175, 129], que se ha descrito en la sección 3.2.2.2. TGVizTab [3], es la principal herramienta en esta categoría.
- Representaciones 3D.- Esta categoría se refiere a todas las visualizaciones basadas en el uso de representaciones tridimensionales. OntoSphere [35] representa la principal propuesta en esta categoría, aunque la herramienta de visualización propuesta por Dmitrieva y Verbeek también utiliza representaciones tridimensionales basadas en geometrías hiperbólicas.
- UML.- Esta categoría incluye aquellas herramientas que utilizan diagramas de clases UML para representar tanto la taxonomía del modelo de ontologías, como las relaciones entre los conceptos. TopBraid Composer e IODT son las principales herramientas en esta categoría.
- Mapas Conceptuales.- Esta categoría define a las herramientas que se basan en el uso de los mapas conceptuales para visualizar o modelar Ontologías, entre las que destaca SemanticWorks.

Algunas de estas herramientas se limitan a la representación visual de los modelos ontológicos como parte de un proceso de análisis, como TgVizTab (Figura 2.7), OntoSphere o Jambalaya (Figura 4.6). Algunas otras son más completas, y poseen capacidades de modelado como TopBraidComposer (Figura 2.5), SemanticWorks (Figura 2.4), OntoStudio (Figura 4.3) o NeOn Toolkit (Figura 4.5).

Sin embargo, la representación visual de modelos ontológicos es un área de investigación que todavía puede enriquecerse, ya que permanece el uso de esquemas o técnicas que han quedado completamente obsoletas y no permiten el desarrollo eficiente de los procesos de Ingeniería de Ontologías. A pesar de la diversa gama de herramientas propuestas, ninguna ha logrado conseguir hacer que los procesos de modelado, visualización y análisis de ontologías sean intuitivos y relativamente fáciles de realizar.

Las causas son diversas: la primera y más común en casi todas las herramientas, es la saturación de la información y de elementos visuales en las representaciones, también la poca expresividad visual y algunos otros problemas como la pérdida del contexto de navegación y mecanismos pobres de interacción con el usuario. Cada uno de estos problemas se describen en detalle en los siguiente párrafos.

El problema de la saturación de elementos visuales se debe primeramente a que la estrategia empleada por la mayoría de las herramientas ha sido la de representar toda la información, o al menos la mayor parte de ella en la misma visualización, lo que provoca que la cantidad de elementos visuales crezca de forma considerable, e impide al usuario el poder abstraer conocimiento. El proceso de abstracción en el cerebro humano no puede desarrollarse eficientemente cuando hay demasiada información. El resultado es que el usuario no es capaz de procesar tal cantidad de información, lo que resulta en una ineficiente e inútil representación.

Este problema se presenta en las herramientas que se basan en el uso de grafos dirigidos y diagramas de clase UML, como OntoViz, OntoSphere, OntoStudio, Neon Toolkit, Jambalaya (Figura 4.6), OntoTrack y TopBraid Composer (Figura 2.5) principalmente. También las visualizaciones basadas en mapas conceptuales, como el caso de Semantic Works (Figura 2.4) tienen este problema, y más aún, esta herramienta presenta la desventaja de la redundancia de información, esto es, información duplicada de forma innecesaria, lo que agrava todavía más esta problemática.

La saturación de información se ilustra en la Figura 4.5 con la herramienta NeOn Toolkit, y también en la Figura 2.7 con la herramienta TgVizTab. El elevado número de arcos (relaciones) así como el número de cruces entre estos,

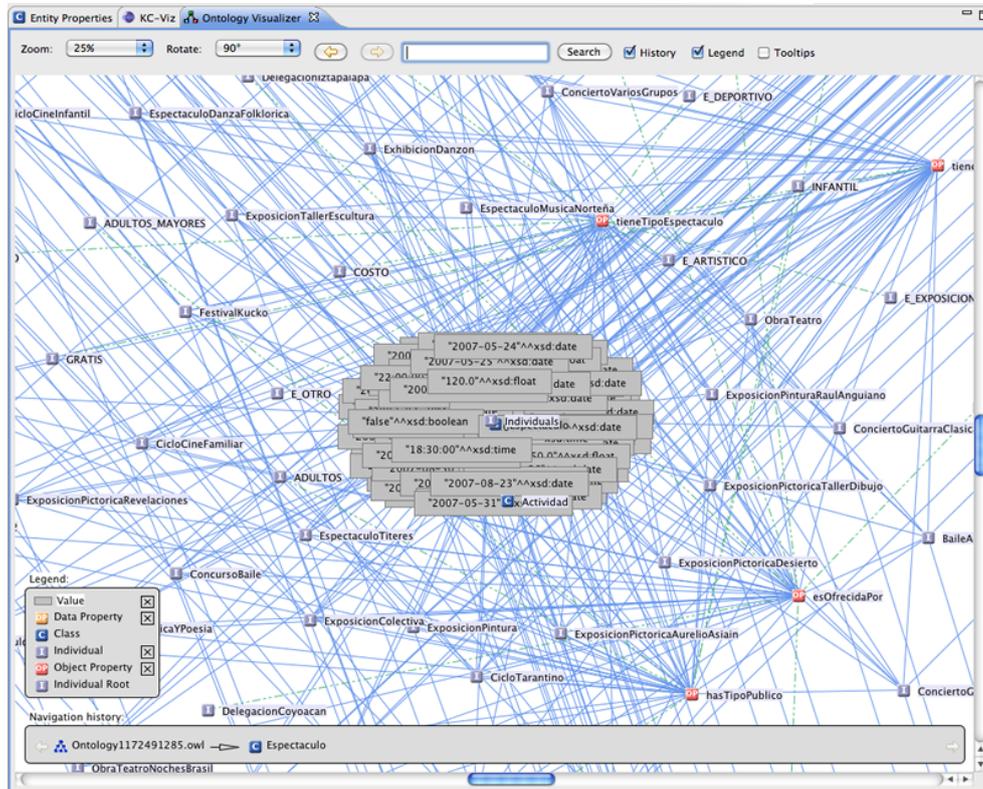


Figura 4.5: La herramienta NeOn Toolkit, en donde la saturación de los elementos no permite el análisis adecuado del modelo ontológico.

hace prácticamente imposible comprender qué nodos se relacionan con qué otros nodos. Es evidente que estas herramientas no permiten un desarrollo eficiente de los procesos de la Ingeniería de Ontologías, ya que la representación visual está saturada de elementos, lo que hace muy difícil al usuario el poder entender el dominio modelado, y en consecuencia, complica en demasía el poder realizar los diversos procesos de creación, gestión, análisis o reutilización.

Otro de los problemas con las herramientas, es la poca o nula expresividad visual. Una de las variables visuales descritas en la sección 3.1.1, es la ubicación, que se refiere a la distribución física de los elementos visuales en el plano dimensional (ejemplificado en la Figura 3.6). Como se puede observar en las Figuras 2.7, 4.5 y 4.6, los elementos carecen de una distribución física en el plano dimensional, lo que dificulta que el usuario pueda crear un mapa mental adecuado. Esta distribución física se refiere a la disposición de los elementos en la pantalla, según algún criterio estructural o geométrico.

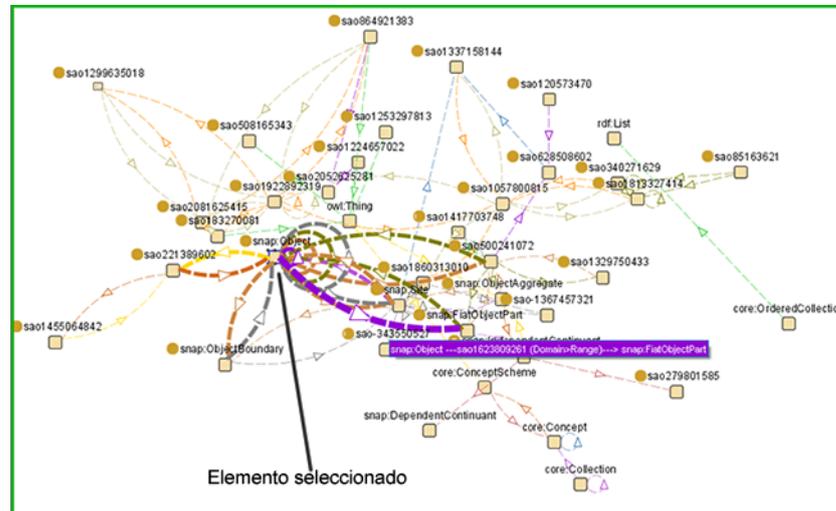


Figura 4.6: Una vista de la herramienta Jambalaya, un *plugin* de Protégé, en donde los elementos visuales no tienen una distribución física sobre el espacio dimensional.

La distribución aleatoria de los elementos, como se muestra en la Figura 4.6, dificulta al usuario, primero, poder localizar un elemento específico en la visualización y, segundo, poder crear un mapa mental de la representación visual; esta situación ha sido descrita en la Figura 3.6 de la sección 3.2.2.4. El problema de la distribución de los elementos visuales se presenta principalmente en las herramientas basadas en grafos dirigidos y en diagramas de clases UML. Es un problema muy relacionado con el problema descrito anteriormente de la saturación de las visualizaciones.

Existen diversas soluciones al problema de la distribución de elementos en un plano bi-dimensional. Una de estas soluciones es la definición de agrupamientos o *clusters*, que consiste en colocar diversos elementos relacionados en una agrupación, lo que evita el uso de aristas para indicar relaciones [13, 228, 206]. Para el caso específico de la visualización de ontologías, el uso de una solución basada en agrupaciones, generaría una visualización con intersecciones entre estas, como se describe en [232].

Un ejemplo de una de estas soluciones basada en agrupaciones es AutoFocus², una herramienta para realizar búsquedas semánticas sobre los documentos almacenados en un ordenador. AutoFocus utiliza una ontología para clasificar los documentos de acuerdo a diversos criterios, como el tipo, la fecha, el tamaño entre otros. También realiza un análisis de los elementos en el equipo

²<http://www.aduna-software.com/>

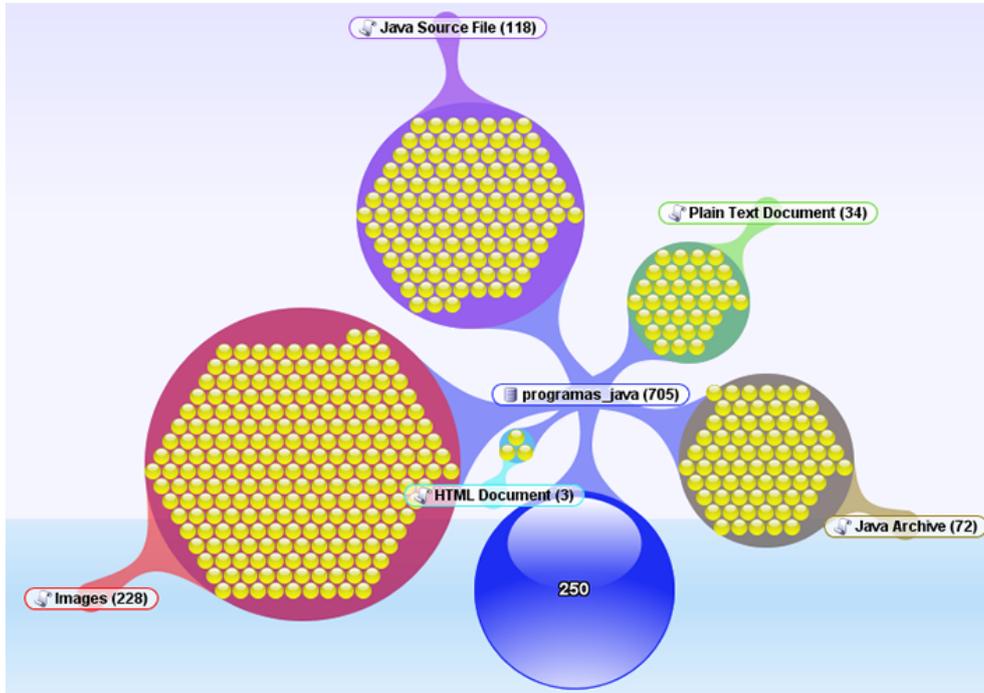


Figura 4.7: AutoFocus utiliza una visualización basada en agrupaciones, para la representación de los resultados de búsquedas semánticas sobre documentos, previamente indexados y clasificados.

del usuario, indexa todos esos documentos y los clasifica. La Figura 4.7 muestra una vista de la herramienta, en la que se representan seis agrupaciones de acuerdo a los tipos de datos. En este ejemplo en específico se representan los documentos HTML, las imágenes, los archivos fuentes de Java, los archivos compilados y archivos en texto plano.

Como se puede apreciar en la Figura 4.7, la representación de elementos relacionados mediante el uso de agrupaciones mejora considerablemente con respecto al uso de grafos, ya que resultan intuitivas para el usuario, las relaciones existentes entre los elementos, sin la necesidad de utilizar aristas que saturan la visualización. Como se puede apreciar en la Figura 4.7, se han representado un total de setecientos cinco elementos, de una forma bastante clara e intuitiva para el usuario, en contraste a la visualización resultante de un grafo con la misma cantidad de elementos.

Otra solución bastante común es la distribución circular también denominada radial, en la que los elementos se colocan formando una circunferencia, o incluso con formas geométricas más complejas como figuras en el plano hiper-

bólico³ [166, 242]. Una solución bastante eficiente a un problema similar, pero en el área de Ingeniería de *Software*, utiliza una distribución radial de los elementos para representar a las clases de un sistema de *software*, y las relaciones entre estas [135].

Un tercer problema se refiere a la pérdida del contexto de navegación. El flujo de navegación sobre diversas visualizaciones define una trayectoria, que el usuario recorre para llegar desde una fase inicial hacia una determinada fase final [175]. Es importante que el usuario sea consciente en todo momento de la fase en la que está ubicado dentro de esta trayectoria, así como las fases que ha realizado, y cuáles más le restan por realizar. En dependencia de la naturaleza de las visualizaciones, el problema de la pérdida del contexto perjudica en un grado mayor o menor.

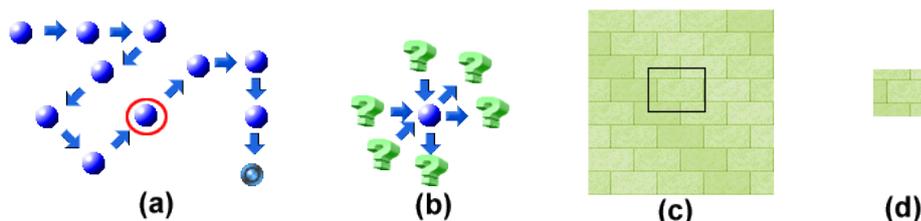


Figura 4.8: (a) Un diagrama de estados con el flujo de navegación completo. (b) Un diagrama de estados sin el flujo de navegación (pérdida del contexto). (c) La selección de un elemento con el contexto completo. (d) La selección del mismo elemento que en (c) pero sin el contexto.

Una analogía para comprender el flujo de navegación sería un diagrama de estados como el que se muestra en la Figura 4.8 (a), en el que el contexto permanece, así como el flujo de navegación completo. En cada uno de los estados siempre se conocen los estados anteriores y los posteriores hasta llegar al estado final (todo el recorrido). Por el contrario, la pérdida del contexto ocasiona que el usuario no sepa exactamente de donde viene y adonde va, como se ilustra con el diagrama de la Figura 4.8 (b).

La pérdida del contexto de navegación provoca que el usuario también pierda el contexto mental dentro del análisis de una representación visual [242, 129]. Cuando un usuario navega a través de diferentes vistas o visualizaciones generalmente sigue un flujo definido, que puede ser desde una vista muy general a una vista muy particular. Si el cambio entre las vistas se produce de manera brusca y repentina, el usuario mentalmente tiene que cambiar la percepción del modelo que intenta procesar, y este cambio puede representar que se pierda la

³El plano hiperbólico es una abstracción matemática, que puede mapearse de forma natural en un plano euclideoano

idea de la navegación, las fases previas a la actual y como regresar a la anterior vista.

La Figura 4.8 (c) muestra un elemento enfocado, en el que se conserva el contexto completo de la visualización, en este ejemplo se puede distinguir claramente una pared de ladrillos. Por otro lado, la Figura 4.8 (d) muestra el mismo elemento seleccionado pero sin el contexto completo, en este ejemplo específico, no es claro lo que representa el detalle en relación al contexto completo.

Un caso muy específico de pérdida del contexto visual, lo presenta Semantic Works, como se muestra en la Figura 2.4. Los diagramas creados por Semantic Works contienen información duplicada. Esta solución de información redundante es considerablemente ineficiente para un esquema de modelado visual, ya que la misma estructura del diagrama puede duplicarse o representarse una cantidad indefinida de veces, lo que resulta en una pérdida del contexto visual y, en consecuencia, también mental. Esta herramienta permite a los usuarios navegar entre los componentes visuales, al expandir u ocultar dichos elementos. Al expandir el mismo elemento más de una vez, se producen diversos caminos iguales por los que el usuario puede transitar, en una especie de laberinto de opciones idénticas.

Otro caso de pérdida del contexto se presenta con las herramientas que utilizan técnicas de *zoom*. Un claro ejemplo es OntoTrack, una herramienta que despliega una visualización saturada de información, y que implementa una técnica de *zoom* para enfocar una parte de la visualización. La solución adoptada por esta herramienta es colocar una visualización pequeña con el contexto global en la esquina superior de la pantalla. Esta solución intenta corregir el problema de la pérdida del contexto, sin embargo, no es una solución eficiente, ya que representa dos escenarios diferentes que el usuario tiene que analizar. Por un lado, una jerarquía muy pequeña que no es claramente distinguible y, por otro lado, una parte de esa jerarquía magnificada, que puede no llegar a ser tan clara para ontologías con algunos centenares o millares de elementos. Esta solución funciona para jerarquías pequeñas, pero para grandes jerarquías presenta la limitante del reducido tamaño de la ventana con el contexto.

Otro caso de pérdida del contexto se presenta con el uso de listas jerárquicas para representar la taxonomía de los conceptos de una ontología. La gran mayoría de las herramientas han adoptado a las listas como la forma de representación de las jerarquías de clases.

La Figura 4.9 muestra el uso de estas listas jerárquicas en tres diferentes herramientas, Protégé, OntoStudio y SWOOP. Claramente se puede apreciar que, para alguna clase específica en la jerarquía, no se tiene el contexto de

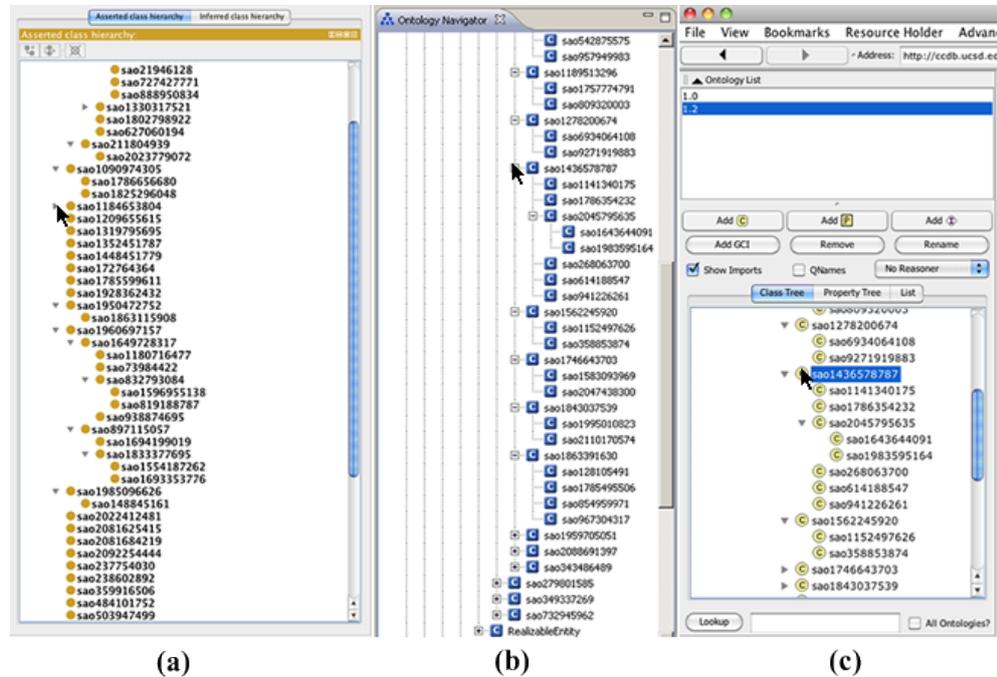


Figura 4.9: Tres diferentes herramientas que utilizan listas jerárquicas: (a)Protégé, (b) OntoStudio y (c) SWOOP. Esta técnica falla principalmente porque no despliega la jerarquía completa de superclases, para una clase que se haya seleccionado.

cuáles son sus superclases; esto es un aspecto fundamental cuando se representa una jerarquía. El principal objetivo de una representación jerárquica, es sin duda el poder conocer la jerarquía de cualquier concepto. Como se muestra en las Figuras 4.9 (a), 4.9 (b) y 4.9 (c), el campo visual que se tiene no es suficiente para la representación de toda la jerarquía de un cierto concepto o clase. Por otro lado, la búsqueda de alguna clase en específico también resulta bastante compleja, ya que el modelo de navegación sobre las listas para grandes jerarquías, resulta confuso.

Existen diversas soluciones al problema de la representación de jerarquías, especialmente aquellas de gran tamaño. Las primeras soluciones propuestas, estaban basadas principalmente en el uso de grafos, distribuciones radiales o circulares [178, 9]. RINGS [257] es una propuesta que se basa en el uso de geometría hiperbólica para visualizar grandes jerarquías. Esta técnica aprovecha mejor el espacio visual, en comparación con otras propuestas como el Hyperbolic Browser [177]. También implementa mecanismos de interacción como animaciones, una escala de colores para indicar profundidad o resaltar elemen-

tos seleccionados. El principal problema de este tipo de soluciones basadas en grafos, es que para jerarquías muy profundas, los niveles inferiores no son fácilmente navegables, ni su representación es intuitiva. Para corregir esta situación, se tienen que implementar robustos mecanismos de interacción con el usuario.

Una solución mejor que los grafos, es el uso de una técnica visual llamada árboles de jerarquías (*trees*). Su característica principal es que conservan el modelo de representación basada en el uso de aristas para conectar los nodos, pero con una distribución física (*layout*) que permite representar de forma muy clara la jerarquía [260]. Los árboles de jerarquías, representan una opción considerablemente mejor para la representación de taxonomías que las listas y los grafos, ya que la distribución de los elementos mejora la percepción visual de los usuarios, y facilita el entendimiento de los niveles en la estructura.

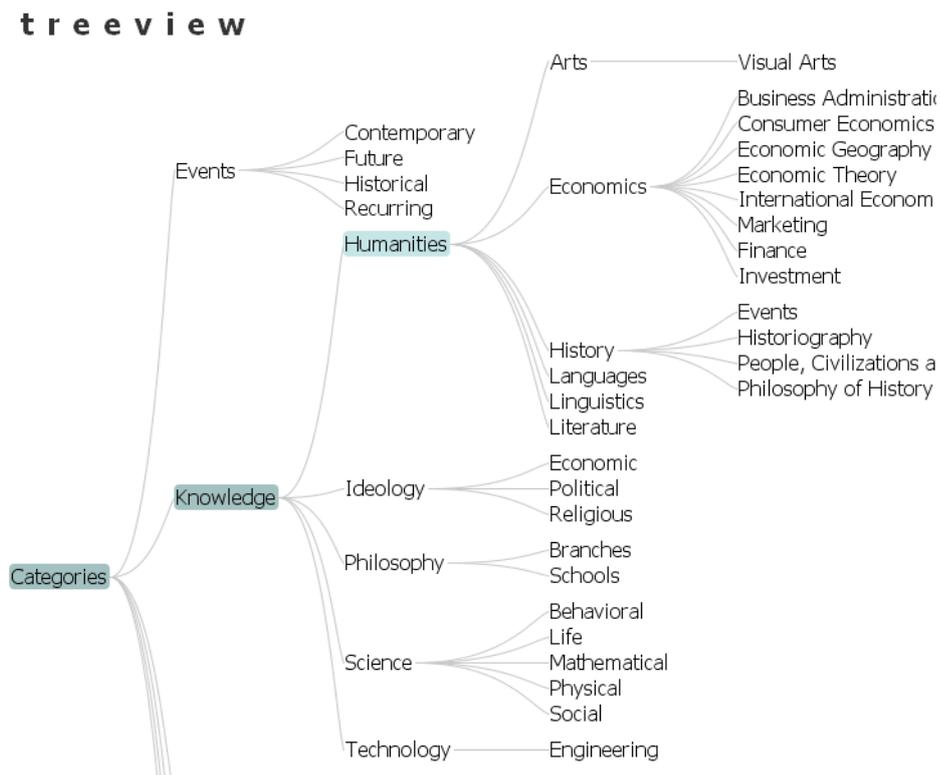


Figura 4.10: Los árboles de jerarquías representan una solución exitosa a la problemática de la representación de grandes jerarquías.

La Figura 4.10 muestra un ejemplo de un árbol de jerarquías⁴, en el que se

⁴<http://prefuse.org/gallery/treeview/>

clasifican diferentes áreas de conocimiento. Una de las principales características y ventajas de esta técnica sobre las listas, es que requiere la asignación de una mayor cantidad del espacio visual disponible. Este requisito de demanda de espacio físico tiene que compensarse con la implementación de capacidades de interacción con el usuario bastante robustas, principalmente para mostrar u ocultar elementos y ramas completas de la jerarquía.

Una de las aplicaciones de los árboles de jerarquías es el proyecto Space-Tree [220, 180], una aplicación basada en árboles para representar grandes jerarquías, con mejores resultados que el uso de grafos [223].

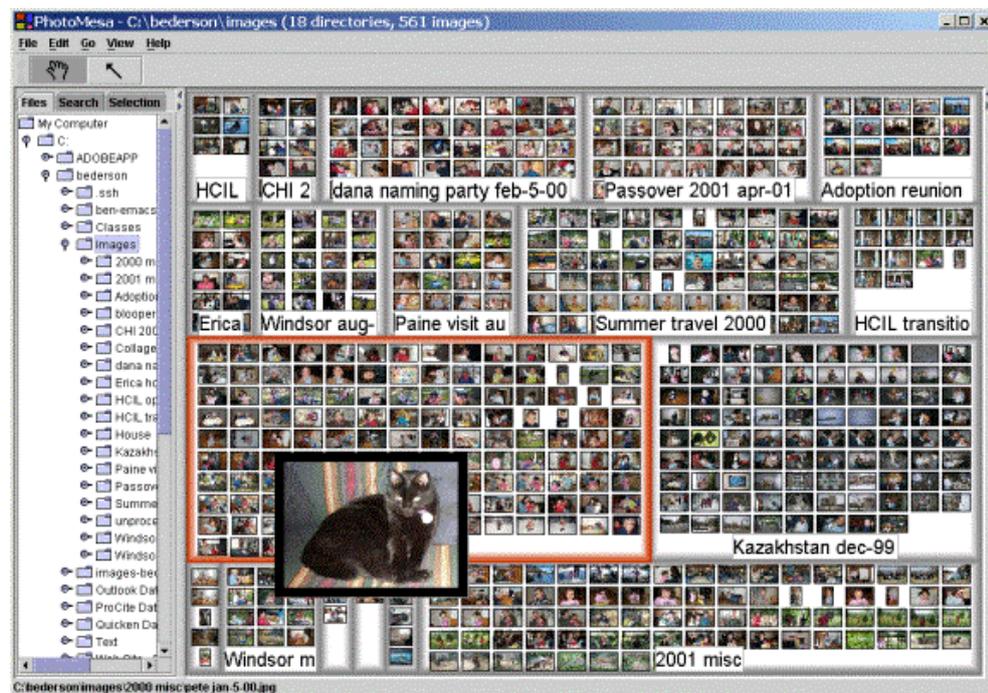


Figura 4.11: Una técnica visual para la representación de grandes jerarquías es el *treemap*, en este caso se ilustra una vista de la aplicación *PhotoMesa*.

Otra solución al problema de representación de grandes jerarquías es el uso de una técnica visual conocida como *treemap* [24]. Los *treemaps* son una técnica de visualización que aprovecha todo el espacio visual disponible en el plano dimensional [151]. Se basan en la representación de mapas rectangulares en dos dimensiones, en donde los niveles inferiores se representan como elementos internos dentro de los mapas de nivel superior, y en los que básicamente se tienen dos variables: por un lado, el área de los mapas rectangulares y por otro lado el color de estos. La técnica de *treemap* divide el área visual en una

secuencia anidada de rectángulos, cuyas áreas corresponden a un atributo del conjunto de datos, en donde se combinan aspectos de los diagramas de Venn-Euler, con los típicos diagramas de tarta [38].

La Figura 4.11 muestra una vista general de un *treemap*, para una aplicación que muestra fotografías, llamada *PhotoMesa*⁵, en donde los niveles superiores de la jerarquía representan la clasificación de las fotografías de acuerdo al lugar o la fecha en la que fueron tomadas, mientras que los niveles inferiores, en este caso en particular, son las propias fotografías, pero en otros casos resultarán ser un mapa rectangular simplemente con algún color de fondo.

Los *treemaps* han demostrado ser una solución muy eficiente al problema de la representación de grandes jerarquías [24], ya que se han empleado en diferentes áreas de investigación que requieren la visualización de grandes cantidades de información, como la bioinformática [18], las finanzas [6], la genética [17], entre otras.

SequoiaView⁶ es otra propuesta que utiliza los llamados *cushion treemaps* [279, 38], para mostrar una vista del disco duro, que incluyen todas las unidades y los archivos. Los *cushion treemaps* difieren de la propuesta original, en que los elementos del *treemap* tienen una variante en cuanto a la tonalidad, iluminación o la textura. Esta variación permite representar más información en la vista, ya que aparte del color se agrega una nueva variable que es la textura.

Otra de la problemática con las herramientas, es la falta de mecanismos de interacción robustos. Un ejemplo de esta falta de mecanismos, se presenta con el manejo de focos y contextos para resaltar determinados elementos del resto, como se ha explicado en la sección 3.2.2.2. Un ejemplo de esta problemática se ilustra en la Figura 4.6 con la herramienta Jambalaya, en donde hay una selección de un elemento visual, y el manejo del foco y el contexto es deficiente. En consecuencia, no resulta suficientemente claro, cuál es el elemento seleccionado, ni tampoco cuáles son los elementos relacionados con este elemento seleccionado.

Esta visualización de la Figura 4.6 mejoraría considerablemente si se aplicara la técnica de opacidad sobre los elementos que no se pretenden resaltar. El difuminado o la opacidad para la definición de planos o contextos es otra de las variables visuales de la que no se han aprovechado las herramientas. El uso de esta técnica permite definir diversos planos focales, lo que mejora considerablemente la interacción y el proceso cognitivo del usuario durante una fase de análisis. Esta problemática se presenta de la misma forma con todas

⁵<http://www.cs.umd.edu/hcil/photomesa/>

⁶http://w3.win.tue.nl/nl/onderzoek/onderzoek_informatica/visualization/sequoiaview/

las herramientas que se basan en el uso de grafos.

Otro ejemplo de la falta de mecanismos de interacción se presenta con las herramientas que utilizan todo el espacio posible de visualización, especialmente con los *treemaps*. Como se ha mencionado previamente, los *treemaps* representan una solución eficiente para la representación de grandes jerarquías [17, 151], especialmente en amplitud. Sin embargo, la propuesta original de los *treemaps* resulta menos eficiente para jerarquías grandes en profundidad, ya que los niveles internos pueden llegar a ser casi imperceptibles durante una fase de análisis.

Una solución a esta problemática se describe en [236], y consiste en la propuesta de *treemaps* con pocos niveles de profundidad (tres o cuatro máximo), y la representación de los restantes niveles internos con un color de densidad, que varía en tonalidad de colores claros a oscuros, en donde los colores más oscuros indican mayor presencia de elementos en los niveles internos. Esta solución de densidad proporciona al usuario una idea de la población en los niveles inferiores, sin embargo, no permite un análisis más detallado de estos niveles.

Blanch y Lécolinet [30, 31] han propuesto proveer a los *treemaps* con la capacidad para implementar la técnica de *zoom*, sobre los elementos que pueden llegar a ser imperceptibles dentro de la jerarquía, aquellos localizados en niveles inferiores del *treemap*. Esta propuesta ofrece una buena solución al problema, excepto por el inconveniente de que al desarrollar la técnica de *zoom*, el contexto general se pierde, situación que no es deseable en el caso de las ontologías.

El problema de interacción en estos casos se debe principalmente a la naturaleza estática de cierto tipo de visualizaciones, como el caso de los *treemaps*, los grafos dirigidos, los mapas conceptuales o los diagramas de clase UML. Algunas soluciones se han implementado para hacer estas visualizaciones menos estáticas, como en el caso de OntoTrack y Semantic Works, que tienen implementada una funcionalidad para expandir y contraer ramas o caminos completos del gráfico en el caso de OntoTrack, o del mapa conceptual en el caso de Semantic Works. Esta interacción intenta resolver de alguna manera, el problema de la saturación visual de elementos en las visualizaciones, sin llegar a ser una solución suficiente.

Las herramientas basadas en el uso de técnicas de *zoom*, implementan mecanismos de interacción, al igual que las herramientas que utilizan listas. Sin embargo, ambos grupos de herramientas comparten la problemática de la pérdida del contexto, como se ha descrito previamente.

Sin duda alguna, y como consecuencia de su propia naturaleza, las representaciones tridimensionales son las visualizaciones que implementan los

mejores mecanismos de interacción con el usuario. En contraste, la escalabilidad representa su principal desventaja, ya que la demanda de recursos de procesamiento y memoria afecta el desempeño de estas herramientas cuando los modelos ontológicos son grandes. Estas representaciones también presentan el inconveniente de la saturación de elementos para representaciones medianamente grandes de elementos visuales, así como el problema de la oclusión de los elementos, según el punto de vista elegido.

La Tabla 4.1 hace un resumen de las herramientas que implementan técnicas de visualización, así como la principal problemática, como se ha descrito en párrafos que anteceden.

Herramienta	Clasificación	Problemática
SemanticWorks	mapas conceptuales	redundancia de información y pérdida del contexto de navegación
OntoStudio	listas jerárquicas y grafos dirigidos	saturación de información y pérdida del contexto de navegación
TopBraid Composer	listas jerárquicas, grafos dirigidos y UML	saturación de información y pérdida del contexto de navegación
IODT	UML	saturación de información
SWOOP	técnicas de <i>zoom</i> y listas jerárquicas	pérdida del contexto y pobre interacción
NeOn Toolkit	listas jerárquicas y grafos dirigidos	saturación de información y pérdida del contexto de navegación
OntoEdit	listas jerárquicas	pérdida del contexto de navegación
OntoTrack	grafos dirigidos y técnicas de <i>zoom</i>	saturación de información
Protégé	grafos dirigidos y listas jerárquicas	saturación de información y pérdida del contexto de navegación
OntoSphere	tridimensional	escalabilidad para representar grandes modelos ontológicos
Dmitrieva	grafos y tridimensional	escalabilidad y saturación de información

Tabla: 4.1: Resumen de la problemática de las herramientas que implementan técnicas de visualización, de acuerdo a la clasificación establecida.

4.5. Conclusiones

Las metodologías representan un mecanismo fundamental para desarrollar los procesos de gestión del ciclo de vida de las ontologías. De todas las propuestas analizadas destacan dos de ellas principalmente: por un lado Methontology, y DOGMA. Methontology propone una serie de flujos de trabajo y actividades basadas en el modelo iterativo de desarrollo de *software*, pero aplicadas a los procesos para la creación de ontologías. También define las tareas básicas, principalmente para la creación de ontologías, por lo que ha servido como base para un conjunto de otras metodologías que han propuesto algunas innovaciones con respecto a esta.

Por otro lado, DOGMA es una propuesta de metodología enfocada completamente a fomentar la reutilización de ontologías. Define la forma en que los diseños ontológicos pueden resultar altamente reutilizables. Principalmente, mediante la definición de diversos submodelos, que parten de niveles más altos de abstracción hacia niveles más específicos del dominio de aplicación que se modela. Esta estrategia permite que los submodelos más abstractos puedan reutilizarse en diferentes dominios de aplicación. La reutilización de ontologías representa uno de los hitos principales de la Ingeniería de Ontologías, por lo que todos los diseños ontológicos deberían cumplir el precepto de ser altamente reutilizables.

Como se ha descrito previamente, no existe una metodología estándar para los procesos de la Ingeniería de Ontologías, sino más bien un conjunto de diversas propuestas, de las que se tiene poco soporte para el desarrollo de las actividades definidas en estas. Como resultado de esta falta de procesos estandarizados se tienen diseños de modelos ontológicos que resultan en la creación de ontologías con baja calidad, difíciles de actualizar, reutilizar e incluso con inconsistencias. En consecuencia, los diversos procesos involucrados en el ciclo de vida de las ontologías, se complican en demasía. Esto se debe a que, la gran mayoría de las veces, los diseños ontológicos son resultado de la propia experiencia, capacidad y conocimiento del diseñador, lo que no siempre resulta ser la mejor opción.

Para lograr sistemas altamente reutilizables, el lenguaje de implementación juega un papel muy importante, ya que los nuevos diseños de sistemas semánticos, tienen la tendencia a utilizar las más recientes especificaciones de lenguajes. Es justo en este aspecto en donde OWL toma ventaja sobre el resto de los lenguajes, ya que agrupa todas las ventajas de los lenguajes antecesores, y además representa un esfuerzo por estandarizar un lenguaje para las aplicaciones de Web Semántica y los sistemas de representación de conocimiento. Al ser una recomendación, su uso debería adoptarse como un estándar para

subsecuentes desarrollos de sistemas ontológicos.

Desde el punto de vista del análisis de un modelo ontológico es importante contar con métricas basadas en la semántica de las ontologías, y no solo en la estructura. Estas métricas requieren una interpretación por parte del usuario, ya que representan más que una simple medida o la estimación de algún valor. También resultaría muy útil el diseño de representaciones visuales que apoyaran al usuario en el análisis y, sobre todo, en la interpretación semántica de este tipo de métricas. Una fase de análisis bien implementada resulta crucial para la reutilización, ya que facilita la evaluación de los modelos ontológicos, e incide directamente sobre la toma de decisiones.

Un aspecto fundamental que ha abordado este capítulo es la problemática de las herramientas de creación, modelado y visualización de ontologías. La gran mayoría de las herramientas han tomado como base Protégé para el diseño de sus visualizaciones, por lo que estas comparten prácticamente la misma problemática. Los editores de ontologías basados en el uso de los navegadores web, no implementan mecanismos de interacción visual, por lo que no se consideran como herramientas robustas para el desarrollo de la Ingeniería de Ontologías, desde la perspectiva del usuario como centro de los procesos de creación, gestión, análisis o reutilización.

En general, se han identificado cuatro problemas principales: la saturación de información, la pobre expresividad visual, la pérdida del contexto de navegación y la falta de mecanismos de interacción con el usuario. De acuerdo a la clasificación que se ha hecho de las herramientas, la problemática detectada más recurrentemente es la saturación de información y la pérdida del contexto. Los mecanismos de interacción con el usuario también suelen ser muy escasos en las herramientas actuales. Estos mecanismos representan un aspecto crucial, ya que permiten al usuario mejorar considerablemente su proceso de abstracción sobre los modelos ontológicos, lo que representa el principal objetivo de las herramientas centradas en el usuario, como son las herramientas de modelado visual.

Capítulo 5

Modelo propuesto para la gestión de ontologías

El capítulo 4 ha discutido el estado del arte y también ha descrito la problemática más común para el desarrollo de los procesos involucrados en el ciclo de vida de las ontologías. Por un lado, la falta de una estandarización en cuanto a las metodologías que soportan estos procesos, lo que provoca que la calidad de los diseños de los modelos ontológicos dependa en gran medida de la experiencia de los diseñadores, ya que no hay procesos, flujos de trabajo y actividades claramente definidas. La solución adoptada por los diseñadores se basa directamente en el conocimiento de esta diversidad de metodologías, y en su propia capacidad para discernir las actividades a desarrollar en cada uno de los procesos. Esto trae como principal consecuencia, diseños poco reutilizables.

Por otro lado, también se ha descrito la problemática involucrada con las herramientas de visualización y modelado de ontologías. Básicamente, se han identificado cuatro problemas principales: la saturación de información, la pobre expresividad visual, la pérdida del contexto de navegación y la falta de mecanismos de interacción robustos con el usuario.

Este capítulo se enfoca en la propuesta de soluciones específicas a toda esta problemática. La primera parte describe la propuesta de diversos flujos de trabajo y actividades definidas con el objetivo de mejorar la eficiencia de los procesos del ciclo de vida de las ontologías (creación, gestión, análisis y reutilización).

Estos flujos de trabajo se han definido tomando en consideración las principales metodologías existentes, y se han planteado desde la perspectiva de los procesos de análisis desarrollados en el área de la Analítica Visual. La filosofía

detrás de los procesos en Analítica Visual es que todas las herramientas, actividades y flujos de trabajo se definen bajo el principio de que el usuario es el elemento principal, por lo que se centran completamente en el aprovechamiento de sus capacidades cognitivas.

La segunda parte del capítulo describe las soluciones que soportan todas las tareas definidas en estos flujos de trabajo, basadas principalmente en un enfoque de Analítica Visual e InfoVis. Estas soluciones son diversas técnicas de visualización e interacción con el usuario, que corrigen la problemática de las herramientas, descrita previamente. Finalmente, se discuten las conclusiones del capítulo.

5.1. Flujos de trabajo para el desarrollo de la Ingeniería de Ontologías

Para la definición de los flujos de trabajo, se han considerado principalmente dos metodologías: DOGMA y Methontology. Si se toman en consideración estas propuestas, un enfoque resultante, que considera los aspectos positivos de ambas, sería el planteamiento de un proceso que permita desarrollar el ciclo de vida de las ontologías, basado en una serie de iteraciones, en donde en cada iteración se genera un submodelo del modelo ontológico final. Las primeras iteraciones, deberían generar submodelos con un alto nivel de abstracción, para permitir que las iteraciones posteriores generen modelos ontológicos más específicos.

Esta propuesta se basa en la filosofía detrás de DOGMA, que consiste en la definición de diversos modelos ontológicos, en donde se comienza desde niveles muy generales de abstracción, hasta niveles muy específicos para ciertas aplicaciones. La propuesta en concreto radica en seguir el enfoque del diseño iterativo, que comienza con la definición de la ontología base según el esquema de la metodología DOGMA, y en cada iteración definir los submodelos ontológicos más específicos para el dominio de aplicación que se pretende. Esta estrategia debe permitir aprovechar las ventajas de las dos propuestas. Principalmente, se basa en la definición de un esquema general iterativo para la construcción de los modelos ontológicos, con la finalidad de desarrollar modelos reutilizables.

Este enfoque debe facilitar la reutilización de los modelos, puesto que para los nuevos modelos no se tendrían que realizar las primeras iteraciones, sino únicamente las últimas, que son las más específicas para determinada aplicación. De esta forma, el desarrollo de los modelos ontológicos potencia el proceso de

reutilización y también soporta el desarrollo del ciclo de vida completo de las ontologías de una forma eficiente.

Los flujos de trabajo para el desarrollo de la Ingeniería de Ontologías, definen las actividades a realizar por el usuario para cada uno de los procesos de creación, gestión, análisis y reutilización. Cada uno de estos flujos de trabajo y actividades, se describen en detalle en las siguientes secciones.

5.1.1. Creación de ontologías

El proceso de creación de ontologías implica el desarrollo de las actividades descritas en la sección 2.3.1. La primera parte corresponde a una fase de abstracción del dominio del problema y el planteamiento de las soluciones. Después comienza la propia fase de implementación del modelo ontológico, que comienza por la definición de la taxonomía de las clases, seguida por la definición de las propiedades intrínsecas a las clases, después se crean las propiedades objeto que representan las relaciones entre las clases y, finalmente, se definen las instancias, que se corresponden con valores específicos de las clases.

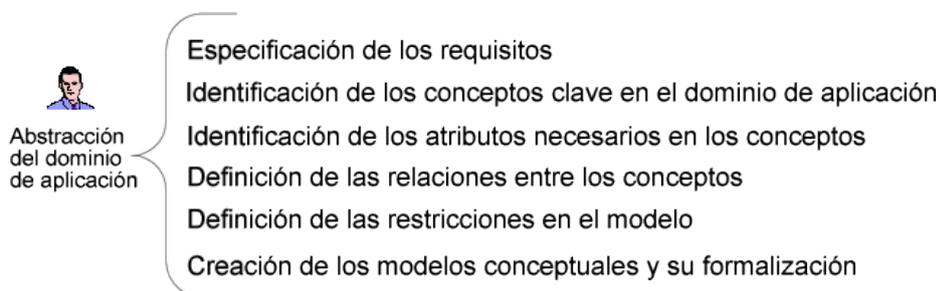


Figura 5.1: Las principales tareas para desarrollar la fase de abstracción del dominio del problema, durante el proceso de creación de ontologías. Estas tareas se desarrollan completamente por el diseñador del modelo ontológico.

El proceso de creación se ilustra en el esquema de la Figura 1.3, en donde la primera fase, denominada abstracción del dominio, se define básicamente para poder determinar aspectos como el uso, análisis y alcance del dominio que cubrirá la ontología. Esta fase también incluye el planteamiento de las soluciones, así como el diseño del modelo ontológico. Se pretende que estas soluciones consideren el principio de reutilización, y que sigan el proceso iterativo propuesto por la metodología Methontology (descrita en la sección 2.3.1), en el que se comienza con la definición de una ontología base muy general, y en cada iteración se construye un submodelo del dominio más específico.

Más detalladamente, las tareas básicas a desarrollar durante la primera fase de abstracción del dominio del problema, se ejemplifican en el esquema de la Figura 5.1, que a su vez provienen principalmente del esquema propuesto por Methontology e ilustrado en la Figura 1.2.

Se pretende que esta propuesta siga el enfoque de la metodología DOGMA [146, 148] (descrita en la sección 2.3.1), en donde el planteamiento de la solución debe definir primeramente una ontología base con los conceptos más generales, sin especificar demasiado en el dominio de la aplicación. La última actividad del esquema de la Figura 5.1, que corresponde a la creación de un modelo conceptual y su formalización, tiene como principal objetivo, definir el número de submodelos u ontologías (que debe comenzar con la ontología base) que formarán el modelo general, así como el número de iteraciones.

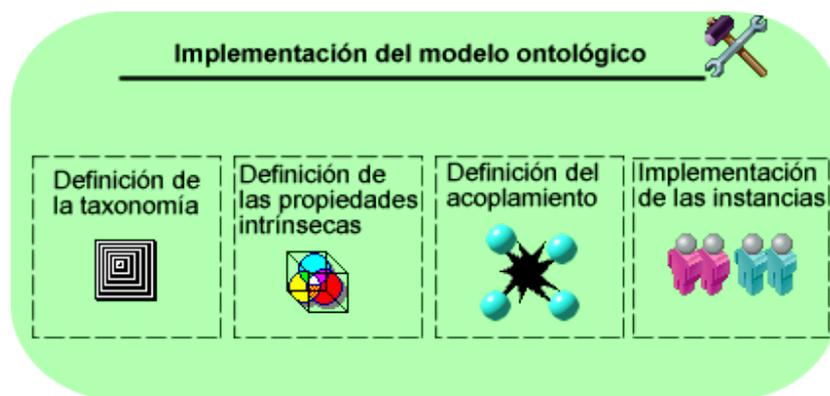


Figura 5.2: Las principales tareas para la fase de implementación de ontologías, dentro de los procesos de creación, gestión y reutilización. Comienza con la definición de la taxonomía de clases, la definición de las propiedades intrínsecas, las relaciones entre clases, y finalmente, la implementación de instancias que definen la base de conocimiento.

Una posible solución para la definición del número de iteraciones sería el planteamiento de la solución como diversos subdominios en lugar de un único dominio general. Cada iteración corresponde con la implementación de uno de los subdominios definidos y, a su vez, cada subdominio debería ser más específico de la aplicación. Esta estrategia permite definir submodelos ontológicos orientados a favorecer en gran medida la reutilización.

Una vez definido el planteamiento de las soluciones, el número de iteraciones y la especificación de la ontología base, comienza la propia fase de implementación, que se describe en la Figura 5.2, y se desarrolla en un número de iteraciones o ciclos definidos durante la fase de abstracción. La primera de

las actividades en esta fase es la definición de la taxonomía de conceptos de la ontología. Para el caso de la primera iteración, la taxonomía debería incluir conceptos suficientemente generales, mientras que para las subsecuentes iteraciones, las taxonomías deberían definir conceptos más específicos. Esta taxonomía implica la definición de diversos niveles de clases, organizadas en una jerarquía.

La segunda actividad del proceso de implementación es la definición de las propiedades intrínsecas a las clases. Estas propiedades, llamadas *datatype properties*, definen la estructura interna que tienen las clases del modelo ontológico. Los tipos de datos más comunes son valores numéricos enteros, flotantes, cadenas de caracteres, valores booleanos, de fecha o de hora.

La siguiente actividad del proceso de implementación es la definición del acoplamiento entre las clases. Este acoplamiento está definido por las propiedades objeto, que relacionan clases en el dominio con clases en el rango. Estas clases, relacionadas entre sí a través de una o varias propiedades objeto, se consideran como clases acopladas.

Finalmente, se crean las instancias de las clases, que generan la base de conocimiento. Estas instancias representan objetos con valores específicos para cada una de las propiedades, lo que garantiza la unicidad de cada instancia. Como han recomendado algunas metodologías como Ontometric, las instancias deberían ubicarse mayormente en los niveles inferiores de la taxonomía, ya que los niveles superiores definen conceptos en un nivel de abstracción mayor, y las instancias se pretende que sean muy específicas de la base de conocimiento.

Otra tarea descrita por algunos autores como Uschold y King [263] es la documentación de las ontologías, un aspecto importante para fomentar la reutilización. No existe mucho trabajo sobre documentación de ontologías, por lo que es un campo en el que se pueden realizar aportaciones importantes, aunque este apartado está fuera del alcance de este trabajo de tesis.

5.1.2. Gestión de ontologías

La gestión de ontologías se refiere al proceso de actualización de los modelos ontológicos previamente creados. Este proceso puede incluir la creación o modificación de ciertos valores de las instancias, o incluso, cambios más importantes dentro de la estructura propia de la ontología, como las clases o las propiedades.

El proceso de gestión de un modelo ontológico se describe en el esquema de la Figura 5.3. Básicamente comienza con un análisis del modelo ontológico

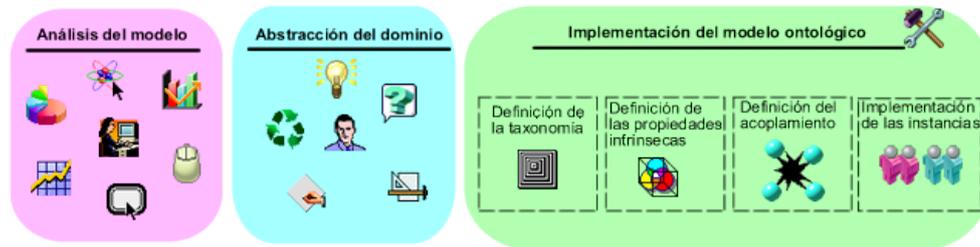


Figura 5.3: El proceso de gestión de ontologías comienza con una fase de análisis del modelo ontológico que se quiere actualizar, con el objetivo de hacer un diagnóstico sobre los cambios exactos que se requieren.

que se pretende modificar. Este análisis se basa en el proceso de análisis y exploración de datos descrito en el diagrama 3.1 en la sección 3.1.1, basado en los flujos de trabajo definidos en el área de Analítica Visual. En este caso, los datos almacenados corresponden al modelo ontológico existente, que se procesa como entrada a las herramientas de visualización, que permiten al usuario el análisis del modelo.

El análisis del modelo ontológico existente es una actividad crucial para su gestión, ya que permite primero, conocer su estado actual y, segundo, continuar con la fase de abstracción y el planteamiento de las nuevas soluciones. Las herramientas de visualización soportan, por un lado, el análisis de los datos mediante la interacción con el usuario y, por otro lado, permiten el modelado de las nuevas ontologías, esto es, la definición o la modificación del esquema actual, o de las instancias de la base de conocimiento. Estas visualizaciones están presentes tanto en la fase de análisis como de implementación o modelado. Este proceso se describe en detalle en la sección 5.1.3.

Una vez realizado el análisis del actual modelo, la gestión continúa con las fases de abstracción e implementación, como se ha descrito en la sección 5.1.1, para los nuevos requerimientos del modelo ontológico.

5.1.3. Análisis de un modelo ontológico

El análisis de una ontología es un proceso crucial cuyo objetivo principal es la obtención de conocimiento sobre el modelo de datos que se analiza. En base al conocimiento obtenido sobre el modelo, se podrán tomar decisiones. El análisis generalmente se desarrolla, ya sea para gestionar un modelo ontológico, o para realizar un proceso de reutilización. Este proceso debe estar completamente soportado por herramientas que implementan técnicas de visualización

e interacción con el usuario. Estas técnicas deben estar diseñadas para utilizar en la mayor medida posible, la capacidad cognitiva del usuario para analizar un modelo de datos semántico.

El proceso de análisis sigue el enfoque definido en el área de la Analítica Visual que se muestra en la Figura 3.1 de la sección 3.1.1. Primero, se desarrolla una fase de preprocesamiento sobre el modelo ontológico, que puede estar definido en un archivo de texto o en una base de datos. Este preprocesamiento permite cargar el modelo y construir las estructuras de datos que alimentan a las visualizaciones. Una vez que las visualizaciones han sido provistas de estas estructuras de datos, las representaciones visuales permiten al usuario comenzar el proceso de análisis del modelo.

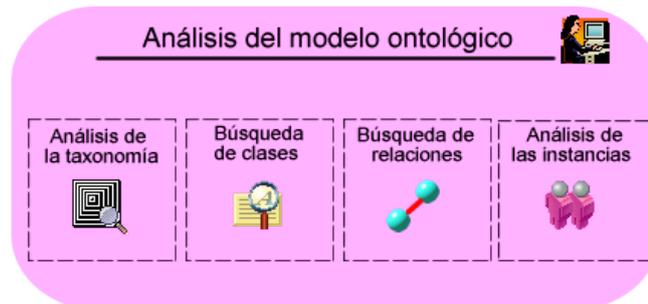


Figura 5.4: El proceso de análisis de ontologías define cuatro actividades principales: el análisis de la taxonomía de los conceptos, la búsqueda específica de clases en el modelo, la búsqueda de relaciones y el análisis de las instancias.

La Figura 5.4 muestra el esquema con las principales actividades definidas para el proceso de análisis de ontologías. Básicamente son cuatro tareas principales, una muy general y las tres restantes más específicas. La primera y más general es el análisis de la taxonomía, ya que implica recorrer la jerarquía de las clases para hacer una evaluación general de los conceptos definidos en el modelo ontológico. El análisis se vuelve más especializado cuando el usuario está interesado en un concepto en particular, es en este momento, cuando se realiza una búsqueda detallada sobre dicha clase. Una vez encontrada la clase buscada, el usuario podría estar interesado en inspeccionar sus propiedades intrínsecas.

Otra actividad en el proceso de análisis es la búsqueda de relaciones específicas entre los conceptos. El análisis de estas relaciones puede darse desde dos perspectivas principalmente: primero, ya sea analizar directamente la propiedad como el elemento principal; segundo, el análisis de la propiedad pero desde las clases que pertenecen a su dominio o a su rango. Idealmente, una herra-

mienta de análisis debería permitir al usuario analizar desde ambos enfoques.

Otra de las actividades básicas para el proceso de análisis se relaciona con las instancias de clase. El análisis de las instancias de clase, debe permitir al usuario conocer los valores internos para cada una de sus propiedades. Esta actividad generalmente comienza con la selección de la clase a partir de la cual se quieren analizar las instancias, de tal forma que el flujo comienza a partir de la clase hacia las instancias de esta.

5.1.4. Métricas de evaluación de ontologías

En esta propuesta de tesis se ha definido formalmente un conjunto de métricas que miden el acoplamiento interno de las clases en una ontología [88]. Estas métricas, se basan en la propuesta de métricas CBO de Chidamber y Kemerer [45] (descritas en la sección 4.2), pero adaptadas a las ontologías. Por similitud, estas métricas se han definido como CBE [84] (acrónimo de *Coupling Between Entities*), y a diferencia de las métricas de acoplamiento definidas por Orme et al. (descritas en la sección 2.3.3) que son globales a nivel de la ontología, las métricas CBE son específicas y se calculan sobre cada una de las clases de forma individual.

Dado que el acoplamiento entre las clases se da a través de las propiedades objeto, la dirección de la propiedad juega un papel importante para poder interpretar la relación de acoplamiento. En términos semánticos, no es lo mismo que una clase pertenezca al dominio de una propiedad, a que pertenezca al rango. La dirección lógica de la propiedad se considera en el sentido desde el dominio hacia el rango, como se ha descrito en la sección 2.1.

La Figura 5.5 ilustra la definición de las métricas de acoplamiento CBE, desde la perspectiva de la clase A, como la clase que se analiza. La primera métrica se denomina *CBE-out*, e indica que la relación es hacia afuera de la clase que se analiza, o lo que es lo mismo, que la clase pertenece al dominio de la propiedad. La segunda métrica denominada *CBE-in*, indica que la relación es en el sentido hacia la clase que se analiza, o dicho de otra forma, que la clase que se analiza pertenece al rango de la propiedad. La tercera métrica denominada *SC* (acrónimo de *self-coupling*), indica que la relación es sobre la misma clase, en otras palabras, que la clase pertenece tanto al dominio como al rango. La cuarta métrica se denomina *CBE-io*, e indica un tipo de relación inversa entre dos clases, esto es, que en un momento una clase puede pertenecer al dominio pero en otro momento, la misma clase puede pertenecer al rango.

Este tipo de relaciones se han definido para el lenguaje OWL, en donde una característica lógica es que una propiedad puede ser inversa de otra, como se

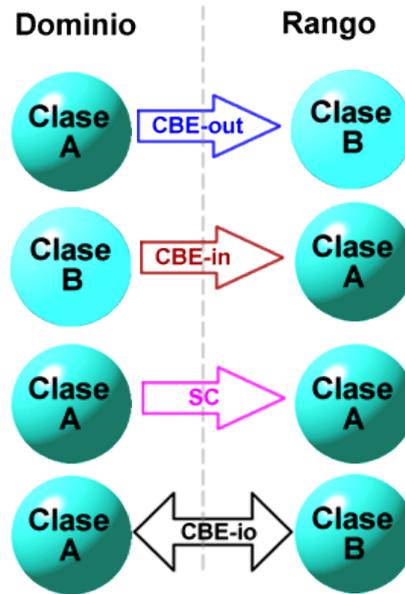


Figura 5.5: Esquema que muestra la definición del acoplamiento en las ontologías OWL. La primera métrica *CBE-out*, indica que la clase se localiza en el dominio de la relación, mientras que la segunda métrica *CBE-in*, indica que la clase se localiza en el rango de la relación. La métrica *SC*, indica que la clase está acoplada consigo misma, tanto en el dominio como en el rango. Finalmente, la métrica *CBE-io* indica que hay una relación de propiedades inversas entre dos clases.

ha descrito en la sección 2.1. Finalmente, la quinta métrica pretende calcular el valor del acoplamiento global de la clase, mediante la suma de los valores calculados para las métricas anteriores. Las métricas de acoplamiento *CBE* se definen a continuación formalmente:

Definiciones:

1. Sea Θ una ontología OWL
2. Sea Φ el conjunto de propiedades , $\Phi \in \Theta$
3. Sea \mathcal{C} el conjunto de clases , $\mathcal{C} \in \Theta$
4. $\exists c \in \mathcal{C}$
5. $\exists \Upsilon \subseteq \Phi$
6. $\exists p y q \in \Upsilon$

Métricas:

1. Se define la métrica *CBE-out* como:

$$\text{si } c \in \text{dominio}(p) \forall p \in \Upsilon \text{ entonces } CBE\text{-out} = | \Upsilon |$$

2. Se define la métrica *CBE-in* como:

$$\text{si } c \in \text{rango}(p) \forall p \in \Upsilon \text{ entonces } CBE\text{-in} = | \Upsilon |$$

3. Se define la métrica *CBE-io* como:

$$\text{si } \exists p \in \Upsilon \text{ y } q \in \Upsilon \mid p \text{ es inversa de } q \text{ entonces } CBE\text{-io} = | \Upsilon | - 2$$

4. Se define la métrica *SC* como:

$$\text{si } \exists c \in \text{dominio}(p) \text{ y } c \in \text{rango}(p) \forall p \in \Upsilon \text{ entonces } SC = | \Upsilon |$$

5. Se define la métrica *CBE* como:

$$CBE = CBE\text{-out} + CBE\text{-in} + CBE\text{-io} + SC$$

Las métricas *CBE* permiten analizar individualmente el acoplamiento de las clases en una ontología, a diferencia del resto de métricas, que analizan las ontologías en general. Estos valores permiten, por un lado, identificar aquellas clases que son las más acopladas en el modelo ontológico, esto es, aquellos conceptos con más relaciones. Este aspecto es importante, ya que generalmente suele pasar que los conceptos con mayor número de relaciones, corresponden a aquellos más importantes dentro del modelo ontológico [88, 85].

También se puede hacer una clasificación de las clases, de acuerdo al tipo de acoplamiento que presentan, ya que desde el punto de vista semántico, no es lo mismo que una clase contenga únicamente relaciones “de salida”, a que contenga únicamente relaciones “de entrada”. El hecho de que una clase pertenezca siempre al dominio de todas sus relaciones implica que su papel dentro del modelo ontológico, está más relacionado con un objeto independiente, mientras que el caso contrario, implicaría un objeto más dependiente. Esta clasificación vista como una analogía del idioma castellano, sería como identificar un sustantivo y un adjetivo calificativo; el sustantivo representa un individuo real independiente, mientras que el adjetivo califica o determina al sustantivo.

Un ejemplo de esta clasificación con el modelo ontológico del dominio de los vinos, descrito en la sección 2.1 y mostrado en la Figura 2.1, sería la clase *Wine* y la clase *WineGrape*. La clase *Wine* que define el concepto de vino en general, contiene mayormente relaciones “de salida”, mientras que por el contrario, la

clase *WineGrape* que define el concepto del tipo de uva para el vino, contiene mayormente relaciones “de entrada”, ya que el planteamiento del modelo ontológico se centra en los vinos, y las uvas en este caso particular, representan un concepto que califica al vino. Si por el contrario, el planteamiento del modelo ontológico se centrara en las frutas para crear vinos, el concepto de uva tendría un significado más independiente, mientras que el concepto de vino sería más dependiente del concepto de uva. Este ejemplo representa dos perspectivas diferentes basadas en el significado que tienen los conceptos definidos.

Por otro lado, a partir de estas métricas se puede tener una vista general del acoplamiento global de una ontología, ya que se pueden detectar todas aquellas clases con relaciones y aquellas otras que no tienen relaciones, lo que se interpreta como que están aisladas, semánticamente hablando. En dependencia del dominio de aplicación, una ontología puede estar acoplada en mayor o en menor medida. Para calcular el acoplamiento global de una ontología, basta con calcular el porcentaje de clases acopladas respecto del total de clases en el modelo. No hay una estimación que indique cuando una ontología está débil o fuertemente acoplada (análogamente a un sistema de *software* [133]), por lo que una futura línea de investigación podría desarrollarse en este sentido.

5.2. Soluciones propuestas

Esta sección se enfoca en plantear soluciones basadas en diversas técnicas de visualización e interacción para la problemática de las herramientas de modelado y visualización, descrita en la sección 4.4. Esta problemática comprende cuatro aspectos principales: la saturación de la información y de elementos visuales en las representaciones, la poca expresividad visual, la pérdida del contexto de navegación y la falta de mecanismos robustos de interacción con el usuario.

5.2.1. Saturación de elementos visuales

El problema de la saturación de información en las visualizaciones genera a su vez una saturación de los elementos visuales que se representan. Una solución a este problema radica en cambiar el enfoque de la visualización del modelo ontológico. En lugar de presentar todos los elementos en la misma visualización (clases, propiedades, relaciones e instancias), como hacen la mayoría de las herramientas, sería conveniente definir nuevos procesos y flujos de trabajo, en los que el usuario pueda analizar el modelo en varias fases y, en cada una de estas, analizar un aspecto específico. Esta solución, primero, evita el tener

que desplegar demasiada información que el usuario es incapaz de analizar y, segundo, permite definir un flujo de actividades específico para cada fase del análisis, de acuerdo al interés del usuario.

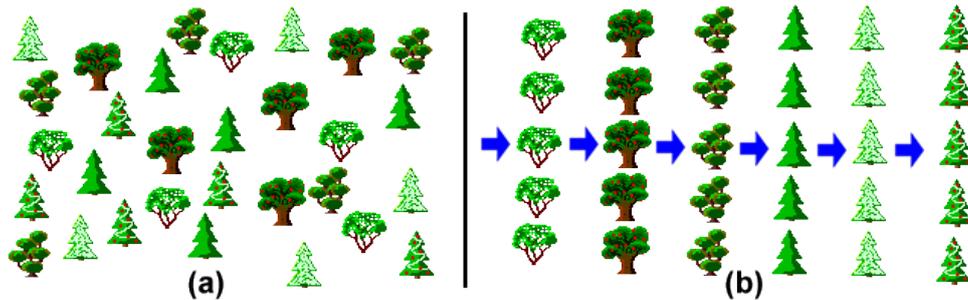


Figura 5.6: (a) Muestra un esquema en el que toda la información se muestra al mismo tiempo, lo que complica en demasía el proceso de abstracción. (b) Muestra un enfoque diferente, en el que el usuario sigue un flujo de trabajo definido para ir conociendo parte del sistema modelado.

Las Figuras 5.6 (a) y 5.6 (b) ilustran una analogía de la propuesta de solución. Por un lado, la Figura 5.6 (a) muestra un esquema en donde toda la información se muestra al usuario al mismo tiempo, en este caso, se trata de un bosque lleno de árboles. En este esquema resulta difícil, primero, identificar toda la información y, segundo, establecer claramente los tipos de elementos (árboles en el bosque), el número de elementos de cada tipo, etc.

Por otro lado, la Figura 5.6 (b) muestra un enfoque diferente para mostrar la información, en donde se define un flujo de actividades que permiten al usuario analizar la información de forma parcial, de acuerdo a sus intereses. A través del flujo de actividades en las diversas fases, el proceso de abstracción del conocimiento, se facilita y se evita saturar de información los modelos visuales.

5.2.2. Poca expresividad visual

La expresividad visual se ha definido en la sección 3.1.1 como el número de variables visuales utilizadas para hacer más eficiente una visualización. Una de estas variables es la distribución espacial de los elementos en el plano dimensional, un aspecto fundamental en una visualización. La distribución física de los elementos se utiliza también para proveer información visual. El propósito más común de una distribución de elementos es establecer una relación de orden entre estos, como se ha ilustrado en la Figura 3.6. Sin embargo, una distribución eficiente puede proveer mucha más información, como por ejemplo, relaciones de propiedad, de dependencia o similitudes entre elementos. Es jus-

to en este aspecto en el que el uso de una distribución física de los elementos, puede mejorar considerablemente las visualizaciones.

Otro problema con las herramientas visuales es la falta de un manejo adecuado de los planos y los contextos. Este problema se ilustra en la figura 4.6 con la herramienta Jambalaya, ya que al seleccionar uno de los elementos, la representación de las relaciones de este con otros elementos no resulta lo suficientemente clara y sencilla. En este caso, las relaciones se muestran como aristas que unen a los propios elementos relacionados. Para el caso de la selección de un elemento, las aristas involucradas se vuelven más anchas para resaltarlas del resto. Sin embargo, esta solución no es suficiente para que realmente el usuario pueda percibir de forma efectiva cuál es el elemento seleccionado, las relaciones de este elemento y los otros elementos relacionados. Esto se debe principalmente al hecho de que no hay un buen manejo de los focos, los contextos y los planos visuales.

Una solución a este problema es el uso de otra de las variables visuales descritas en la sección 3.1.1, que es la opacidad, mediante el difuminado de elementos para definir contextos. Para el caso de la selección de elementos en la herramienta Jambalaya mostrada en la Figura 4.6, el difuminado de los elementos no seleccionados permitiría resaltar mucho más los elementos que se pretenden mostrar en un primer plano (el elemento seleccionado, sus relaciones y los elementos relacionados con este). Esta solución ha sido descrita en la sección 3.2.2.2 relacionada con el *Foco+Contexto*, una de las técnicas de interacción definidas en el área de la Analítica Visual.

Otra solución a este mismo problema con la selección de los elementos visuales, es el uso de la técnica de *zoom semántico*, descrita en la sección 3.2.2.3. Esta técnica se utiliza justamente en los casos en los que se quiere seleccionar alguno de los elementos, y destacar información relevante a este, como pueden ser sus relaciones y los elementos relacionados.

5.2.3. Pérdida del contexto de navegación

La pérdida del contexto de navegación es otra problemática descrita en la sección 4.4. El principal problema que comparten la gran mayoría de las herramientas es la pérdida del contexto en la representación de la taxonomía de conceptos. Esto se debe a que implementan una solución basada en listas, que como se ha descrito en la sección 4.4, no es una solución eficiente desde el punto de vista de las visualizaciones.

La sección 4.4 describe dos soluciones al problema de la pérdida del contexto para la representación de jerarquías. Básicamente, la principal solución

radica en el uso de técnicas de visualización especializadas en la representación de grandes taxonomías de conceptos, como el *treemap* y los árboles de jerarquías. Entre las ventajas en el uso de estas técnicas, se pueden destacar que aprovechan de una manera eficiente el espacio físico visual disponible, y que la distribución física (*layout*) de los elementos resulta bastante intuitiva. Sin embargo, incluso estas visualizaciones pueden llegar a resultar insuficientes, por lo que es necesario proporcionarles mecanismos de interacción robustos.

Uno de los mecanismos de interacción que han implementado los árboles jerárquicos, es la posibilidad de contraer y expandir ramas completas de la taxonomía. Este mecanismo evita la saturación de la vista con elementos visuales que están fuera del alcance del análisis del usuario, ya que este no puede analizar simultáneamente todas las ramas de la taxonomía.

Otra de las soluciones para la pérdida del contexto es el uso de la técnica de *zoom semántico*. Como se ha descrito en la sección 3.2.2.3, esta técnica se utiliza para desplegar información específica sobre un elemento seleccionado. Todas las visualizaciones en esta propuesta utilizan el *zoom semántico* en combinación con la técnica de *Foco+Contexto*, en donde el foco del elemento seleccionado se realiza a través del *zoom semántico*, y el contexto de la vista permanece visible.

Esta solución del uso del *Foco+Contexto* se genera como resultado de la interacción con una vista general, a partir de la cual, el usuario está interesado en analizar detalles específicos de alguno de los elementos, por lo que se combinan en una sola vista, tanto el contexto de la visualización general, como los detalles específicos del elemento seleccionado.

5.2.4. Falta de mecanismos de interacción

Otra problemática discutida en la sección 4.4 es la referente a la falta de mecanismos robustos de interacción con el usuario. Son precisamente los *treemaps* un ejemplo de una visualización estática, con poca interacción con el usuario, especialmente cuando se trata de navegar los niveles inferiores en una jerarquía profunda. Como se ha descrito, los *treemaps* representan una solución eficiente para la representación de grandes jerarquías, sobre todo, jerarquías grandes en amplitud. Sin embargo, la representación de jerarquías grandes en profundidad representa una situación más compleja.

El problema radica en llegar a representar clara e intuitivamente los elementos en los niveles inferiores de la jerarquía. En el caso específico del *treemap*, una solución es la posibilidad de recorrer la jerarquía completa desde los niveles superiores hasta los niveles inferiores. Esta interacción permite al usuario

explorar cada nivel, pero también, la visualización debe preservar el contexto completo, tanto de los niveles superiores como los inferiores, porque de lo contrario, sería la misma problemática discutida con las listas jerárquicas.

Esta problemática se ha resuelto con la implementación de un mecanismo para la navegación a través de los niveles del *treemap*. Esta solución se describe a detalle más adelante en la sección 5.3.2.1.

5.3. Soporte a las fases de implementación y análisis

Las secciones 5.1 y 5.2 describen las actividades de los procesos para el desarrollo del ciclo de vida de ontologías, así como soluciones específicas para corregir los problemas de las actuales herramientas de visualización y modelado de ontologías. Esta sección describe los aspectos técnicos más importantes de la herramienta denominada OWL-VisMod (acrónimo de *OWL Visual Modelling Tool*), la propuesta resultado de la investigación de esta tesis. Esta herramienta soporta los procesos de creación, gestión, análisis y reutilización de los modelos ontológicos y, más específicamente, todas las actividades de los flujos de trabajo definidos en las secciones 5.1.1, 5.1.2, 5.1.3 y 5.1.4.

OWL-VisMod es una herramienta de libre distribución¹, que tiene como objetivo principal, gestionar el ciclo de vida de las ontologías OWL. Su diseño es una capa visual superior sobre Jena², como se muestra en la Figura 5.7. OWL-VisMod interactúa con el *framework* Jena, (responsable de cargar y administrar las ontologías), y realiza consultas sobre el modelo ontológico, para después procesar esta información en sus propias estructuras de datos internas y, finalmente, visualizarla.

El diseño de OWL-VisMod incluye diferentes visualizaciones y técnicas de interacción, enfocadas en la corrección de los problemas de saturación de la información, la poca expresividad visual, la pérdida del contexto de navegación e implementa mecanismos robustos de interacción con el usuario. Los usuarios interactúan directamente con las visualizaciones en la capa superior, bajo un enfoque modelo-vista-controlador. La herramienta se encarga del control de las vistas de acuerdo a la interacción con el usuario, así como de los datos a desplegar acorde a la visualización activa.

Por otro lado, también posee funcionalidad para cargar ontologías en línea, para transmitir archivos a través del protocolo FTP (*File Transfer Protocol*)³,

¹<http://www.analiticavisual.com/juan/OWL-VisMod.html>

²<http://jena.sourceforge.net/>

³<http://www.faqs.org/rfcs/rfc765.html>

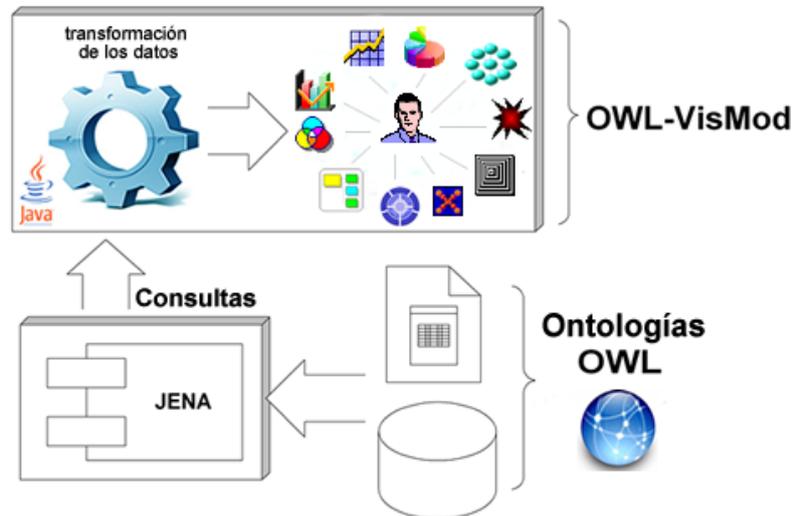


Figura 5.7: La arquitectura de OWL-VisMod representa una capa superior sobre Jena, que se encarga de administrar los modelos ontológicos directamente.

o para cargar modelos desde bases de datos locales o remotas.

5.3.1. Transformación de los datos

Como se ha ilustrado en la Figura 3.1 de la sección 3.1.1, los procesos de visualización y análisis, conllevan una fase de preprocesamiento y transformación de los datos. OWL-VisMod desarrolla un proceso de transformación de los datos originales del modelo ontológico en estructuras de datos propias, que mejoran el rendimiento de la herramienta durante las fases de análisis e implementación. A partir de estas estructuras de datos se realizan diversos cálculos en el momento de cargar las ontologías, por lo que, durante el tiempo en el que el usuario interactúa con las visualizaciones, algunos valores han sido previamente calculados, lo que hace más eficiente el proceso de navegación en la herramienta. La Figura 5.7, que muestra la arquitectura, ilustra la fase de transformación de los datos, previa a la fase de visualización.

El preprocesamiento de los datos conlleva realizar diversos cálculos previos al proceso de visualización, lo que implica también un uso de memoria para almacenar toda esa información. En ocasiones, mucha de la información calculada previamente no se utiliza, lo que implica un manejo poco eficiente de los recursos. Por otro lado, los cálculos en tiempo real durante la ejecución del sistema, cuando son muy costosos en tiempo de ejecución, pueden ralentizar el

esquema de visualización y la interacción con el usuario.

Es importante encontrar un equilibrio entre el preprocesamiento de datos y los cálculos en tiempo real. La decisión de qué cálculos realizar durante la fase de preprocesamiento y cuáles más realizar en tiempo real, es un aspecto determinante para el buen rendimiento del sistema. En el caso de OWL-VisMod, los cálculos en tiempo real implican la ejecución de consultas sobre el modelo ontológico directamente a través de la biblioteca Jena, lo que añade otro costo en tiempo y procesamiento.

Al considerar estos aspectos, el diseño de OWL-VisMod contempla el cálculo de todas las métricas de las clases, durante la fase de preprocesamiento. También se procesa la taxonomía de clases, los valores de las propiedades intrínsecas y las relaciones de acoplamiento, ya que estos datos se muestran en visualizaciones más generales, mientras que los cálculos que se llevan a cabo en tiempo real, corresponden principalmente a toda la información que se despliega en las visualizaciones de *zoom semántico*, ya que es información muy específica, resultado de alguna cierta consulta.

5.3.2. Soporte a la fase de implementación

El flujo de trabajo para la fase de implementación⁴, descrito en la sección 5.1.1 y mostrado en la Figura 5.2, comprende cuatro actividades principales: la definición de la taxonomía de conceptos, la definición de las propiedades intrínsecas, la definición de las relaciones de acoplamiento y finalmente, la implementación de las instancias de las clases.

Para soportar estas actividades, OWL-VisMod incorpora dos grupos principales de visualizaciones. El primero de los grupos se enfoca en la implementación y análisis de la taxonomía de los conceptos, mientras que el segundo grupo está basado en el uso del *zoom semántico* para definir aspectos muy específicos de las clases, como las propiedades intrínsecas, el acoplamiento y las instancias. La Figura 5.8 muestra el flujo de trabajo y las actividades para el proceso de creación de ontologías. Cada actividad tiene asociadas algunas visualizaciones que soportan su desarrollo con la herramienta OWL-VisMod.

De acuerdo a la propuesta descrita en la sección 5.1.1, se pretende que este proceso sea iterativo, en donde cada iteración termina con un submodelo ontológico. Las secciones 5.3.2.1 y 5.3.2.2 describen cada uno de los grupos de visualizaciones que soportan el proceso de creación de modelos ontológicos con OWL-VisMod, específicamente en la fase de implementación.

⁴Este proceso de creación se ilustra con algunas imágenes y vídeos en: <http://www.analiticavisual.com/juan/snapshots.html>.

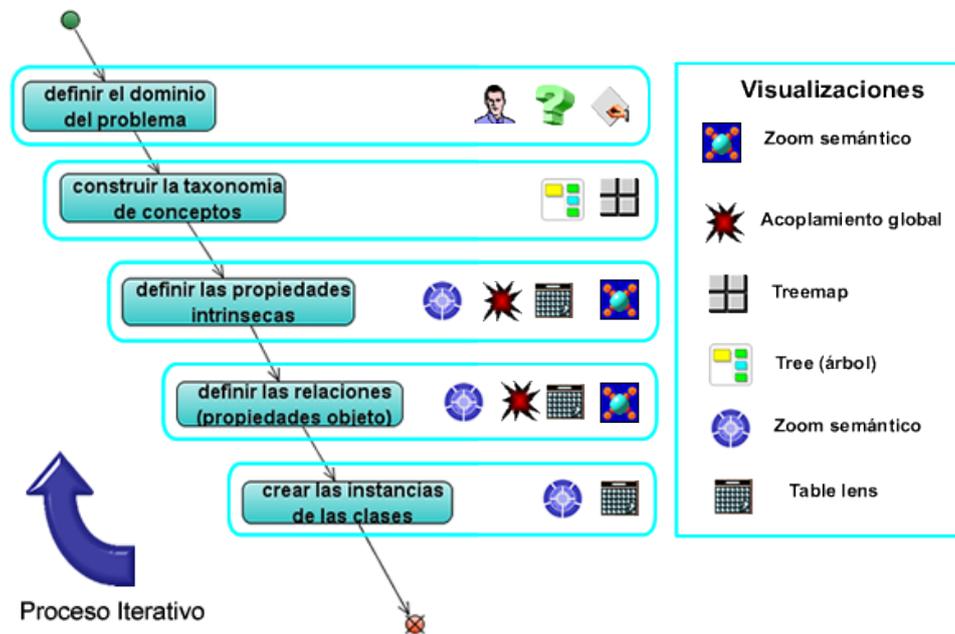


Figura 5.8: El flujo de trabajo que define las principales actividades que se desarrollan en el proceso de creación de ontologías. Cada actividad tiene asociadas algunas herramientas, técnicas o visualizaciones, que soportan su desarrollo.

5.3.2.1. Taxonomía de conceptos

El primero de los grupos está enfocado en modelar y analizar la taxonomía de los conceptos, y está formado por los *treemaps* y los árboles de jerarquías. Ambas soluciones se han implementado en OWL-VisMod, ya que han demostrado en diversas áreas, que son lo suficientemente robustas como para visualizar eficientemente, grandes jerarquías de conceptos, sin la pérdida del contexto, como se ha documentado en la sección 5.2.3. Un vídeo que ilustra todo el procedimiento para crear una clase se encuentra disponible en el sitio web del proyecto⁵. Otra ventaja es que son dos técnicas muy intuitivas para el usuario, lo que le permite fácilmente entender la información que se muestra, en este caso, una taxonomía de conceptos. La implementación de las dos técnicas permite al usuario analizar la jerarquía de conceptos desde perspectivas diferentes y complementarias.

Las principales actividades relacionadas con la taxonomía de conceptos son,

⁵<http://www.analiticavisual.com/juan/videos/createClass.avi>

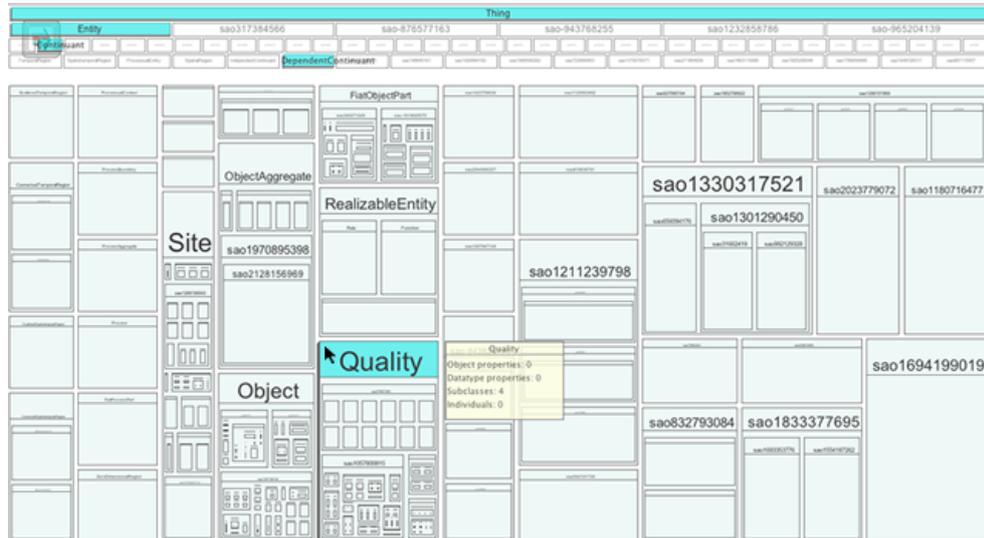


Figura 5.9: Una vista general de una jerarquía de clases, con una visualización *treemap* en OWL-VisMod. Los cuatro niveles superiores se han colocado por encima del *treemap*, el cual ha sido calculado a partir del quinto nivel. Esta funcionalidad permite recorrer en profundidad todos los niveles internos de la taxonomía.

por un lado, la búsqueda de conceptos o clases específicas así como la jerarquía de las superclases de estos conceptos, y por otro lado, la creación de nuevos conceptos o clases, con lo que se define la taxonomía.

Una de las principales funciones de los *treemaps* es la representación de jerarquías, y pueden utilizar dos variables para representar información: el área de los rectángulos y el color [151]. En el caso de OWL-VisMod, el área de los rectángulos que representa a las clases, es proporcional al número de clases internas, mientras que el color está asociado al tipo de clase, ya sea por ejemplo una enumeración, una clase disjunta, una clase normal, etc. Esta visualización se basa en el uso de las variables visuales del color y el tamaño para proveer información al usuario.

Una de las mayores ventajas en el uso de los *treemaps* para representar jerarquías es que la visualización resulta bastante intuitiva. La naturaleza rectangular de los elementos en el *treemap* permite representar las clases del modelo ontológico de forma similar a la tan ampliamente utilizada representación de cajas rectangulares de los diagramas de clases UML. De esta forma, se aprovecha, por un lado, la naturaleza de los *treemaps* y, por el otro, la representación de clases en los diagramas UML.

A pesar de que estas visualizaciones utilizan el espacio visual de forma muy eficiente, para jerarquías excesivamente grandes sobre todo en profundidad, puede resultar incluso insuficiente, ya que los elementos de los niveles inferiores pueden pasar prácticamente desapercibidos por su tamaño tan pequeño. Para resolver esta problemática, es necesario incluir mecanismos de interacción en el esquema original del *treemap* [151]. La solución consiste en proveer funcionalidad de tal forma que el usuario pueda recorrer todos los niveles de la jerarquía, pero sin perder el contexto global de la jerarquía.

La Figura 5.9 muestra una vista general de la jerarquía de la ontología SAO⁶ por medio de una visualización de *treemap*, con setecientos sesenta y cuatro clases, cuarenta y cinco propiedades de datos, treinta y seis propiedades objeto, y una profundidad de once niveles.

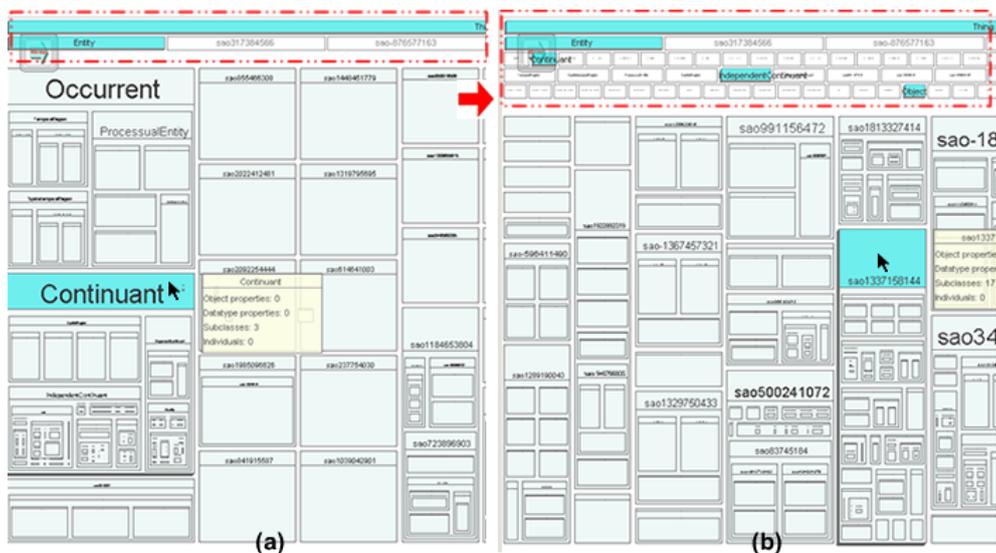


Figura 5.10: Una secuencia de la navegación sobre los niveles internos del *treemap*. (a) muestra el *treemap* a partir del tercer nivel de la jerarquía, los dos niveles superiores se muestran en la parte de arriba. (b) muestra el *treemap* a partir del sexto nivel de la jerarquía, los cinco niveles superiores se muestran en la parte de arriba.

OWL-VisMod implementa una solución que permite recorrer los diversos niveles en la jerarquía del *treemap*⁷. Las Figuras 5.10 (a) y (b) muestran la navegación sobre los niveles internos del *treemap*, a partir del tercer nivel para

⁶<http://ccdb.ucsd.edu/SAO/1.2/SAO.owl>

⁷Un vídeo que ilustra esta solución se encuentra disponible en:
<http://www.analiticavisual.com/juan/videos/zoomingInTreemap.avi>.

la Figura 5.10 (a) y a partir del sexto nivel para la figura Figura 5.10 (b).

Es importante que el usuario en todo momento pueda conocer la jerarquía de las superclases de la clase seleccionada. Las Figuras 5.10 (a) y 5.10 (b), muestran que la solución implementada permite al usuario conocer la jerarquía superior de la clase. Esta jerarquía se resalta para desplegar todo el camino o la rama que conduce desde la clase raíz a la clase seleccionada.

Esta funcionalidad de navegación en profundidad sobre la jerarquía es una solución nueva y diseñada específicamente para las necesidades de OWL-VisMod, para el modelado de ontologías especialmente grandes y profundas. Esta interacción conserva el contexto completo y permite en todo momento conocer la jerarquía superior de cada clase seleccionada, las clases en la jerarquía superior se resaltan del resto, para indicar claramente la jerarquía, tal y como se muestra tanto en la Figura 5.9 como en las Figuras 5.10 (a) y 5.10 (b).

Por otro lado, la técnica de los árboles de jerarquías (*trees*) representa una segunda opción para la representación de la taxonomía de conceptos⁸. Esta técnica complementaria a los *treemaps*, es una vista exploratoria de la taxonomía, provista con una mayor interacción dinámica en cuanto a animación, en contraste con el *treemap*. Esta interacción es necesaria, ya que la visualización tiene que implementar mecanismos para ocultar nodos o ramas completas de la jerarquía, para evitar la saturación de elementos visuales en el caso de grandes taxonomías. El proceso de animación sobre una técnica visual es importante, ya que permite al usuario distinguir claramente los cambios efectuados en las visualizaciones, como resultado de los mecanismos de interacción, sin que ello suponga una carga cognitiva adicional.

La estrategia de análisis que siguen los árboles es analizar a través de un recorrido por las ramas de la jerarquía, reduciendo el esfuerzo cognitivo del usuario durante el proceso de exploración. Al seleccionar algún elemento en particular, toda la rama inferior del elemento seleccionado se despliega, mientras que las otras ramas se ocultan. Esta interacción permite por un lado, establecer una navegación sobre el árbol, y por otro lado, evita la saturación visual de elementos.

Al igual que en el *treemap*, todas las superclases en la jerarquía de la clase seleccionada se resaltan, para lograr establecer todo el camino desde el nivel superior al inferior dentro de la taxonomía de clases. La Figura 5.11 muestra una vista de esta técnica con la misma taxonomía y la misma clase seleccionada

⁸Un vídeo se encuentra disponible en:
<http://www.analiticavisual.com/juan/videos/hierarchicalTree.avi>.

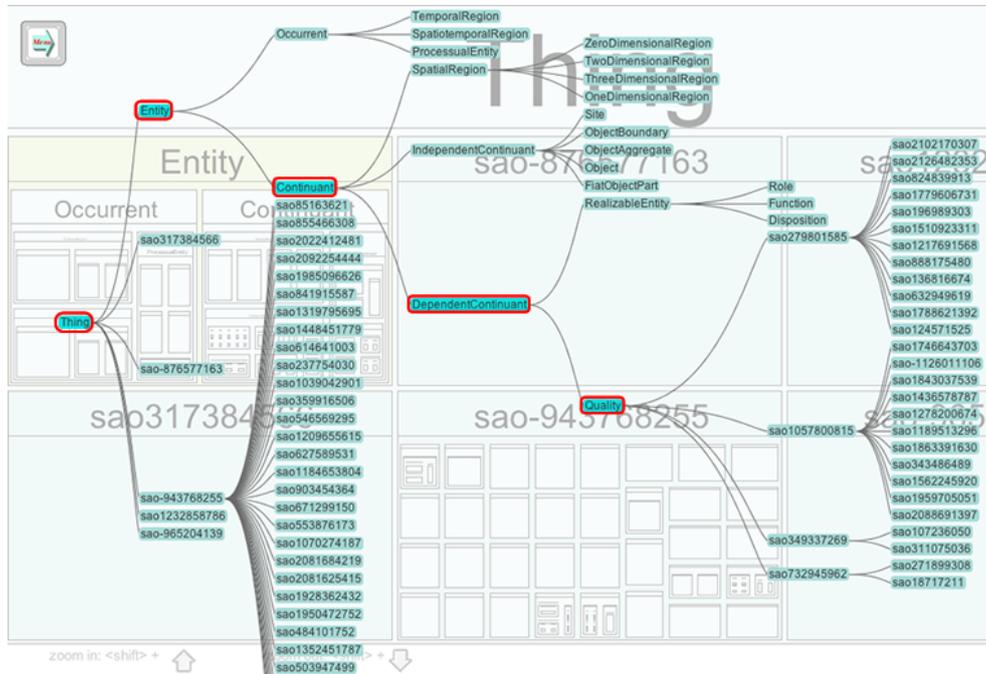


Figura 5.11: Una vista general de la jerarquía de clases mediante el uso de una visualización basada en árboles jerárquicos. La jerarquía de la clase seleccionada se ha resaltado con recuadros en color rojo. El *treemap* se mantiene en un segundo plano, como efecto de la interacción que hace la transición entre las dos vistas.

que en la Figura 5.9⁹. Ambas figuras muestran las dos perspectivas para la representación visual de las taxonomías.

La principal ventaja que potencia el uso de estas dos técnicas para la representación de taxonomías es, definitivamente, el uso de una distribución de elementos (*layout*) eficiente, que transmite perfectamente al usuario tanto la posición de una clase dentro de la jerarquía, como toda la secuencia de sus superclases en los niveles superiores. Ambas visualizaciones permiten al usuario recorrer la taxonomía de la ontología de una forma eficiente, incluso para jerarquías grandes en amplitud o profundidad. La vista de árbol implementa funcionalidad de usuario más dinámica que el *treemap*, con el uso de animaciones, técnicas de *zoom* físico, redimensionado, expansión o contracción de ramas en la taxonomía, etc.

⁹Un vídeo de la navegación del árbol de jerarquías se encuentra disponible en: <http://www.analiticavisual.com/juan/videos/hierarchicalTree.avi>

Como se ha descrito en la sección 5.2.3, las listas representan la opción más utilizada por parte de las herramientas existentes, para la representación de taxonomías. La Figura 4.9 muestra la misma taxonomía de conceptos (ontología SAO) que las Figuras 5.9 y 5.11. Es muy claro que el uso de *treemaps* y árboles mejoran considerablemente con respecto al uso de listas. Principalmente porque el espacio asignado a las listas no permite desplegar la jerarquía completa de un concepto, por lo que se tiene que hacer uso de las barras de desplazamiento.

Estas técnicas de visualización mejoran considerablemente las actuales propuestas de las otras herramientas, ya que resultan muy claras y corrigen el problema de la saturación de elementos. También permiten al usuario generar fácilmente un mapa mental de toda la jerarquía de un cierto concepto (superclases), con lo que el usuario puede comprender perfectamente el grado de especificación de la clase.

5.3.2.2. Visualizaciones que muestran los detalles

Tanto el *treemap* como los árboles de jerarquías representan visualizaciones muy generales, centradas en la representación taxonómica de los conceptos del modelo ontológico. Sin embargo, estas técnicas no están diseñadas para analizar los detalles de las clases, por dos razones principalmente: la primera, porque el propio diseño de la herramienta establece la separación de la funcionalidad para evitar la saturación de elementos visuales y, en segundo lugar, por la propia naturaleza de estas técnicas, que no es apropiada para la representación detallada de los elementos.

Las vistas que muestran los detalles tienen diferentes actividades, en dependencia de la visualización general que interactuá con la vista detallada. Para el caso específico de las vistas que representan la taxonomía de conceptos, el interés de los usuarios en los detalles de una clase, son por un lado las propiedades tanto objeto como de datos, y por otro lado, las instancias que puede tener la clase seleccionada.

Para el caso de la vista global que representa el acoplamiento de las clases, el principal interés de los usuarios en una vista detallada, es analizar el acoplamiento específico de la clase. Este acoplamiento está dado por las propiedades y por las clases acopladas con la propia clase seleccionada. Esta vista detallada, a diferencia de la simple representación de las propiedades, tiene que permitir al usuario entender la relación establecida entre la clase que se representa en detalle, con las diversas clases acopladas.

La alternativa propuesta para este tipo de visualizaciones es el uso de la

técnica conocida como *zoom* semántico, definida en la sección 3.2.2.3, y que se utiliza para desplegar los detalles de los elementos visuales seleccionados. El *zoom* semántico se utiliza en conjunto con la técnica de *Foco+Contexto*. El contexto en una visualización es un aspecto fundamental, como se ha descrito en secciones que anteceden. Las visualizaciones implementadas permiten mantener el contexto general que ubica exactamente al usuario, en la fase del análisis en la que se encuentra. La Figura 5.12 muestra una visualización de *zoom* semántico sobre una clase llamada *Actividad*, en la que el contexto completo permanece visible, aunque con una opacidad. Este ejemplo ilustra la aplicación de la técnica de *Foco+Contexto*, en donde el enfoque principal lo tiene la visualización de *zoom* semántico, mientras que el contexto permanece como fondo en un plano secundario.

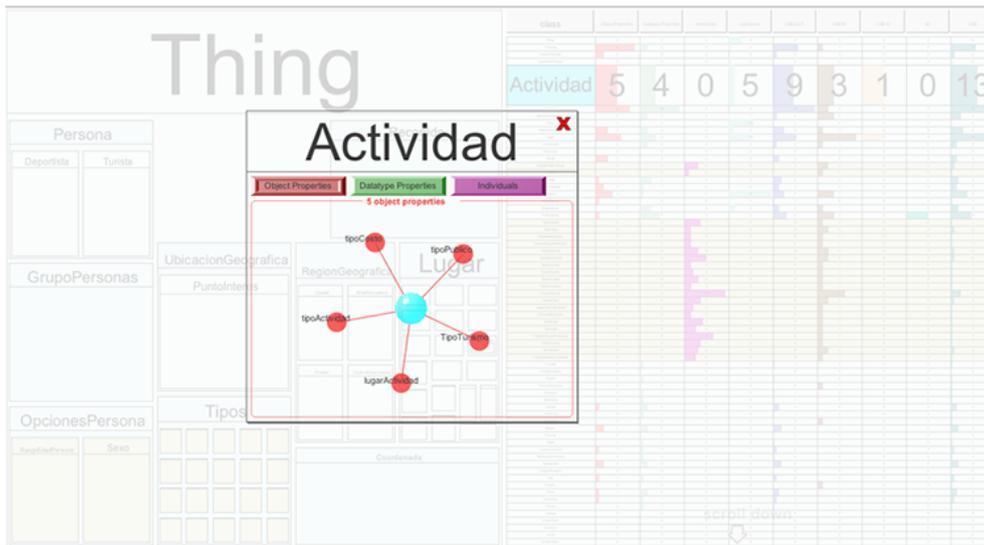


Figura 5.12: Una vista de la implementación del *zoom semántico* en una visualización en donde el contexto completo permanece.

El flujo general de navegación de la herramienta se ilustra en la Figura 5.13, comienza con una técnica general (como el *treemap*), en la cual un usuario interesado en conocer los detalles de alguno de sus elementos, lo selecciona para poder cambiar la vista a un *zoom* semántico. Este flujo se ilustra en la Figura 5.12, en donde se utiliza la técnica de *Foco+Contexto*, en la que el contexto permanece para ilustrar la transición entre la vista general y la vista detallada del elemento seleccionado, mediante el *zoom* semántico. Esta vista de *zoom* semántico proporciona el foco de la aplicación al nuevo elemento seleccionado, y permite al usuario navegar sobre los elementos internos de la nueva visualización. Para el caso de las clases, sus elementos internos son las

propiedades de datos, las propiedades objeto y las instancias.

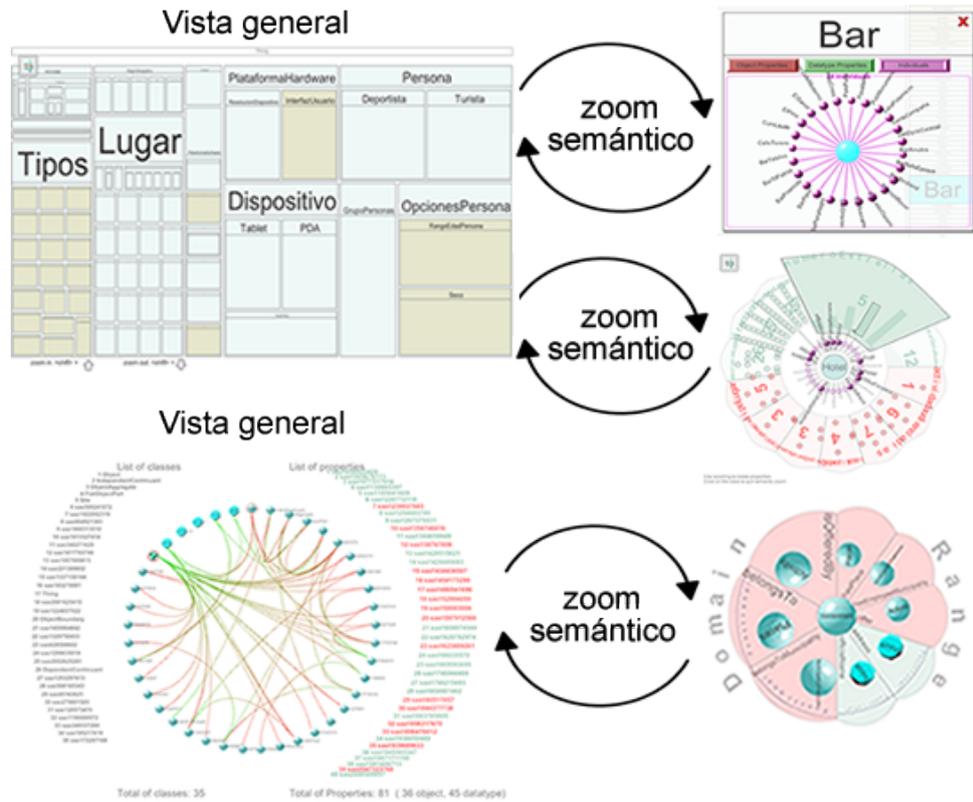


Figura 5.13: Una vista general del flujo de navegación, que comienza desde una vista general hacia lo particular. Una vez presentados los detalles, el flujo permite regresar hacia la vista global, y repetir esta tarea de navegación indefinidamente.

El diseño del flujo de navegación en la herramienta, que se muestra en la Figura 5.13, sigue un enfoque descendente en cuanto al nivel de abstracción se refiere. Dicho de otra forma, parte de niveles superiores de abstracción, hacia niveles muy específicos en cuanto a detalle. Este diseño tiene principalmente dos propósitos; el primero, es evitar visualizaciones que combinen vistas generales con detalles específicos, lo que resulta en una saturación innecesaria de elementos visuales. El segundo objetivo es la definición de una secuencia de tareas, que mejoren los procesos de abstracción y la obtención de conocimiento por parte del usuario. Esta secuencia define la forma en la que el usuario navega e interactúa con la herramienta, siempre con el claro objetivo de mejorar la cognición del modelo de datos representado.

El uso de visualizaciones basadas en la técnica de *zoom* semántico corresponde al segundo grupo, formado por tres distintas visualizaciones en total, de las cuales, el *treemap* tiene asociados dos de estos tipos. Estas representaciones visuales se enfocan en mostrar las propiedades objeto, las propiedades de datos y las instancias de la clase seleccionada.

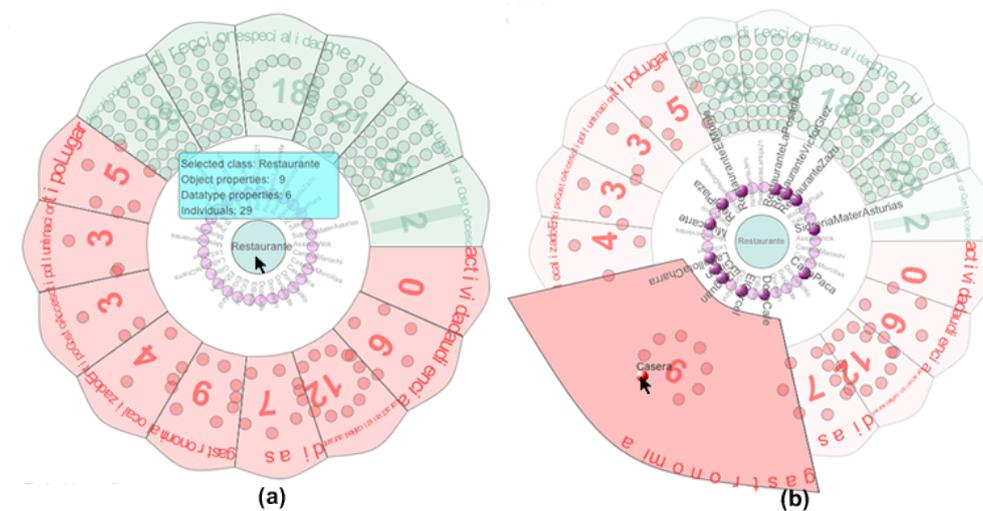


Figura 5.14: (a) muestra el *zoom* semántico sobre una clase llamada *Restaurante*, con nueve propiedades objeto (coloreadas en tono rojo), y seis propiedades de datos (coloreadas en tono verde). La Figura (b) muestra el *zoom* semántico sobre la misma clase *Restaurante*, con una selección de la propiedad *gastronomía* que tiene un total de nueve valores de instancias. La consulta visual despliega todas aquellas instancias de restaurantes, que tienen definida en su *gastronomía* al valor “casera”.

El primero de los esquemas de *zoom* semántico definido en [96], se ilustra en la Figura 5.14¹⁰. Se utiliza para desarrollar la técnica de consultas visuales (*visual querying*), definida y descrita en detalle en la sección 3.2.3. El uso de esta técnica tiene como principal característica que todos los elementos de una clase (propiedades, instancias y valores) se representan en la misma visualización, lo que permite que se puedan hacer consultas directamente sobre estos datos. La interacción con el usuario incluye la creación de propiedades objeto¹¹, propiedades de datos¹² e instancias¹³.

¹⁰Imágenes con una mayor resolución, se encuentran disponibles en: <http://www.analiticavisual.com/juan/snapshots2.html>.

¹¹<http://www.analiticavisual.com/juan/videos/createObject.avi>

¹²<http://www.analiticavisual.com/juan/videos/createDatatype.avi>

¹³<http://www.analiticavisual.com/juan/videos/createIndividual.avi>

Las Figuras 5.14 (a) y 5.14 (b) muestran dos imágenes de *zoom* semántico sobre una de las clases durante la fase de desarrollo de la ontología *turisem.owl*, descrita como caso de estudio más adelante¹⁴. Esta técnica de *zoom* semántico despliega las propiedades con una distribución radial alrededor de la clase seleccionada, la cual se coloca al centro de la visualización. Los valores de las instancias de la clase en el rango de la propiedad, se colocan dentro del polígono que representa a la propiedad. Estos valores pueden ser numéricos para el caso de las propiedades de datos de tipo numéricas (rango de los enteros o de punto flotante), o nominales para el resto de los tipos de datos y las propiedades objeto. Para el caso de los valores numéricos, la representación interna de los valores es a través del uso de histogramas, mientras que para el resto de los tipos de datos, se utiliza una representación de agrupaciones (*clusters*) de elementos con forma esférica.

La Figura 5.14 (a) muestra la clase *Restaurante*, que tiene nueve propiedades objeto coloreadas en rojo, y seis propiedades de datos coloreadas en verde. Por otro lado, la Figura 5.14 (b) muestra la misma clase con la selección de una de las propiedades objeto (*gastronomía*), lo que origina que el foco dentro de la visualización se le otorgue a esta propiedad. El ángulo del sector circular asignado a la propiedad se incrementa para poder desplegar los valores internos más claramente, y el resto de las propiedades se difuminan, lo que visualmente hace que la propiedad seleccionada sobresalga del resto. El usuario puede interactuar con cada uno de los elementos internos, que en este caso en específico suman un total de nueve. La Figura 5.14 (b) muestra el resultado de una consulta visual en la que se buscan todas las instancias de la clase *Restaurante* que para la propiedad *gastronomía* tengan el valor “casera”.

La expresividad visual es un aspecto fundamental que se ha considerado con esta propuesta de visualización; primeramente, con el uso de un patrón de colores distintivos sobre las propiedades, clases e instancias, que permanece constante en todas las visualizaciones de la herramienta. Por otro lado, el uso de simbología distintiva como esferas, histogramas o agrupaciones de elementos, enriquecen la expresividad visual. Así mismo, el uso de técnicas de difuminado de elementos para poder resaltar algunos elementos de otros, hace que visualmente la interacción del usuario sea intuitiva, sencilla y clara.

Esta visualización es una propuesta nueva [96], que se ha diseñado específicamente para resolver el problema de las consultas visuales sobre los elementos internos de las clases. Entre las ventajas que tiene, destaca el uso de diversas variables visuales para fomentar la expresividad visual. Estas variables incluyen el difuminado de elementos, la definición de formas (esferas, histogramas), la

¹⁴Un vídeo con la interacción está disponible en:
<http://www.analiticavisual.com/juan/videos/tesis/queryingRestaurante.avi>

distribución circular o radial de los elementos y el uso de colores distintivos. Una de las ventajas en el uso de la distribución radial es que permite la representación intuitiva de los elementos sin el uso de aristas para relacionarlas, lo que origina visualizaciones más claras y limpias.

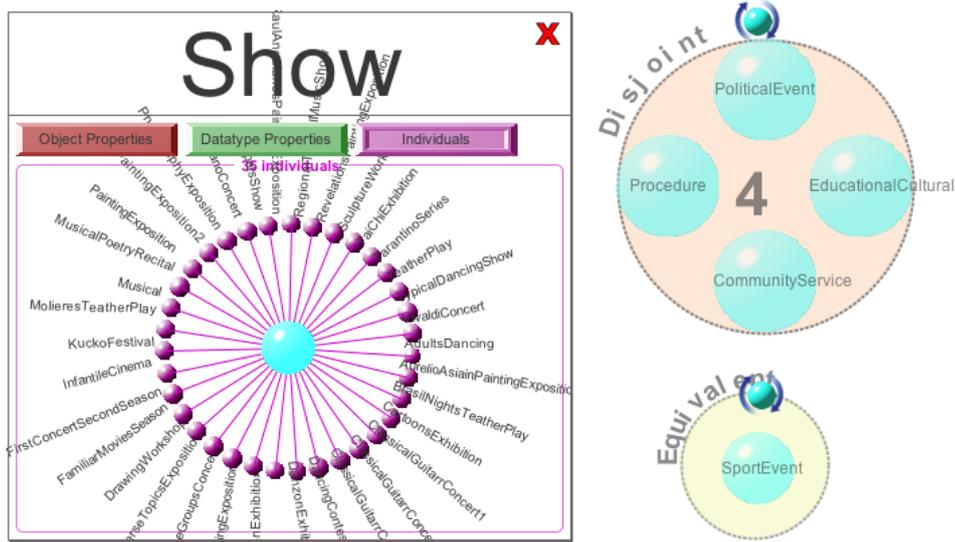


Figura 5.15: Esta técnica de *zoom* semántico se basa en la distribución radial de sus elementos para representar las instancias de la clase seleccionada. Las clases disjuntas y las clases equivalentes a la clase seleccionada se muestran en una representación de agrupaciones de clases.

El segundo esquema de *zoom* semántico se basa en la representación de los componentes internos de una clase por separado de forma excluyente, esto es, por un lado se representan las propiedades objeto, por otro lado las propiedades de datos y por otro las instancias. Esta visualización de *zoom* semántico se ilustra en la Figura 5.15¹⁵, y difiere del *zoom* semántico descrito anteriormente, en que no desarrolla la técnica de consultas visuales (*visual querying*), sino que por el contrario, se limita a desplegar por separado, los valores de las propiedades y las instancias de la clase seleccionada.

Los elementos se representan como esferas con un color distintivo (rojo para las propiedades objeto, verde para las propiedades de datos y púrpura para las instancias), en una distribución radial con la clase concéntrica. Otro aspecto importante a considerar, es la representación de las clases disjuntas y las clases

¹⁵Más imágenes y vídeos están disponibles en:
<http://www.analiticavisual.com/juan/snapshots2.html#Semanticzoom>.

equivalentes en ambas visualizaciones de *zoom* semántico. Esta representación se muestra en la Figura 5.15, con una agrupación (*cluster*) que indica las clases disjuntas y equivalentes a la clase seleccionada.

Claramente se pueden distinguir las dos perspectivas en cuanto al *zoom* semántico: por un lado, la visualización mostrada en la Figura 5.14 que es inclusiva, ya que permite a todos los elementos interactuar entre sí, mientras que la visualización mostrada en la Figura 5.15 es excluyente, ya que los elementos se muestran por separado. Esta estrategia de múltiples vistas es muy comúnmente utilizada por las aplicaciones de Analítica Visual, ya que permite analizar los mismos datos desde perspectivas diferentes, para lograr una mejor abstracción del conocimiento.

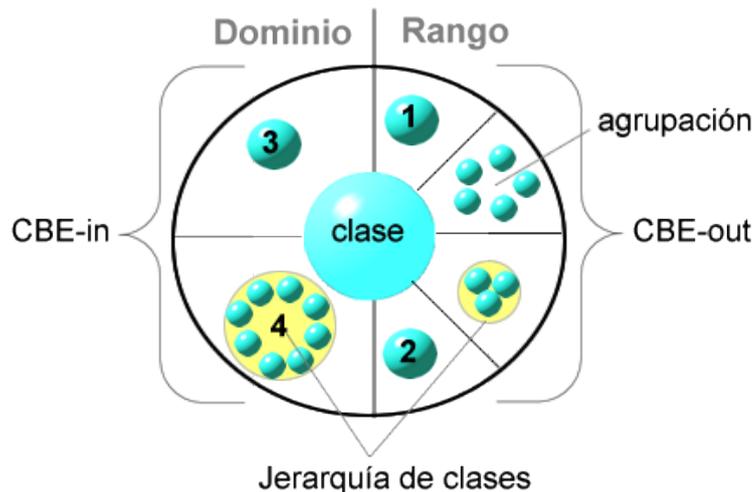


Figura 5.16: La visualización del acoplamiento de una clase de acuerdo a las métricas CBE. El acoplamiento de la clase seleccionada se representa con las propiedades alrededor de esta. En cada propiedad se colocan las clases que representan el acoplamiento ya sea CBE-out o CBE-in.

El tercer esquema de *zoom* semántico parte de un principio diferente. En vez de representar los elementos internos de una clase, representa el acoplamiento de la clase, esto es, las relaciones de la clase con otras clases. Esta visualización es nueva, y ha sido implementada específicamente para la representación del acoplamiento de una clase en OWL-VisMod [85].

Esta visualización de *zoom* semántico se basa en el uso de las métricas de acoplamiento (CBE), descritas en la sección 5.1.4 y definidas en [84]. Las métricas generalmente no son más que simples fórmulas matemáticas, cuyo resultado es un valor numérico, que puede oscilar dentro de un cierto rango.

Este valor numérico proporciona una estimación o evaluación de un determinado aspecto en particular, por lo que, en ocasiones resulta conveniente una propuesta visual que ayude al usuario a comprender mejor lo que las métricas representan. Como se ha descrito, las métricas CBE son de tipo semántico, esto significa que necesitan una interpretación por parte del usuario, y es justo este aspecto el que se pretende con la visualización.

La Figura 5.16 ilustra la representación visual del acoplamiento de acuerdo a las métricas CBE. Esta técnica se basa en una distribución circular o radial de los elementos, donde la clase seleccionada se localiza en el centro. Todas las propiedades relacionadas con la clase, se distribuyen en el radial a lo largo de la circunferencia. En dependencia de si la propiedad contiene a la clase en el dominio o en el rango, se coloca en el lado izquierdo o en el lado derecho de la clase. Este tipo de relaciones vistas desde la perspectiva de la lógica matemática, así como el dominio y el rango, han sido descritas en la sección 2.1.

El acoplamiento CBE-in se puede ver en la Figura 5.16 para la clase etiquetada con el número 3, ya que esta clase se localiza en el dominio, mientras que la clase seleccionada se ubica a su lado derecho. En contraparte, el acoplamiento de la métrica CBE-out se muestra en la misma Figura 5.16 para las clases etiquetadas con los números 1 y 2, ya que estas clases se ubican en el rango. La visualización también puede opcionalmente desplegar, si se desea, la jerarquía interna sobre las clases. Esta representación de la jerarquía se muestra como una agrupación con las subclases dentro, como se ilustra en la misma Figura 5.16 con la clase etiquetada con el número 4. En el caso de que el dominio o el rango de alguna de las propiedades contenga más de una clase, la visualización representa una simple agrupación de estas clases, como se ilustra en la misma Figura 5.16.

La Figura 5.17 muestra el *zoom* semántico que representa el acoplamiento de la clase *Persona*, de la ontología *turisem.owl*, que se describe como caso de estudio más adelante.

Semánticamente se puede claramente distinguir lo que representa que la clase se localice en el dominio o en el rango de una propiedad. Análogamente, la interpretación de la visualización es de izquierda a derecha, por lo que la posición de la propiedad con respecto a la clase indica el tipo de acoplamiento. Para clarificar esta situación, la propiedad *personaGrupo* en la Figura 5.17 tiene a la clase *Persona* en el rango, por lo que la propiedad se representa en el lado izquierdo. Por el contrario, el resto de las propiedades contienen a la clase *Persona* en el dominio, por lo que se representan en el lado derecho de la misma. En esta Figura 5.17, se ha enfocado la propiedad *actividadPreferida*,

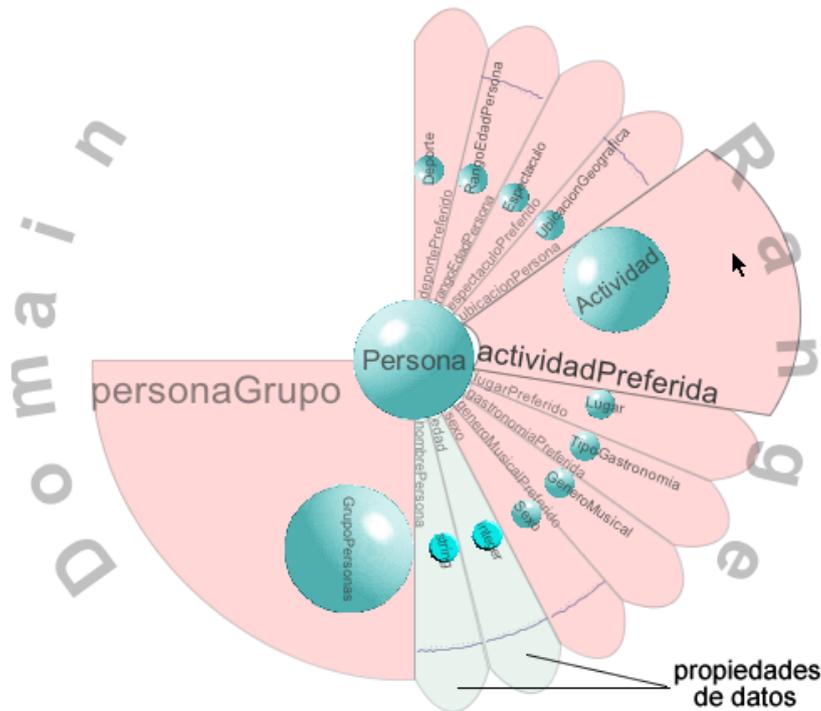


Figura 5.17: El *zoom* semántico sobre el acoplamiento de la clase *Persona*. En esta imagen se ha enfocado la propiedad *actividadPreferida*, que tiene a la clase *Persona* en el dominio y a la clase *Actividad* en el rango.

que tiene a la clase *Persona* en el dominio y a la clase *Actividad* en el rango. La interpretación semántica de esta propiedad sería que una persona tiene gusto por alguna actividad preferida.

Esta visualización se ha diseñado por la necesidad de tener una representación semántica del acoplamiento entre las clases. Como se ha descrito en la sección 2.1, el acoplamiento entre las clases está dado por las propiedades objeto, que representan relaciones con clases en el dominio y clases en el rango. Esta definición resulta mucho más fácil de entender, cuando se puede visualizar una representación semántica del acoplamiento. Este es el objetivo principal de esta propuesta de visualización, el poder presentar el acoplamiento, en lugar de simplemente limitarse a números calculados con fórmulas, como son las métricas.

5.3.3. Soporte a la fase de análisis

El análisis de los modelos ontológicos representa otro de los aspectos cruciales dentro de los procesos de gestión y reutilización de ontologías. Este proceso requiere que el usuario esté apoyado por el uso de herramientas, técnicas de interacción y visualización que permitan su desarrollo. Las actividades principales dentro de un proceso de análisis se muestran en la Figura 5.18, y son el análisis de la taxonomía de clases, el análisis de las propiedades de datos y las instancias, y el análisis del acoplamiento de las clases. Estas actividades se describen en detalle en las secciones 5.3.3.1, 5.3.3.2 y 5.3.3.3.

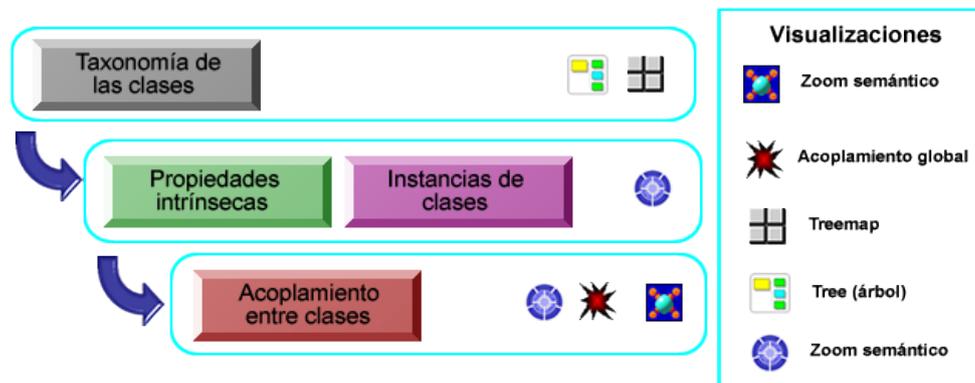


Figura 5.18: El proceso de análisis de un modelo ontológico está definido por cuatro actividades principales: el análisis de la taxonomía de las clases, el análisis de las propiedades de datos, el análisis de las instancias y el análisis del acoplamiento de las clases.

5.3.3.1. Análisis de la taxonomía de clases

El análisis de la taxonomía de las clases es el punto inicial dentro de un proceso de análisis de un modelo ontológico. Consiste en la evaluación de la jerarquía de clases, principalmente para la búsqueda de ciertos conceptos que pueden representar mayor interés en el modelo. Este análisis permite por un lado, identificar las clases existentes y sus elementos internos (propiedades e instancias) y, por otro lado, especialmente para procesos de reutilización, identificar aquellas clases no presentes en el modelo que es necesario agregar.

OWL-VisMod permite el análisis de la taxonomía de clases a través del uso de dos visualizaciones específicas, el *treemap* y el árbol de jerarquías. Ambas visualizaciones se han descrito en detalle en la sección 5.3.2.1, que describe la

fase de implementación del proceso de creación de ontologías.

La técnica de *treemap* (que se muestra en la Figura 5.9) permite, como se ha comentado previamente, recorrer los niveles de la taxonomía, con lo que se pueden localizar las clases que se buscan. Otro aspecto fundamental del *treemap* es que permite realizar el *zoom* semántico una vez que se ha localizado la clase buscada, para analizar sus detalles específicos.

La técnica del árbol de jerarquías (mostrada en la Figura 5.11) también permite analizar la taxonomía desde otra perspectiva diferente al *treemap*. Ambas visualizaciones son complementarias y son parte del diseño de OWL-VisMod para permitir al usuario diversos mecanismos para analizar los modelos ontológicos.

5.3.3.2. Análisis de las propiedades e instancias

Una vez identificada una clase específica que se quiera analizar, es necesario utilizar alguna técnica basada en *zoom* semántico para realizar este análisis. Esta fase de análisis se enfoca principalmente en dos aspectos: por un lado las propiedades intrínsecas en la clase, y por otro lado, las instancias de la clase. El primer caso es muy sencillo, y únicamente se limita al análisis de las propiedades definidas y los tipos de datos asociados. Para un análisis de este tipo, se puede utilizar la visualización de *zoom* semántico mostrada en la Figura 5.15.

El segundo caso representa un reto más interesante; cuando hay una base de conocimiento con instancias creadas, se requiere que el usuario pueda ser capaz de analizar y realizar consultas específicas sobre dicha base de conocimiento. Una solución a esta problemática es la implementación de mecanismos como las consultas visuales, definidas en la sección 3.2.3, que representan una de las técnicas de interacción en el área de la Analítica Visual.

Este tipo de análisis a través de consultas visuales se enfoca principalmente en las instancias de una base de conocimiento. Cada instancia en una clase representa un individuo específico que puede ser identificado unívocamente entre las demás instancias de la clase, de acuerdo a los valores en sus propiedades. Todas las instancias de una misma clase comparten las mismas propiedades, sin embargo, los valores internos en las propiedades permiten distinguir claramente a algún elemento del resto.

Para el análisis de las instancias en una clase, OWL-VisMod ha implementado una técnica de *zoom* semántico que permite realizar consultas sobre las instancias y los valores dentro de las propiedades. La Figura 5.19 ilustra dos

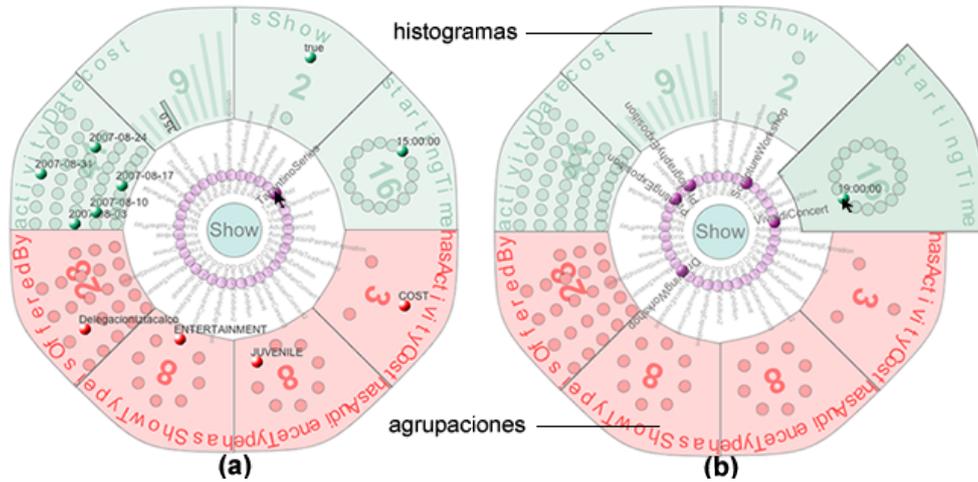


Figura 5.19: El *zoom* semántico usado para realizar consultas visuales (*visual querying*). La Figura (a) ilustra la consulta de todos los valores para cada una de las propiedades para la instancia seleccionada, mientras que la Figura (b) ilustra la consulta de todas las instancias que tienen un específico valor para una de las propiedades.

imágenes de la técnica de consultas visuales (*visual querying*). Por un lado, la Figura 5.19 (a) muestra la consulta sobre una instancia específica, de todos los valores internos para cada una de las propiedades de la clase. En este caso, la clase se refiere a espectáculos, en donde la instancia enfocada se refiere a todos los valores de las series del director Quentin Tarantino. La interacción con el usuario es simple e intuitiva; únicamente consiste en posicionar el puntero sobre alguna de las instancias, lo que provoca que la herramienta realice la consulta sobre todos los valores internos para cada una de las propiedades. Todos estos valores se resaltan para ser identificados claramente por el usuario en el análisis.

Por otro lado, la Figura 5.19 (b) ilustra el proceso inverso al mostrado en la Figura 5.19 (a). Al posicionar el puntero sobre alguno de los valores internos de las propiedades, todas las instancias que contienen dicho valor para esa propiedad se resaltan automáticamente. En este caso específico, se resaltan todas las instancias que representan espectáculos que comienzan a las 19:00.

Esta interacción permite al usuario analizar la base de conocimiento desde dos perspectivas, ya sea consultar todos los valores de una instancia en específico, o al revés, consultar todas las instancias que tienen un determinado valor para una propiedad.

Una de las ventajas de esta visualización es que distingue entre dos tipos de propiedades de datos: las numéricas y las nominales, en las que únicamente se considera el valor. Estos tipos de datos se visualizan de formas diferentes, ya que representan diferentes cosas; por un lado, los valores numéricos se representan con histogramas, por lo que el usuario puede visualizar todos los valores de acuerdo a su relación de orden. Los valores nominales, se representan como agrupaciones de elementos (*clusters*) ya que no hay relación de orden entre estos. Esta representación también puede verse en la Figura 5.19 (b), en donde se han señalado dos propiedades, una de las cuales es de tipo numérica y utiliza histogramas para establecer una relación de orden entre sus valores, y otra propiedad nominal que utiliza representaciones basadas en agrupaciones.

Otra ventaja de esta visualización es la escalabilidad como lo muestran las mismas Figuras 5.19 (a) y 5.19 (b), ya que se representan un total de ciento cincuenta y cuatro elementos sin saturar la vista. Los elementos mostrados son: una clase, cuatro propiedades objeto (en color rojo), cuatro propiedades de datos (en color verde), treinta y cinco instancias (en color púrpura) y ciento diez valores en las propiedades (esferas o histogramas dentro de las propiedades).

Una de las características que permiten la escalabilidad es la distribución de los elementos radialmente con la clase concéntrica, ya que establece implícitamente una relación entre estos sin necesidad de utilizar aristas que saturan la vista, como lo hacen las otras herramientas descritas anteriormente. Por otro lado, la ubicación de los valores dentro de los polígonos que representan a las propiedades, indica implícitamente una relación de propiedad. También, esta visualización hace uso de otras variables visuales como la forma, la opacidad y el uso de colores distintivos.

Un aspecto crucial en esta visualización, es precisamente la distribución física de los elementos, ya que permite representar las relaciones entre los distintos elementos involucrados de una forma implícita e intuitiva. La interacción de las consultas se realiza mediante el cambio en la tonalidad de los elementos para resaltar algunos elementos de otros.

5.3.3.3. Análisis del acoplamiento

Un aspecto fundamental en el proceso de análisis de un modelo ontológico, lo representa el análisis del acoplamiento de las clases que, como se ha descrito en secciones previas, está dado por las relaciones objeto. El análisis del acoplamiento semántico se realiza principalmente desde dos perspectivas. La primera se enfoca al acoplamiento en general del modelo ontológico, mientras que la segunda se enfoca en el acoplamiento de una clase específica [86].

La primera visualización se basa en la propuesta de una distribución radial de los elementos para la representación de las relaciones, en la cual se utiliza una representación del acoplamiento entre las clases, mediante el uso de ejes con una cierta distorsión de acuerdo con la propuesta denominada *Hierarchical Edge Bundles* [135]. Esta distorsión se ha utilizado en diferentes técnicas de visualizaciones jerárquicas, para que de forma inmediata resulte evidente cuáles son los nodos principales que se relacionan con mayor número de otros nodos. La implementación de esta técnica considera el valor de la métrica de acoplamiento CBE¹⁶ descrita en la sección 5.1.4 y definida en [84], para distribuir los elementos, de tal forma que las clases más acopladas se localizan en la parte superior del radial.

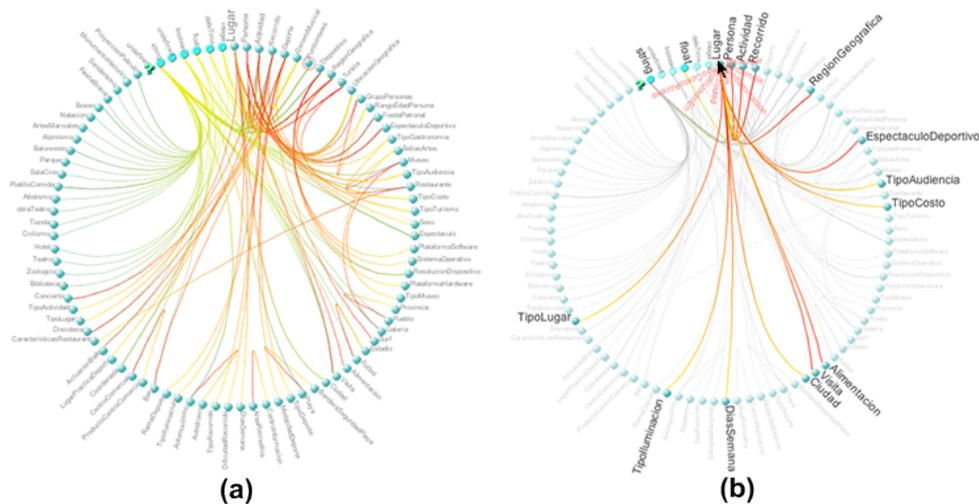


Figura 5.20: (a) La vista con el acoplamiento global de todas las clases en la ontología *turisem.owl*. (b) El acoplamiento de la clase llamada *Lugar*, la clase con más relaciones en este modelo ontológico.

Esta forma de distribuir los elementos permite fácilmente la identificación de las clases más acopladas, esto es, aquellas que tienen más relaciones, y que son las que generalmente representan los conceptos más importantes dentro del modelo ontológico.

La Figura 5.20 (a) muestra la vista del acoplamiento global de la ontología *turisem.owl*¹⁷. La visualización utiliza una distribución radial para acomodar las clases acopladas (únicamente aquellas que tienen relaciones), y por el centro se colocan las aristas que representan estas relaciones entre las diferentes

¹⁶Esta métrica es un contador de las relaciones de una clase

¹⁷Más imágenes y videos se encuentran disponibles en:
<http://www.analiticavisual.com/juan/snapshots3.html>.

El análisis del acoplamiento de una clase radica en conocer las propiedades objeto que semánticamente representan el acoplamiento, así como las clases relacionadas. El usuario puede navegar a través de la visualización para conocer el acoplamiento de cada una de las clases. Las relaciones de la clase enfocada se resaltan sobre las demás, lo mismo que las clases acopladas, como se muestra en la Figura 5.20 (b) con la clase *Lugar*. En este ejemplo en concreto, esta clase es la más acoplada del modelo ontológico, esto es, la que más relaciones de acoplamiento tiene en la ontología.

Otra perspectiva en el análisis se enfoca en las propiedades, con la finalidad de conocer las clases que están relacionadas a través de la propiedad. Análogamente a las clases, las propiedades se visualizan de acuerdo a su significado semántico, mediante una representación conceptual. La Figura 5.21 ilustra esta representación, en la que se muestran todos los elementos que caracterizan a una propiedad: primeramente el nombre de la propiedad, en este caso *sexo*, para definir el género de una persona (masculino o femenino), segundo, sus características intrínsecas (en este caso la propiedad es funcional), y tercero, una agrupación de clases (*cluster*) que indica el dominio de la propiedad y otra agrupación de clases para indicar el rango.

Para este ejemplo en específico, esta propiedad ha sido definida en la ontología *turisem_personas.owl*. Como puede verse en la Figura 5.21, la propiedad llamada *sexo* tiene a la clase *Persona* en el dominio y a la clase *Sexo* en el rango. El significado de esta propiedad se interpreta como que una *Persona* tiene definido una variable *sexo* para indicar el género (Masculino o Femenino).

Otra representación del acoplamiento de una clase se muestra en la Figura 5.22, en donde se utiliza la técnica de *zoom* semántico para visualizar el acoplamiento de una clase. Como se ha descrito en la sección 5.3.2.2, esta visualización representa el acoplamiento individual de una clase. En este caso en específico, las Figuras 5.20 (b) y 5.22, muestran dos vistas diferentes del acoplamiento de la clase *Lugar*. Por un lado, la Figura 5.20 (b) muestra el acoplamiento desde una vista general con todas las clases, mientras que la Figura 5.22 muestra el resultado de la vista de *zoom* semántico de la misma clase.

En resumen, el análisis del acoplamiento de los modelos ontológicos se puede realizar desde diversas perspectivas. Por un lado, una visualización general permite analizar el modelo completo, en donde las clases se distribuyen de acuerdo a la métrica que mide el acoplamiento, lo que permite al usuario fácilmente identificar las clases más acopladas que, generalmente, son las clases más importantes en un modelo ontológico.

Otro esquema de análisis es a través de una vista de *zoom* semántico sobre una clase específica. Esta vista, complementaria a la vista general, y permite

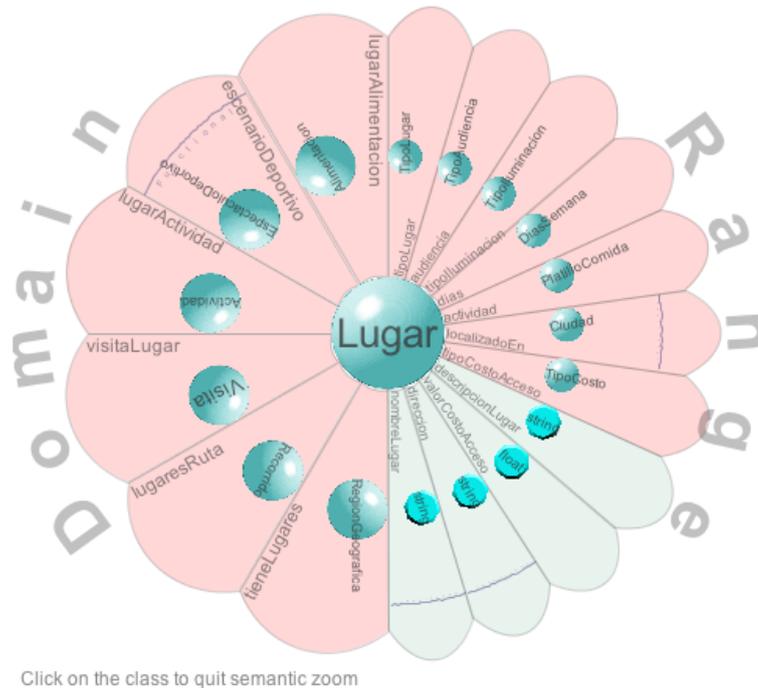


Figura 5.22: La vista del acoplamiento de la clase *Lugar*, desde la perspectiva del *zoom* semántico. Esta clase tiene trece propiedades objeto, y cuatro propiedades de datos.

analizar muy específicamente las propiedades y las clases acopladas con la clase seleccionada. Por otro lado, también se puede analizar el acoplamiento desde la perspectiva de una propiedad objeto, que define una relación entre clases.

5.3.3.4. Visualización de las métricas

Como se ha descrito en la sección 2.3.3, las métricas se definen principalmente por valores numéricos que permiten establecer una evaluación o estimación de alguna característica en un sistema de información, ya sea para realizar comparaciones entre diversos sistemas de información o para evaluar la calidad.

La gran mayoría de las herramientas que calculan métricas sobre las ontologías se limitan únicamente a desplegar una lista de los valores resultado. Esta situación se ha ilustrado en la Figura 2.3 de la sección 2.3.3 con dos

diversas variables visuales que permiten una expresividad visual robusta, como el uso de colores distintivos para las columnas, la implementación de la técnica de *Foco+Contexto* a través de la distorsión, así como de una nueva solución que despliega no solo el valor numérico, sino también los valores nominales, como se muestra en la Figura 5.23. Esta interacción también se considera dentro de la categoría de las consultas visuales, ya que la herramienta realiza un procesamiento de la información, en este caso, de los valores de las métricas de la clase seleccionada.

La Figura 5.23 muestra una vista de la aplicación del *Foco+Contexto* sobre una de las columnas, en este caso específico sobre las propiedades objeto de la clase *Persona*, que han sido previamente analizadas en la Figura 5.17. Esta interacción causa que la herramienta consulte todos los valores de la métrica enfocada, y los despliega en una lista. Esta nueva funcionalidad se ha agregado a la propuesta original del *table lens*, con la finalidad de enriquecer esta técnica para desarrollar las consultas visuales.

Otra funcionalidad agregada a esta técnica visual, es la posibilidad de filtrar datos de acuerdo a un criterio o un rango numérico, y también se ha implementado la opción de ocultar ciertas columnas. Estas dos nuevas funcionalidades implementadas, tienen el principal objetivo de mejorar el análisis de los datos representados en el *table lens*, ocultando información no requerida por el usuario.

OWL-VisMod implementa las métricas de acoplamiento definidas en la sección 5.1.4 y descritas en [84], que explican analíticamente el acoplamiento de una clase. Estas métricas permiten relacionar el acoplamiento desde el punto de vista semántico que una clase específica tiene con respecto a otras. También se calculan las métricas del esquema de la ontología, los totales de propiedades objeto, de propiedades de datos, subclases e instancias de una clase.

5.4. Conclusiones

Este capítulo ha definido una serie de flujos de trabajo para los procesos involucrados en el desarrollo del ciclo de vida de las ontologías (creación, gestión, análisis y reutilización). Estos flujos de trabajo se han definido tomando como base dos de las principales metodologías existentes, Methontology y DOGMA. Dentro de estos flujos de trabajo destacan tres fases de actividades: una fase de abstracción del dominio del problema y planteamiento de soluciones, una fase de análisis de un modelo ontológico existente y una fase de implementación del

modelo ontológico.

La primera fase de abstracción del dominio corresponde al usuario desarrollarla en su totalidad, ya que involucra principalmente los procesos de entendimiento del problema y la definición de la solución. Esta solución se pretende que siga el enfoque iterativo de Methontology y que plantee el diseño propuesto por DOGMA, en el que se propone el planteamiento de la solución, mediante el diseño de diversos submodelos ontológicos, que comienzan con un mayor grado de abstracción y derivan en submodelos más especializados en el dominio de aplicación. La finalidad de este planteamiento es la de generar desarrollos altamente reutilizables y desarrollos con reutilización, dos perspectivas generales de la reutilización de ontologías.

Las fases de análisis e implementación conllevan el uso de mecanismos que soporten las actividades definidas en los flujos de trabajo. Estos mecanismos deben enfocarse y diseñarse de tal forma que utilicen al máximo las capacidades cognitivas del usuario, lo que permite que este pueda llevar a cabo las diversas tareas involucradas, de forma relativamente fácil e intuitiva, lo que da como resultado, procesos más eficientes.

La herramienta que implementa estos mecanismos se ha diseñado mediante un enfoque diferente del resto de las herramientas disponibles, con el objetivo de corregir la problemática existente descrita en la sección 4.4 del capítulo 4.

Las soluciones propuestas abarcan dos grandes enfoques principalmente. Por un lado, el planteamiento de flujos de trabajo para los procesos de gestión de las ontologías, que definen una secuencia de actividades que permiten al usuario analizar o modelar las ontologías de una forma más eficiente. Por otro lado, la implementación de técnicas de visualización e interacción para soportar estas actividades propuestas. Todas estas técnicas se han definido completamente desde el enfoque en el que el usuario es el centro del proceso, análogamente al esquema de la Analítica Visual.

El planteamiento de los flujos de actividades en los procesos implica el desarrollo de nuevas estrategias diferentes a las que se utilizan actualmente. Estas estrategias conllevan el diseño de actividades y flujos de navegación, que eviten la saturación de los esquemas visuales con demasiada información para el usuario. En consecuencia, estos flujos de actividades permiten una navegación definida y un mayor control sobre la información que se muestra y la forma en que se hace. El resultado es un conjunto de procesos claramente definidos, en los que el usuario puede sacar el mayor provecho posible, para generar mejores diseños de modelos ontológicos.

Además, se ha propuesto un conjunto de nuevas métricas denominadas

CBE, que representan el acoplamiento individual de las clases en una ontología. Entre las ventajas de estas métricas se puede mencionar que se puede establecer una clasificación de las clases de acuerdo a los niveles de acoplamiento, o dicho de otra forma, el número de relaciones en las que está involucrada. La medida del acoplamiento de una clase identifica, en términos generales, los conceptos más importantes de un modelo ontológico, ya que las clases más acopladas casi siempre suelen representar los conceptos clave del dominio de aplicación. Estas métricas tienen principalmente un significado semántico que el usuario tiene que interpretar, para lo que se ha implementado una visualización específica que ayude en este aspecto.

En lo que respecta a las técnicas visuales y de interacción implementadas, se han utilizado diversas estrategias. Una de estas estrategias es la definición de un flujo que permite la navegación sobre las diferentes visualizaciones, partiendo desde un enfoque general, hacia objetivos muy específicos. Otra de las estrategias permite evitar la saturación visual, mediante la separación de funcionalidad e información, sin mezclar vistas generales con detalles específicos y viceversa. Otra estrategia es la implementación de mecanismos robustos de interacción con los usuarios, que permitan a este, seguir un enfoque basado en la filosofía de la Analítica Visual, para el modelado y análisis de modelos ontológicos.

Toda la funcionalidad implementada en la herramienta OWL-VisMod, y que se ha descrito en este capítulo, se corresponde al planteamiento de lo que corresponde a la primera versión de la propuesta. Alguna otra funcionalidad que puede enriquecer la herramienta, no se ha considerado para esta primera versión, sino que, puede ser considerada para versiones posteriores.

Tanto los flujos de trabajo definidos, como los mecanismos visuales e interactivos que se han implementado, permiten una gestión más eficiente, de los procesos involucrados en el ciclo de vida de las ontologías. Estas soluciones mejoran considerablemente las propuestas existentes, y permiten a los usuarios desarrollar eficientemente los procesos de la Ingeniería de Ontologías. Sin embargo, es necesario el desarrollo de evaluaciones tanto de funcionalidad como de usabilidad, con el objetivo primero de obtener retroalimentación por parte de los usuarios acerca de los mecanismos implementados y, segundo, de validar que, efectivamente, las soluciones implementadas mejoran con respecto a las propuestas existentes.

Capítulo 6

Evaluación de usabilidad

La Organización Internacional para la Estandarización (*ISO*) define la usabilidad como la *calidad de un producto que le permite ser utilizado por usuarios específicos, para alcanzar metas específicas, con efectividad, eficiencia y satisfacción, en un cierto contexto de uso específico (ISO/IEC 9241)*. La usabilidad es la capacidad o conveniencia que tiene un producto para ser utilizado, especialmente aquellos productos que interactúan con las personas [179].

En el contexto específico de los sistemas para la representación de conocimiento es fundamental realizar diversas evaluaciones de usabilidad, que aseguren que estos sistemas sean, por un lado, fáciles de aprender, pero también efectivos, funcionales y eficientes desde la perspectiva del usuario. Todos estos aspectos permiten asegurar la usabilidad de los productos de *software* en general y, en particular, también de los sistemas basados en conocimiento.

Existe una disciplina llamada Ingeniería de la Usabilidad [203] que provee métodos estructurados para alcanzar la usabilidad en el diseño de interfaces de usuario, durante el desarrollo del producto. Una de las partes en este proceso de desarrollo es la evaluación de la usabilidad. En dependencia del tipo de sistema e interacción con el usuario se han definido diversos métodos de evaluación [234]. Uno de estos métodos es una evaluación centrada en el usuario (*user-centered evaluation*) [179, 235], en la que los usuarios son la principal fuente de retroalimentación.

Este capítulo describe una evaluación de usabilidad centrada en el usuario, que permite validar, por un lado, las actividades en los flujos de trabajo para los procesos involucrados en el ciclo de vida de las ontologías, descritas en la sección 5.1, mientras que por otro lado, las pruebas de usabilidad también permiten validar las técnicas de visualización e interacción que soportan las tareas

definidas en las fases de análisis e implementación de ontologías, descritas en la sección 5.2. Esta evaluación de usabilidad de la herramienta OWL-VisMod se ha desarrollado en dos partes; la primera parte corresponde al desarrollo de las tareas para la fase de análisis, y la segunda parte corresponde al desarrollo de las tareas propias de la fase de implementación de las ontologías. La sección A contiene un manual de usuario que describe la funcionalidad de la herramienta.

El capítulo comienza con una descripción de los criterios utilizados para desarrollar esta evaluación, continúa con la evaluación de usabilidad en las fases de análisis e implementación, y termina con las conclusiones.

6.1. Pruebas de Usabilidad

Las actuales propuestas metodológicas no cubren suficientemente la forma de desarrollar pruebas de usabilidad sobre modelos ontológicos. OntoMetric y OntoClean, descritas en la sección 2.3.1, proporcionan algunos mecanismos para evaluación de ontologías, pero no desde la perspectiva de la usabilidad. Vrandeic [270] propone un marco de trabajo para la evaluación de ontologías, pero desde el punto de vista semántico. Esta propuesta se basa en una meta-ontología, o dicho de otra forma, una ontología de ontologías.

García et al. [98] describen un enfoque basado en ontologías para el diseño de cuestionarios de evaluación de usabilidad de sitios web, mientras que en [97] los mismos autores describen un ejemplo de una evaluación de usabilidad a los editores de ontologías. Esta evaluación se basa por un lado, en el análisis heurístico propuesto en [202], y por otro lado, en dos planteamientos generales: *¿Qué tan fácil es crear una nueva ontología con los actuales editores?*, y *¿Qué tan fácil es editar, buscar y actualizar tareas en grandes ontologías con los actuales editores?*

Para la evaluación de la usabilidad en OWL-VisMod se ha seguido el mismo enfoque de los planteamientos de García et al. [97] combinado con la teoría del área de Ingeniería de Usabilidad. Dentro de esta parte teórica, básicamente se ha tomado el enfoque de una evaluación de usabilidad centrada en el usuario [234], mediante la aplicación de cuestionarios a los participantes de la evaluación, de forma similar a algunas otras evaluaciones de ontologías [41, 8, 218].

La técnica de evaluación heurística es el método más ampliamente utilizado, ya que las evaluaciones centradas en el usuario suelen ser menos costosas en cuanto a tiempo. Más aún, es una forma muy directa de obtener una retroalimentación directa de los usuarios. Nielsen [204] propone el uso de una técnica de evaluación heurística, que requiere un conjunto de especialistas que evalúan

y juzgan que una cierta interfaz de usuario, cumpla con los principios del diseño de usabilidad. Scholtz [234] resume los diversos tipos de evaluaciones, entre los que se encuentran las evaluaciones basadas en expertos y las evaluaciones basadas en modelos.

Este tipo de evaluación centrada en el usuario es consecuencia de la propia naturaleza de la herramienta, ya que como se ha descrito previamente, el diseño de OWL-VisMod está completamente orientado al usuario como el centro de los procesos. Todas las técnicas han sido implementadas tomando en consideración la capacidad cognitiva de una persona, por lo que la evaluación de usabilidad también debe considerar a la persona como el centro del proceso. La diferencia con la evaluación basada en expertos es que este último esquema involucra un reducido número de usuarios con un alto grado de conocimiento y muy específico, mientras que el esquema basado simplemente en los usuarios, involucra a un mayor número de usuarios sin un alto grado de especialización.

La usabilidad de un producto de *software* es un aspecto fundamental que se evalúa a través de pruebas que involucran la interacción de los usuarios con el sistema. La evaluación de este tipo de pruebas se lleva a cabo generalmente mediante el uso de cuestionarios con una serie de preguntas específicas sobre la experiencia del usuario con el sistema. A través de las respuestas de los usuarios, los diseñadores del sistema pueden advertir en una etapa temprana de la fase de desarrollo, sobre los posibles problemas y conflictos que pudieran haber pasado inadvertidos.

Esta evaluación de usabilidad sobre OWL-VisMod se ha enfocado en la efectividad en el uso de las técnicas visuales y de interacción, que soportan las tareas definidas en las fases de análisis e implementación de ontologías, ya que estas fases son las que requieren el uso de la herramienta. También se ha considerado en la evaluación, la satisfacción del usuario en general, así como sus comentarios sobre diversos aspectos a mejorar.

Para evaluar OWL-VisMod se aplicó, como se ha mencionado, una evaluación centrada en el usuario. Este tipo de evaluaciones se basan en las siguientes actividades: la identificación de los usuarios representativos, la identificación y desarrollo de las tareas representativas y, finalmente, el desarrollo de un cuestionario que permita al usuario otorgar una evaluación de los diversos aspectos de la herramienta.

La primera actividad consiste en la identificación de aquellos usuarios representativos que participan en la evaluación. Para el desarrollo de esta tarea es necesario determinar quiénes son los usuarios representativos de OWL-VisMod. Este segmento de usuarios está formado por todos aquellos, principalmente ingenieros en ontologías, interesados en crear, reutilizar, analizar o modificar

bases de conocimiento para aplicaciones semánticas. En este contexto, los usuarios de la herramienta son aquellos con un cierto grado de especialización, o al menos con cierto conocimiento de modelos ontológicos.

Para el desarrollo de esta evaluación se utilizó como segmento de usuarios veintiún estudiantes del Máster Universitario en Sistemas Inteligentes impartido en la Universidad de Salamanca. A estos estudiantes se les impartió una charla relacionada con la web semántica, las ontologías y, finalmente, se les mostró la herramienta y se les explicaron algunos detalles del diseño de las visualizaciones, así como una introducción general de la interacción con los usuarios. El grado de conocimiento de los estudiantes sobre ontologías no era muy especializado, sin embargo, al ser estudiantes de Informática, todos ellos tenían conocimiento sobre modelos de datos e Ingeniería de *Software*.

Una vez definido el segmento de los usuarios, la segunda actividad en el esquema de evaluación corresponde a la identificación de las tareas representativas. Estas tareas deben estar siempre enfocadas a los aspectos específicos que intentan evaluarse; en este caso, los tres aspectos más importantes a evaluar son la efectividad, la eficiencia y la interacción con el usuario. La efectividad es un objetivo muy general y se refiere a lo bueno que es un producto al hacer lo que se supone que tiene que hacer. Por otro lado, la eficiencia se refiere a la forma en la que un producto apoya a los usuarios en el desarrollo de sus tareas [179, 235]. Finalmente, la interacción con el usuario se refiere a los mecanismos con los que el usuario puede ejecutar acciones sobre el modelo visual, lo que representa un aspecto fundamental en cualquier sistema de *software*, y especialmente en OWL-VisMod.

La evaluación fue dividida en dos partes: la primera parte consistió en el análisis de una ontología públicamente disponible, con la finalidad de que el usuario se familiarizara con la herramienta, y también para probar el rendimiento de las visualizaciones con una gran cantidad de datos. La segunda parte correspondió al desarrollo de todo el proceso de creación de un modelo ontológico, basado en las actividades del flujo de trabajo descrito en la sección 5.1.1.

6.2. Análisis de un modelo ontológico

Para el desarrollo de esta fase, se utilizó la ontología SAO v1.2¹ (*Subcellular Anatomy Ontology*), de libre distribución y descrita en [75]. Esta ontología describe la anatomía subcelular del sistema nervioso, más específicamente, las

¹<http://ccdb.ucsd.edu/SAO/1.2/SAO.owl>

células nerviosas, sus partes y la interacción entre esas partes. Esta ontología ha sido construida usando Protégé 3.2.x, de acuerdo a la especificación OWL DL, con un total aproximado de ochocientas clases.

La fase de análisis de este modelo se ha llevado a cabo de acuerdo a las actividades descritas en el esquema de la Figura 5.18 en la sección 5.3.3. Comienza con un análisis de la jerarquía de clases en el modelo ontológico. Esta jerarquía es bastante grande en amplitud y profundidad como lo muestra la Figura 6.1, que ilustra una parte del árbol de jerarquías del modelo ontológico.

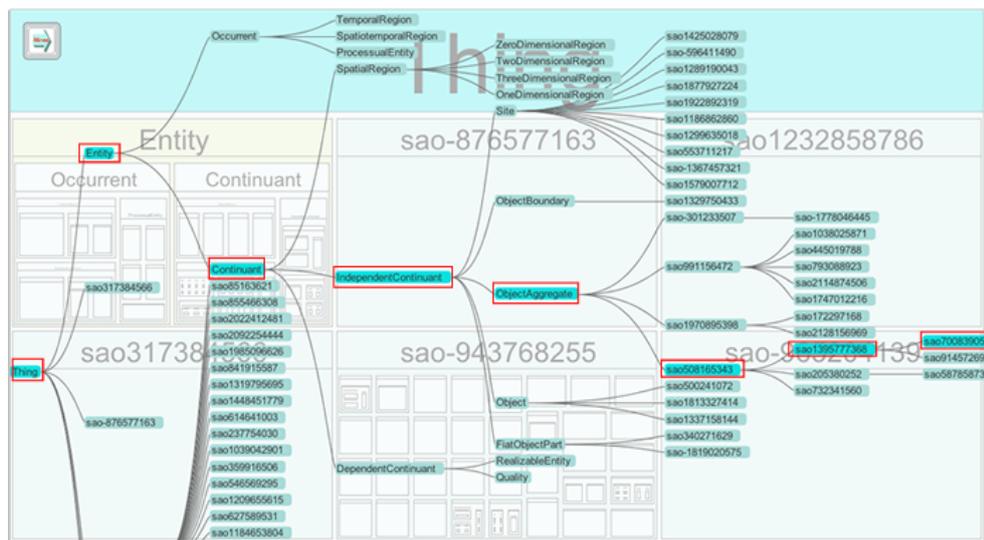


Figura 6.1: Se muestra la jerarquía de una clase en el octavo nivel de la ontología SAO. Se ilustra toda la jerarquía de las superclases en los niveles previos.

La fase de análisis implica navegar a través de la jerarquía de clases en la búsqueda de ciertos conceptos específicos. Para el caso específico de profundas jerarquías como SAO, OWL-VisMod implementa mecanismos que permiten desarrollar eficientemente esta búsqueda. La Figura 6.1 muestra la forma en la que el árbol de jerarquías permite recorrer todos los niveles de la taxonomía, en este caso, se ha desplegado una jerarquía con ocho niveles de profundidad.

Como parte de la evaluación, a los usuarios se les pidió primero, analizar la jerarquía a través del *treemap* y del árbol de jerarquías. La interacción consistió en la navegación de los diversos niveles, así como la búsqueda de algunos conceptos específicos, de forma similar a como sería un análisis real.

El análisis continúa con las técnicas de *zoom semántico*, para las propiedades intrínsecas y las instancias. En este caso específico, este modelo ontológico no tiene instancias definidas, por esta razón, no puede utilizarse la técnica de

las consultas visuales.

Esta ontología tiene bajo nivel de acoplamiento, ya que únicamente treinta y cinco de las clases tienen acoplamiento, con un total de ochenta y una propiedades, de las cuales treinta y seis son propiedades objeto y cuarenta y cinco propiedades de datos.

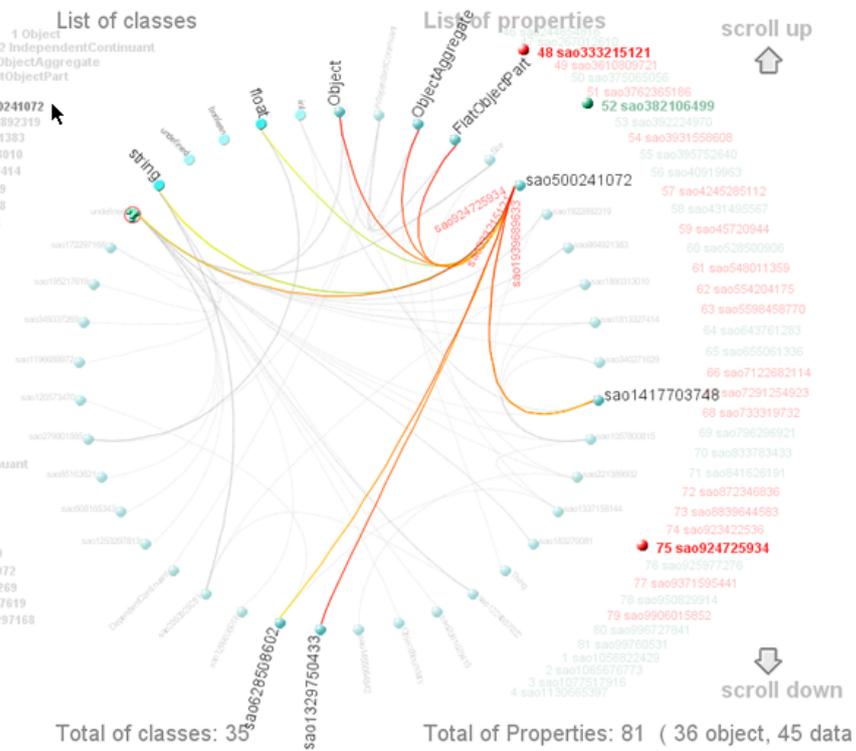


Figura 6.2: Se muestra el acoplamiento global de la ontología SAO, con treinta y cinco clases con relaciones de acoplamiento, ochenta y una propiedades, de las cuales, treinta y seis son propiedades objeto y cuarenta y cinco propiedades de datos.

La Figura 6.2 muestra la visualización con el acoplamiento global de esta ontología. El análisis sobre el acoplamiento global, se realiza ya sea, a través de las clases, o a través de las propiedades listadas al lado derecho en la visualización. En el caso específico de la Figura 6.2, en donde se ha seleccionado la clase *sao500241072* para ilustrar el análisis de acoplamiento, se pueden distinguir seis clases acopladas *Object*, *ObjectAggregate*, *FlatObjectPart*, *sao1417703748*, *sao1329750433* y *sao628508602*. Esta clase también tiene propiedades de tres tipos de datos diferentes, de tipo *float*, *string* y propiedades de tipo indefinido.

6.3. Creación de un nuevo modelo ontológico

La creación de un nuevo modelo ontológico constituye el aspecto más importante a evaluar, para lo cual, se pidió a los usuarios el diseñar un modelo de conocimiento basado en el dominio académico universitario. Este dominio fue seleccionado con la finalidad de que el desarrollo del modelo resultara fácil y además conocido por todos, ya que al ser estudiantes todos ellos podían llevar a cabo dicha tarea.

Cada usuario pudo desarrollar el modelo acorde a su propio conocimiento y con la libertad plena de incluir los conceptos, atributos, relaciones e instancias que considerara necesarios. No había una limitante establecida en cuanto al número de elementos, sino que la única limitante era el factor tiempo destinado a la evaluación. En esta medida, los modelos ontológicos de los estudiantes resultaron diferentes, aunque la mayoría de ellos compartían los conceptos principales como *Estudiante*, *Profesor*, *Universidad*, *Asignatura*, *Facultad* y *Curso Académico*.

La figura 6.3 muestra la visualización de árbol (*tree*) con la jerarquía de uno de los modelos ontológicos creado por alguno de los usuarios. Se ha resaltado en un recuadro en color rojo, la especialización de la clase *Persona*, en las clases *Profesor*, *Investigador*, *Estudiante* y la clase *ProfesorInvestigador*, definida como la unión de las clases *Profesor* e *Investigador*.

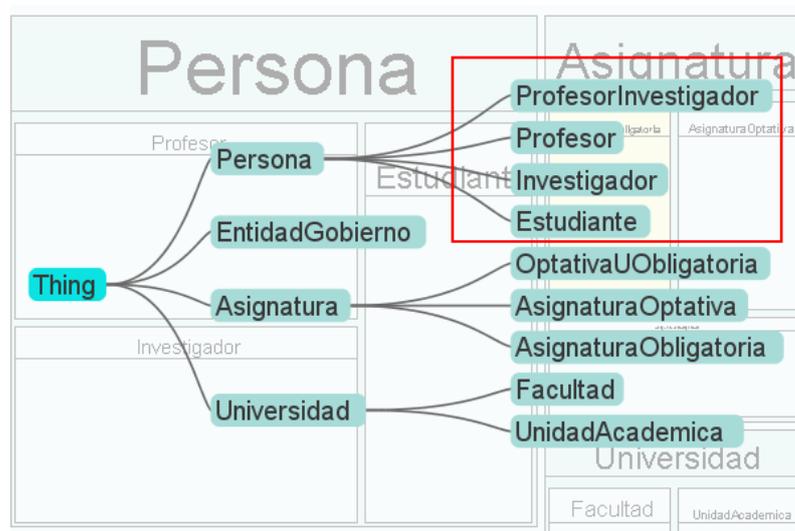


Figura 6.3: Árbol con la vista de la representación jerárquica de las clases en uno de los modelos ontológicos creado por un usuario para la evaluación de usabilidad.

Por otro lado, la Figura 6.4 muestra una vista del *zoom* semántico del mismo modelo mostrado en la Figura 6.3. Esta vista muestra los diversos aspectos del *zoom* semántico y el desarrollo de la técnica de consultas visuales. La clase *Profesor* contiene siete instancias que se ubican alrededor de la esfera central que representa a la clase. Cuando el usuario se centra en una de las instancias, la herramienta consulta cada uno de los valores de esta instancia y se resaltan en las propiedades.

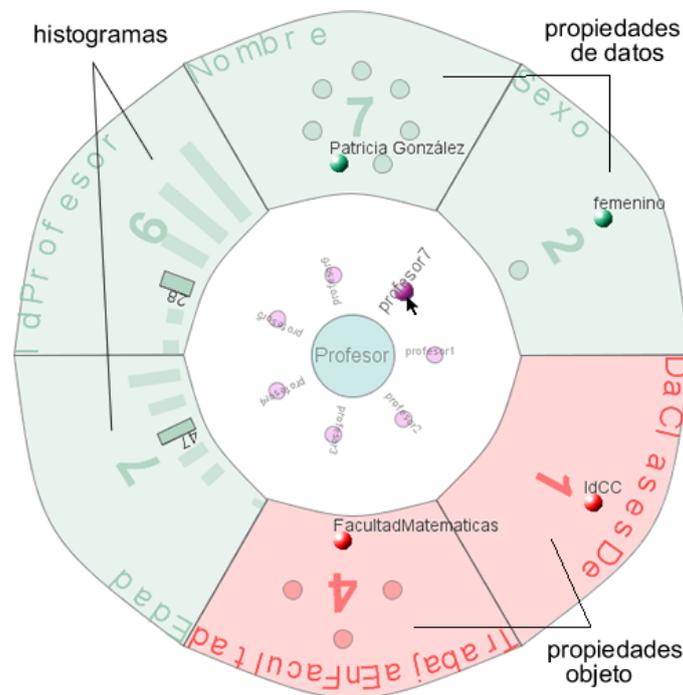


Figura 6.4: Vista detallada del *zoom* semántico de uno de los modelos creado por los usuarios. La visualización incluye el uso de la técnica de consultas visuales, en la que se muestra una consulta de una instancia de la clase *Profesor* y los valores de esta instancia para cada una de las propiedades.

En este caso, al consultar sobre una de estas instancias, todos los valores dentro de las propiedades se resaltan y, como se ha detallado en la sección 5.3.2.2, las propiedades de datos de tipo numérico (como el identificador y la edad), se representan como histogramas que establecen una relación de orden entre todos los valores.

6.4. Evaluación de los procesos definidos

Una vez que los usuarios crearon su modelo ontológico, la siguiente fase del proceso de evaluación consistió en dar respuesta a un cuestionario con un total de veintidós preguntas cerradas y cuatro preguntas abiertas. Las preguntas cerradas fueron definidas en cuatro grupos y organizadas de acuerdo al mismo flujo de trabajo. Estos cuatro grupos de visualizaciones son: la taxonomía de conceptos, el zoom semántico, acoplamiento de las clases y el cuarto grupo evalúa la satisfacción general del usuario con respecto a toda la herramienta. Para la evaluación de las preguntas cerradas se ha utilizado una escala de uno a cinco, en la que uno representa la peor evaluación y cinco la mejor. Todas las preguntas del cuestionario se describen en los siguientes apartados.

6.4.1. Taxonomía de la ontología

1. ¿En tu opinión, la jerarquía de conceptos está suficientemente bien representada usando *treemaps* y árboles de jerarquías?
2. ¿Consideras que el *table lens* es útil para dar una vista general de los elementos en el modelo ontológico?
3. ¿Consideras que el proceso de modelado es suficientemente claro e intuitivo para construir una ontología OWL?
4. ¿Consideras que estas visualizaciones (*treemap* y *tree*) son fáciles de aprender, usar y entender?
5. ¿Cuál sería tu grado de satisfacción con esas visualizaciones?
6. ¿Qué comentarios adicionales quieres agregar?

6.4.2. Zoom semántico

1. ¿Consideras el *zoom* semántico suficientemente intuitivo para mostrar los elementos internos de una clase seleccionada?
2. ¿Consideras que el modelo de navegación es fácil de entender?
3. ¿Es suficientemente claro para qué han sido creadas esas visualizaciones?
4. ¿Cuál sería tu grado de satisfacción con estas visualizaciones?
5. ¿Qué comentarios adicionales quieres agregar?

6.4.3. Acoplamiento de las clases

1. ¿Consideras la visualización de acoplamiento suficientemente clara para representar las relaciones entre las clases en una ontología?
2. ¿Consideras que la navegación en esta visualización es fácil?
3. ¿Consideras que la visualización del zoom semántico es suficientemente clara para representar el acoplamiento de alguna clase seleccionada?
4. ¿Consideras estas visualizaciones fáciles de aprender, usar y entender?
5. ¿Cuál es tu grado de satisfacción con estas visualizaciones?
6. ¿Qué comentarios adicionales quieres agregar?

6.4.4. Satisfacción general de OWL-VisMod

1. ¿Consideras la interacción con el usuario suficientemente intuitiva?
2. ¿Consideras la ayuda del usuario suficientemente clara y comprensible?
3. ¿Es el flujo de navegación coherente y correcto desde tu punto de vista?
4. ¿Consideras que las visualizaciones satisfacen los propósitos para los que fueron creadas?
5. ¿Qué tan fácil consideras que sería aprender a usar esta herramienta?
6. ¿Consideras que las visualizaciones en general son intuitivas y claras?
7. ¿Consideras que la herramienta es suficientemente capaz de modelar ontologías OWL?
8. ¿Cuál es tu grado de satisfacción con la herramienta?
9. ¿Qué comentarios adicionales quieres agregar?

6.5. Resultados de la evaluación

Como se ha mencionado previamente, se definieron cuatro grupos de preguntas descritas en las secciones 6.4.1, 6.4.2, 6.4.3 y 6.4.4. El análisis de los resultados se basa en el cálculo del valor de la media aritmética de todas las evaluaciones de los usuarios a cada una de las preguntas. Este tipo de evaluación

aunque sencilla, es bastante robusta ya que permite reconocer las preguntas peor evaluadas por parte de la mayoría de los usuarios. Todas las preguntas detectadas con un valor de media relativamente bajo, son las que en mayor medida hubo que analizar para poder corregir estos aspectos específicos.

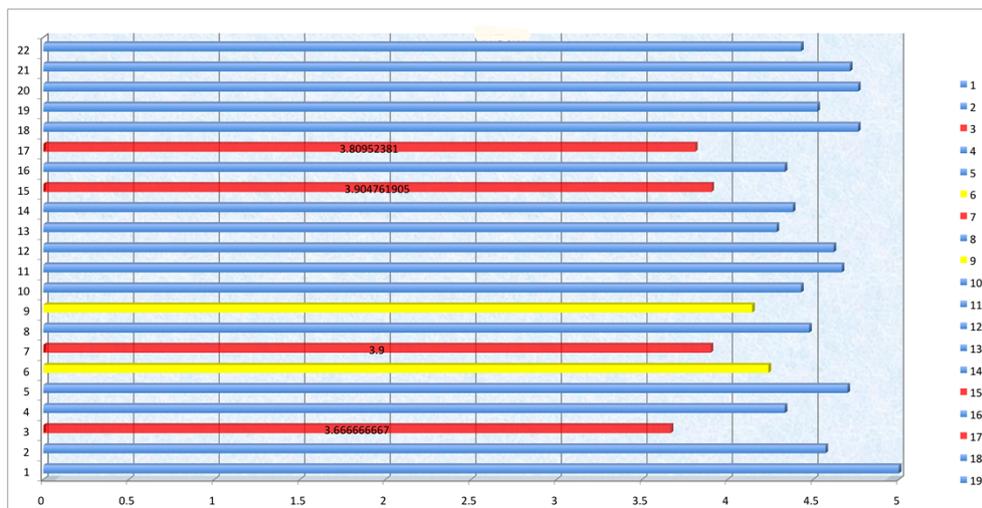


Figura 6.5: La gráfica de los resultados de las preguntas cerradas, se resaltan en color rojo aquellas preguntas en las que ha habido que poner mayor atención y en amarillo otras preguntas que también se consideraron como importantes.

En términos generales los resultados de la evaluación son aceptables, los valores de las medias son en general altos, lo que indica que los objetivos generales de las visualizaciones se han cumplido, sobre todo el hecho de ofrecer una herramienta intuitiva, fácil de usar y robusta para el modelado de ontologías. Los resultados de la evaluación se muestran en la Figura 6.5. Las barras resaltadas en color rojo y en amarillo representan aquellas preguntas con los valores de evaluación más bajos.

La primera de las preguntas marcadas en color rojo corresponde a la tercera barra en el histograma, que se refiere a que si el proceso de modelado es suficientemente claro para construir una ontología OWL. De acuerdo a las respuestas de los usuarios en las preguntas abiertas, se pudo obtener en conclusión que algunos de ellos consideraban que el proceso de modelado no era un asunto intuitivo a primera instancia. Es decir, que los usuarios requerían una breve explicación antes de comenzar a utilizar la herramienta, acerca del flujo de trabajo y el desarrollo del proceso de modelado. Este aspecto es importante considerarlo, aunque el hecho de que los usuarios no habían creado ontologías previamente a su uso con OWL-VisMod es un factor que repercute en sus respuestas.

Por otro lado, las preguntas seis, siete y nueve se refieren a las visualizaciones de *zoom* semántico; más específicamente, preguntan si son suficientemente intuitivas y el modelo de navegación fácil de entender. En este sentido, la retroalimentación obtenida a través de las preguntas abiertas fué principalmente el hecho de que no era suficientemente clara la navegación desde el *treemap* hacia el *zoom* semántico, especialmente el regreso a la vista general. También hubo comentarios en el sentido de que era poco clara, la interacción de la funcionalidad en la herramienta.

Esta problemática se ha resuelto incluyendo pequeños mensajes que indican los mecanismos de interacción con las visualizaciones, así como con la mejora de la ayuda. La navegación e interacción con todas las visualizaciones en general es un aspecto que requiere, como en cualquier herramienta que se utiliza por primera vez, de un proceso de aprendizaje que, sin embargo, no resulta complejo ya que utiliza los mecanismos comunes de interacción, como selecciones con el puntero, arrastrar y soltar (*drag and drop*), elementos visuales o funciones de *scrolling*.

Las preguntas etiquetadas con los números quince y diecisiete hacen referencia al flujo de navegación y la interacción con el usuario en toda la herramienta. En este caso, los motivos son los mismos que en el caso de la interacción y el flujo descrito anteriormente para el *zoom* semántico. Las soluciones también han sido mejorar la ayuda general de toda la herramienta, así como incluir mensajes de información que describen la interacción, y el uso de *tooltips* sobre los elementos.

Por lo que respecta a las preguntas abiertas, primeramente han permitido a los usuarios describir claramente los aspectos que consideran como mejorables o poco intuitivos respecto a las preguntas cerradas. En segundo lugar, también han permitido conocer otros aspectos no considerados en las preguntas cerradas que podían mejorarse de acuerdo a la perspectiva de los usuarios. Algunos ejemplos de estos comentarios fueron las posibles mejoras en el aspecto de ciertos componentes gráficos (*look and feel*), la inclusión de histogramas para la representación de tipos de datos (sugerencia implementada posteriormente), así como algunas propuestas de nueva funcionalidad. También la evaluación permitió descubrir algunos errores de programación (*bugs*), no detectados previamente y que fueron solucionados.

6.6. Conclusiones

La usabilidad de un sistema de *software* interactivo representa un aspecto fundamental a evaluar, especialmente durante la última fase de desarrollo del

mismo. Este capítulo describe el desarrollo de una prueba de usabilidad de OWL-VisMod con un grupo de estudiantes de máster, cuyo objetivo principal era obtener una retroalimentación de los usuarios, sobre todos aquellos aspectos que pudieran mejorarse con respecto a la interacción con el usuario. La evaluación tenía como primer objetivo validar el uso de la herramienta con las tareas definidas para llevar a cabo las fases de análisis e implementación de ontologías. Esta validación implica el hecho de que la herramienta tuviera la capacidad para desarrollar todas esas tareas.

Debido a que el usuario está siempre en el centro de las tareas involucradas con el uso de la herramienta, la evaluación se basa en un enfoque centrado en el usuario. El alcance de la evaluación comprende por un lado, el análisis de un modelo existente y, por otro lado, la creación de un nuevo modelo ontológico que resultara sencillo para los usuarios.

El cuestionario de evaluación se ha segmentado de acuerdo a los mismos grupos de visualizaciones definidos en la herramienta y, finalmente, un grupo de preguntas globales. En términos generales, la evaluación ha arrojado buenos resultados sobre los objetivos planteados. Los usuarios han sido capaces de analizar un modelo y crear uno nuevo desde el comienzo.

La evaluación ha permitido obtener retroalimentación en dos aspectos principalmente: el primero con la detección de algunos errores de programación, y el segundo con detalles específicos de navegación e interacción en los que los usuarios necesitaban una mayor descripción. Tanto los errores de programación detectados, como los comentarios sobre la interacción y navegación, han sido considerados e implementada la solución.

Una vez realizada la evaluación de usabilidad, el siguiente paso es la evaluación de funcionalidad de la herramienta, con un caso de estudio que permita evaluar las capacidades que tiene OWL-VisMod para cubrir todos los aspectos de un caso de estudio real. Esta evaluación se describe en el capítulo siguiente.

Capítulo 7

Caso de Estudio

Desde el punto de vista de las tecnologías de la información, la funcionalidad es el conjunto de características y capacidades de una aplicación de *software*, que permiten a los usuarios el uso del sistema para desarrollar diversas tareas. La funcionalidad de cualquier sistema de *software* es un aspecto crucial que tiene que ser evaluado antes de que el sistema sea puesto en un ambiente de producción.

Este capítulo está dedicado a describir la evaluación de la funcionalidad de OWL-VisMod, como complemento a la evaluación de usabilidad descrita en el capítulo 6 y en [91, 89]. A lo largo de este, se describe un caso de estudio para evaluar la funcionalidad de la herramienta con un proyecto que involucra el uso de ontologías en una aplicación para dispositivos móviles. Algunos vídeos y las imágenes en alta resolución se encuentran disponibles en <http://www.analiticavisual.com/juan/thesis.html>. Este capítulo comienza con la descripción del problema, la solución planteada, la descripción de la fase de implementación de los modelos ontológicos y, finalmente, se describen las conclusiones.

7.1. Proyectos Turimov y Ulises

El uso de modelos semánticos para sistemas de recomendación [1] en dominios turísticos es un área que se ha venido desarrollando desde la última década, con una serie de propuestas al respecto [74, 78, 154].

Este caso de estudio describe el proceso de desarrollo de los modelos ontológicos para dos proyectos: el primero llamado *Turimov*, un sistema in-

teligente de soporte para dispositivos móviles para el fomento del turismo en la región de Castilla y León, mientras que el segundo se denomina *Ulises*, un sistema inteligente de soporte móvil para la toma de decisiones y recomendaciones en el sector turístico en general. El objetivo principal de estos proyectos es el desarrollo de un sistema basado en el uso de diversas plataformas tecnológicas móviles para apoyar el desarrollo de actividades que ayuden a los turistas en su búsqueda de lugares, eventos, actividades, etc., durante sus viajes en la región de Castilla y León para el caso del proyecto *Turimov*, y su respectiva versión más general fuera de esta región, el proyecto *Ulises*. Más detalles específicos del proyecto pueden encontrarse en el sitio web de la empresa Interra¹.

Los proyectos *Turimov* y *Ulises* se basan en el uso de ontologías OWL para administrar toda la información, acerca de lugares, eventos o actividades, así como los perfiles de los usuarios, para que el sistema de recomendación pueda sugerir viajes o itinerarios específicos de acuerdo a los perfiles de los usuarios. Toda la información se almacena y administra usando ontologías, y estas juegan un papel crucial en el proceso de desarrollar inferencia sobre los datos para crear diversas sugerencias de viajes. El caso de estudio para el desarrollo de los proyectos *Turimov* y *Ulises* sigue el flujo de trabajo descrito en la sección 5.3.2 e ilustrado en la Figura 5.8. Ambos proyectos comparten la misma definición de los modelos ontológicos.

7.1.1. Dominio del problema

De acuerdo al flujo de trabajo, la primera fase consiste en la definición y abstracción del problema, así como el planteamiento de la solución. Esta fase se ha cubierto por completo en la definición de los respectivos proyectos, en donde se han establecido claramente los objetivos, las especificaciones, así como el alcance global de los proyectos. Este proceso se ha desarrollado enteramente por especialistas de la empresa Interra, en donde la descripción del proyecto indica que se trata del desarrollo de un sistema de toma de decisiones y recomendaciones para el sector turístico basado en web semántica. El objetivo de ambos proyectos es prácticamente el mismo, se basa en el desarrollo de un sistema personalizado según el perfil del usuario y la información del contexto. La arquitectura debe ser compatible con las plataformas existentes de comercio electrónico y los estándares tecnológicos corrientes basados en componentes y multicapa.

Una vez definido el alcance, los objetivos y los requisitos tecnológicos del sistema por parte de la empresa, el acuerdo con Interra establece que como caso

¹<http://www.interra.es/>

de estudio para la validación de la funcionalidad de OWL-VisMod se realizará el modelo ontológico de dicho sistema. El desarrollo de este modelo ontológico se ha dividido en tres submodelos de acuerdo a los objetivos del dominio. El primer submodelo se enfoca en los usuarios del sistema, y se basa en la recopilación de las características de las personas para establecer perfiles, de acuerdo a los cuales el sistema de recomendación proporciona diferentes rutas, lugares o actividades. El segundo modelo se utiliza para establecer una clasificación de los dispositivos móviles, de acuerdo a sus principales características. Finalmente, el tercer modelo se utiliza para definir todos los recursos en el sistema, como las actividades, espectáculos, los diferentes lugares, recorridos o rutas de senderismo, etc.

El primer modelo (*turisem_personas.owl*²) contiene los conceptos y las propiedades que definen el perfil de una persona, sus gustos en gastronomía, actividades deportivas, culturales, espectáculos, lugares preferidos, etc. El objetivo de este pequeño modelo ontológico es el de capturar las características de los usuarios, para que el sistema pueda recomendar diferentes actividades de acuerdo a estas.



Figura 7.1: Los submodelos ontológicos de persona (*turisem_personas.owl*) y dispositivos (*turisem_dispositivos.owl*). Las principales clases son *Persona*, *Dispositivo*, *PlataformaSoftware* y *PlataformaHardware*.

El segundo modelo (*turisem_dispositivos.owl*³) contiene las características

²http://www.analiticavisual.com/juan/archivos/turisem_personas.owl

³http://www.analiticavisual.com/juan/archivos/turisem_dispositivos.owl

de los dispositivos tecnológicos móviles (teléfonos, *tablets* o *PDA's*). Este modelo se ha definido para que el sistema tenga información de los dispositivos de los usuarios, con fines exclusivamente de rendimiento y prestaciones tecnológicas. Los principales conceptos en este modelo son las plataformas de software que incluyen sistemas operativos, lenguajes de programación, las plataformas de *hardware* y los propios dispositivos, clasificados en tres grandes categorías: los teléfonos inteligentes, las tabletas y las *PDA's*.

La Figura 7.1 muestra los dos submodelos más pequeños; por un lado, el modelo de las personas *turisem_personas.owl* y, por otro lado, el modelo de los dispositivos *turisem_dispositivos.owl*, en donde se aprecian los principales conceptos mencionados previamente. Las principales clases de estos modelos son *Persona*, *Dispositivo*, *PlataformaSoftware* y *PlataformaHardware*. Estos dos submodelos son los más pequeños, en conjunto suman un total de veintitres clases, cinco de las cuales son clases enumeradas, nueve propiedades de datos y dieciocho propiedades objeto.

El tercer modelo ontológico (*turisem.owl*⁴) es el más extenso, ya que contiene toda la información de los recursos turísticos como las actividades (deportes, espectáculos, exhibiciones, conciertos, etc.), los lugares (hoteles, bares, restaurantes, parques, galerías, museos, casinos, etc.) y las diferentes rutas para senderismo, entre otra información. Esta ontología está formada por ciento veintiún clases, ochenta y tres propiedades de datos y cuarenta y nueve propiedades objeto. Toda la información contenida en esta base de conocimiento ha sido proporcionada por la empresa Interra, y comprende principalmente lugares dentro de la ciudad de Salamanca, aunque también se incluye alguna otra información de lugares dentro de la provincia como Ciudad Rodrigo, e incluso de algunas otras provincias de Castilla y León como Segovia, Béjar y Ávila.

Entre la información que contiene esta base de conocimiento destacan los principales bares, restaurantes, hoteles, museos y monumentos históricos dentro de la ciudad de Salamanca. También se incluye información sobre parques naturales, bodegas de vino, campos de golf, parques temáticos, centros ecuestres o estaciones de esquí, ubicados la gran mayoría dentro de la provincia de Salamanca y, en menor medida, en otras provincias de Castilla y León. La Figura 7.11 muestra las instancias de la clase *MonumentoHistorico*, que contiene dieciséis valores de los más importantes monumentos históricos dentro de la ciudad de Salamanca. Una descripción más detallada de las instancias en los modelos ontológicos se provee en la sección 7.1.5.

Dentro del planteamiento del dominio se ha contemplado la reutilización

⁴<http://www.analiticavisual.com/juan/archivos/turisem.owl>

de modelos ontológicos relacionados, como se describe en la sección 4.3, de acuerdo al esquema ilustrado en la Figura 4.4. Uno de estos modelos es la ontología *wine.owl*, descrita en la sección 2.1. Este modelo describe un dominio gastronómico y de vinos, que puede reutilizarse para los proyectos *Turimov* y *Ulises*.

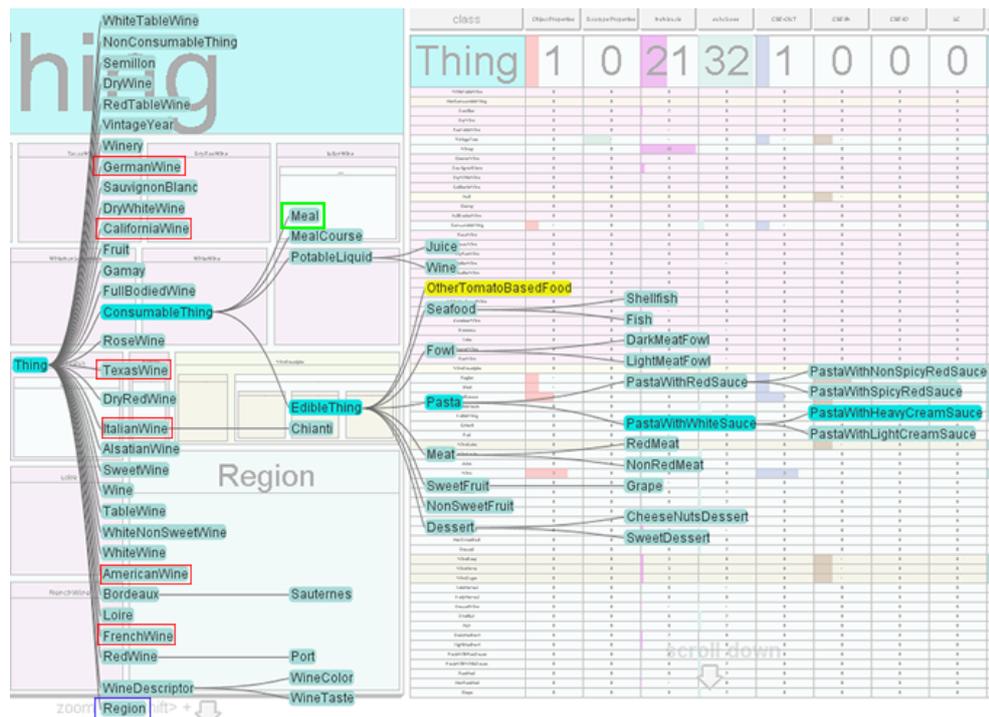


Figura 7.2: La jerarquía completa del modelo ontológico *wine.owl*; se ha extendido la rama con la jerarquía de las clases relacionadas con gastronomía definidas en esta ontología.

La ontología *wine.owl* ha tenido que someterse a un análisis y evaluación basados en el esquema que se ilustra en la Figura 4.4, con la finalidad, primero, de decidir si la ontología puede reutilizarse dentro del dominio que se está modelando y, segundo, para hacer un diagnóstico sobre las modificaciones que se requieren realizar. Este análisis se centra en la búsqueda de conceptos relacionados con vinos españoles y también con gastronomía española. Estos dos aspectos son los requisitos necesarios para que esta ontología pueda reutilizarse en el contexto de los proyectos *Turimov* y *Ulises*.

El primer aspecto a considerar es la diferencia en los idiomas, ya que la ontología *wine.owl* está definida en inglés, mientras que los modelos *turisem.owl*, *turisem_personas.owl* y *turisem_dispositivos.owl* se han diseñado en español.

Esta diferencia idiomática no debería, en principio, representar un impedimento para su reutilización.

El análisis comienza con la búsqueda de los conceptos relacionados con la gastronomía y los vinos españoles⁵. La Figura 7.2 muestra la jerarquía completa de la ontología *wine.owl*, en donde se ha extendido la rama de conceptos relacionados con la gastronomía⁶. Como se puede apreciar en la Figura 7.2, esta ontología no contiene clases específicamente relacionadas con la gastronomía española, sino que, por el contrario, contiene conceptos muy generales y definidos en un nivel de abstracción alto. La clase *Meal* (resaltada en un recuadro en color verde) podría usarse para definir una especialización de alimentos españoles, aunque el concepto más apropiado sería *Cuisine* o *Gastronomy*.

Por lo que respecta a los vinos se han definido diferentes regiones que se han resaltado con un recuadro en color rojo en la Figura 7.2: *GermanWine*, *CaliforniaWine*, *TexasWine*, *ItalianWine*, *AmericanWine* y *FrenchWine*. Esta taxonomía no incluye los vinos españoles, aunque incluye el concepto *Region* (resaltado en un recuadro azul) que permite definir más regiones vinícolas a las ya existentes.

En términos generales, la ontología *wine.owl* incluye conceptos en un nivel de abstracción alto, tal y como se ha descrito en la sección 5.1, que ha definido los flujos de trabajo en base al enfoque de la metodología DOGMA. Debido a que no se han incluido conceptos específicos de vinos y gastronomía española, el trabajo de reutilización de este modelo ontológico abarcaría la definición de estas clases, por lo que el costo sería prácticamente el mismo, si se reutiliza esta ontología que si se definen estas clases en los submodelos *Turisem*.

Una vez analizada la taxonomía del modelo *wine.owl*, la siguiente actividad corresponde al análisis de las propiedades de las clases, para responder preguntas más específicas, tales como: *¿cuáles son las características de los vinos?*, *¿qué tipos de vinos se han incluido en la ontología?*, o las regiones que se han considerado en el modelo ontológico. Estas preguntas pueden contestarse mediante el análisis de las propiedades de las clases, ya que estas definen las características generales, que en este caso corresponden al color, sabor, los niveles de azúcar y el cuerpo del vino. Estas características son comunes a todos los vinos, independientemente de la región de la que provengan, por lo que pueden aplicarse también a los vinos españoles.

Otro resultado del análisis es el hecho de que la gran mayoría de las uvas

⁵Imágenes y vídeos relacionados se encuentran disponibles en:
<http://www.analiticavisual.com/juan/taxonomy.html>

⁶Un vídeo que ilustra este análisis se encuentra disponible en:
<http://www.analiticavisual.com/juan/videos/video1.avi>

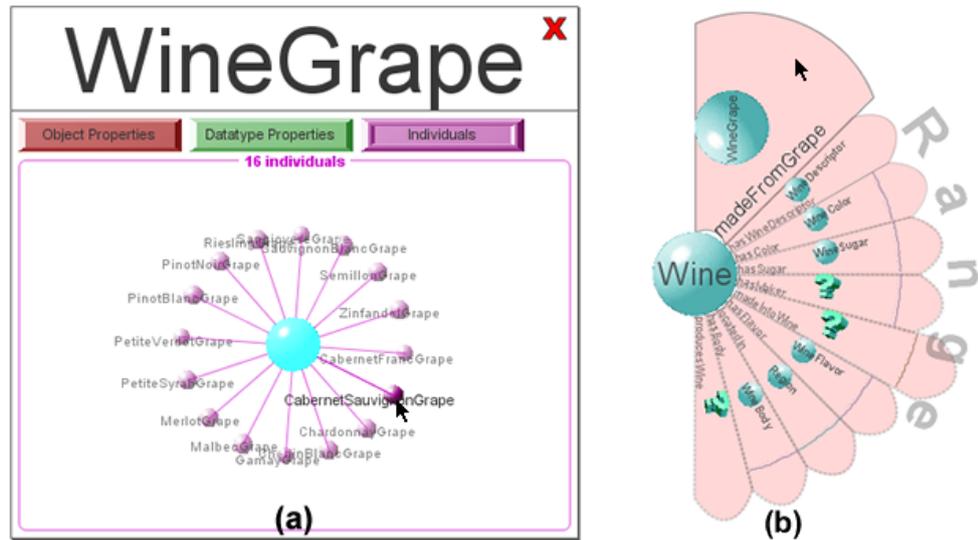


Figura 7.3: (a) muestra las instancias de la clase *WineGrape*, que corresponden a los tipos de uvas. (b) muestra una vista del *zoom semántico* que representa el acoplamiento de la clase *Wine*.

consideradas son de origen francés, como se muestra en la Figura 7.3 (a). Se pueden distinguir las instancias *Cabernet Sauvignon*, *Chardonnay*, *Merlot*, *Semillon*, entre otras. En esta clasificación sería necesario también incluir variedades de uvas autóctonas del territorio español.

En cuanto al análisis del acoplamiento global, la Figura 7.4 muestra una vista global de este acoplamiento. En esta figura se ha resaltado la clase *Wine* para analizar sus relaciones de acoplamiento con el resto de las clases. Para el caso específico de esta clase tan importante se tienen tres relaciones, una es la propiedad *hasColor*, otra es la propiedad *hasWineDescriptor* y, finalmente, la propiedad *madeFromGrape*. La Figura 7.3 (b) muestra el acoplamiento de la misma clase *Wine*, pero desde la perspectiva del *zoom semántico*, y se incluyen propiedades de acoplamiento heredadas de sus superclases. Estas relaciones de acoplamiento se indican con un contorno discontinuo en el polígono que define a la propiedad. Se ha resaltado la propiedad *madeFromGrape* y se distingue en el rango la clase *WineGrape*; esta relación había sido previamente representada en el esquema de la Figura 2.1.

La Figura 7.5 despliega una representación de la propiedad *madeFromGrape* desde otra perspectiva diferente a la de las Figuras 7.3 (b) y 7.4. Sin embargo, se puede apreciar que la propiedad tiene a la clase *Wine* en el dominio y a la clase *WineGrape* en el rango. Estas tres Figuras permiten realizar

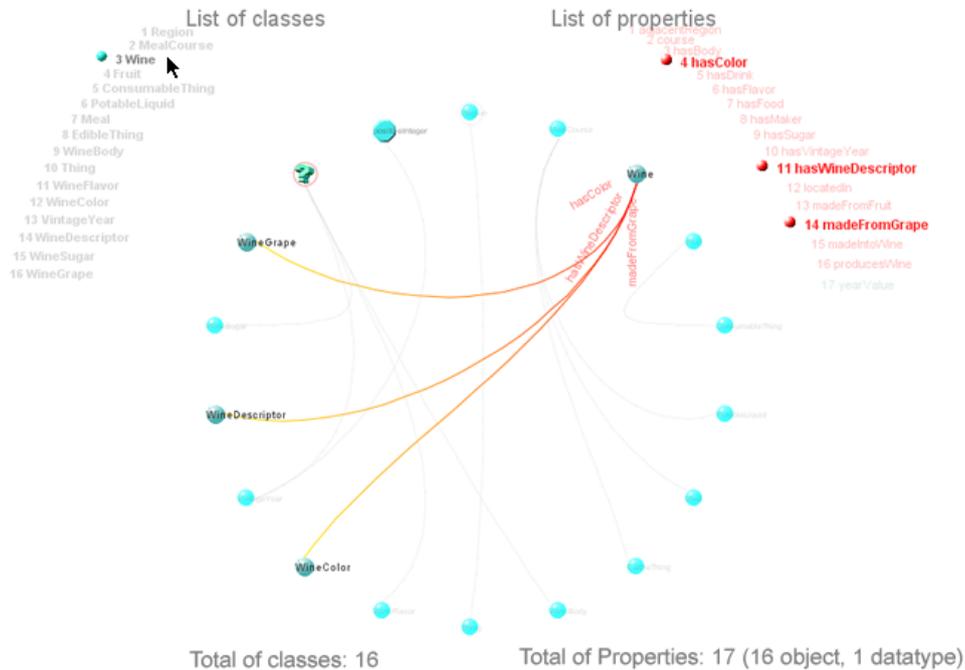


Figura 7.4: Una vista del acoplamiento global de la ontología *wine.owl*, en donde se ha resaltado la clase *Wine* y se muestran sus relaciones. En este caso hay tres relaciones definidas por las propiedades objeto: *hasColor*, *hasWineDescriptor* y *madeFromGrape*.

el análisis del acoplamiento desde tres diferentes perspectivas, en este caso el análisis se ha centrado en dos de las clases más importantes *Wine* y *WineGrape*, y la relación entre estas *madeFromGrape*. Este acoplamiento representa una relación de dependencia entre dos conceptos: el vino y la uva.

En base al análisis desarrollado se puede realizar la toma de decisiones sobre la reutilización de la ontología *wine.owl* en el contexto de los proyectos *Turimov* y *Ulises*. En este sentido, se ha decidido incluir la ontología *wine.owl* en una versión posterior de ambos proyectos, que permita definir muy específicamente aspectos gastronómicos y vinícolas. La primera versión de los modelos se ha definido en un nivel de abstracción alto, sin especialización sobre la gastronomía y los vinos.

Una vez que se ha tomado la decisión acerca de la no reutilización del modelo ontológico *wine.owl* en esta primera versión, se procede con el desarrollo de las ontologías de acuerdo al flujo de trabajo que se muestra en la Figura 7.6, y que se ha descrito previamente. Se comienza con la definición de la taxono-

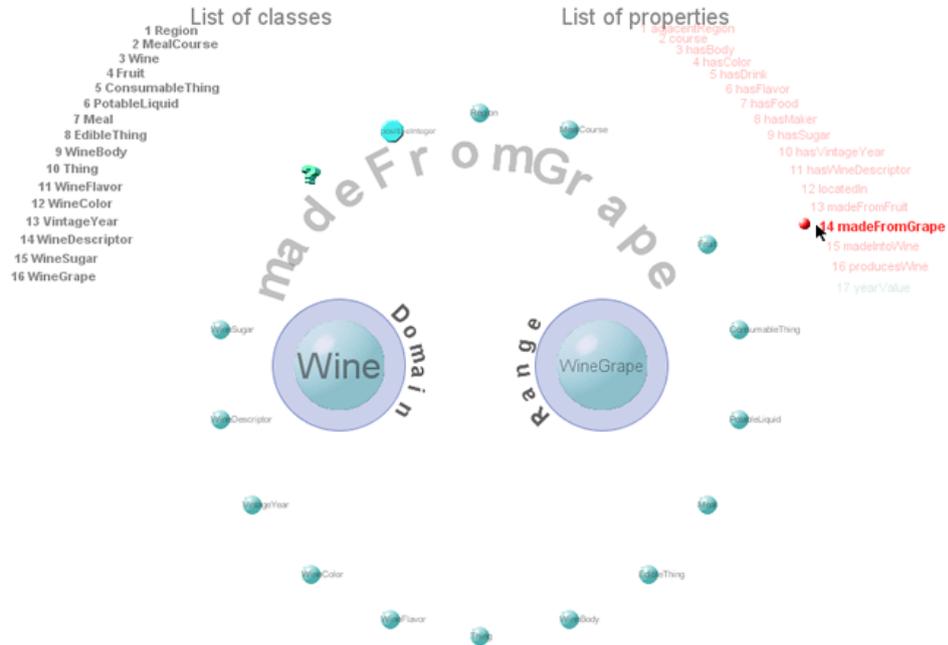


Figura 7.5: Una vista de la representación de la propiedad *madeFromGrape*, con la clase *Wine* en el dominio y la clase *WineGrape* en el rango.

nomía de los conceptos, para lo cual, se utiliza principalmente, la visualización del *treemap*. El segundo paso es la creación de las propiedades intrínsecas, que representan las características de una clase, para lo cual se requiere el uso de una técnica de visualización centrada en mostrar los detalles. La tercera tarea es la definición del acoplamiento entre las clases del modelo ontológico y, finalmente, la cuarta y última actividad corresponde a la creación de las instancias de los modelos ontológicos.

7.1.2. Taxonomía de las ontologías

Una vez realizados los procesos de abstracción, el planteamiento de la solución del problema y el diseño de los submodelos ontológicos, sin incluir reutilización en esta primera versión, entonces dio comienzo el propio desarrollo de las ontologías *turisem.owl*, *turisem_personas.owl* y *turisem_dispositivos.owl*. Este proceso de la creación de las taxonomías se describe con algunos vídeos e imágenes en <http://www.analiticavisual.com/juan/taxonomy.html>.

La implementación de estos submodelos ontológicos, de acuerdo a las activi-

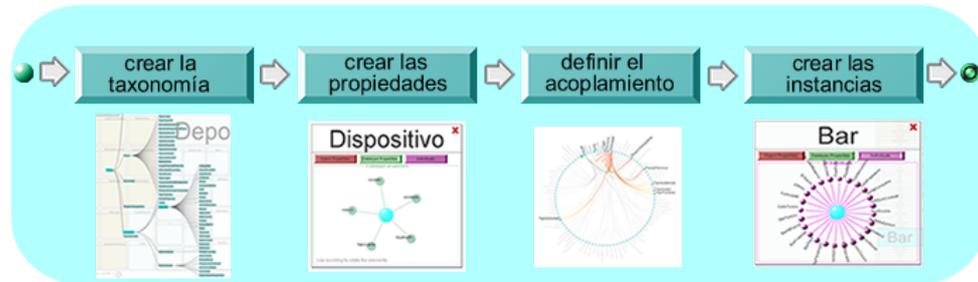


Figura 7.6: El flujo de trabajo de las tareas para el proceso de creación de los modelos ontológicos.

dades mostradas en la Figura 5.2, comienza con la definición de la taxonomía de conceptos de cada uno de ellos. Para el caso del modelo de las personas, las clases más importantes son *Persona* y *OpcionesPersona*, esta última define enumeraciones⁷ que tienen diversas características de las personas, como el sexo o el rango de edad. Algunas actividades muy específicas se pueden definir tomando en cuenta estos dos aspectos, como por ejemplo algunas actividades deportivas en las que se compita en la rama varonil y femenil por separado.

La especialización de los usuarios dentro del sistema comprende básicamente dos categorías; por un lado, la categoría de los turistas en su sentido más común y, por otro lado, la de aquellos usuarios interesados en desarrollar actividades deportivas. Esta especialización se refleja en la jerarquía de subclases de la clase *Persona*, como se muestra en la Figura 7.1. Las relaciones de acoplamiento de la clase *Persona* se ilustran en la Figura 5.17 de la sección 5.3.2.2, que describe las visualizaciones basadas en el uso de la técnica de *zoom semántico*. Esta visualización muestra nueve relaciones de acoplamiento (propiedades objeto) definidas sobre la clase *Persona*, al igual que la Figura 5.23 que muestra la visualización *table lens* con las mismas nueve propiedades objeto.

Para el caso del submodelo de los dispositivos se tienen tres grandes jerarquías de conceptos: la primera se enfoca a los tipos de dispositivos móviles a los cuales se dirige el sistema, la segunda a las plataformas de *hardware* y la tercera a las plataformas de *software*. La especialización de los dispositivos móviles del sistema (mostrada en la Figura 7.1) comprende tres categorías: la primera abarca los teléfonos inteligentes, conocidos como *smartphones*, la segunda las tabletas electrónicas y la tercera las *PDA's*. La especialización de la clase *PlataformaHardware* incluye dos aspectos importantes, por un lado aquellos relacionados con la interfaz del usuario y la resolución del dispositivo.

⁷Clases con instancias con valores constantes como el sexo (masculino y femenino).

Estos dos aspectos juegan un papel crucial en cuanto al uso de los elementos tecnológicos en los dispositivos, ya que estos pueden variar considerablemente entre los diferentes tipos de dispositivos.

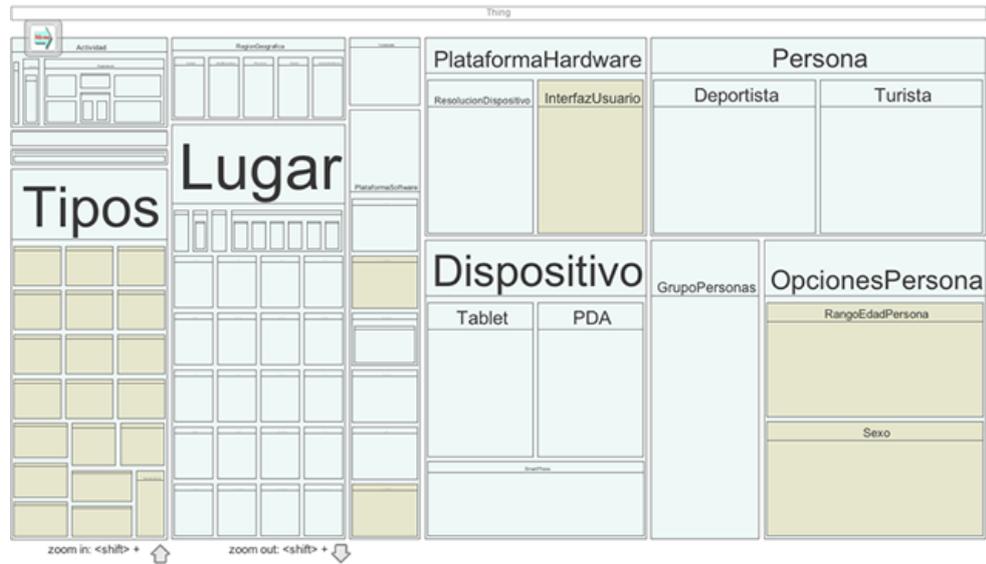


Figura 7.7: La jerarquía completa con los tres submodelos, *turisem.owl*, *turisem_dispositivos.owl* y *turisem_personas.owl*. El modelo ontológico completo contiene un total de ciento cuarenta y tres clases.

Por otro lado, la especialización de la clase *PlataformaSoftware* incluye varias plataformas como los estándares web, el lenguaje de programación *Java*, los protocolos inalámbricos o los sistemas operativos para plataformas móviles. La jerarquía resultante de los submodelos más pequeños *turisem.owl* y *turisem_dispositivos.owl*, se muestra en la Figura 7.1.

El tercero y el más grande de los submodelos ontológicos define los diferentes recursos turísticos, y tiene como sus clases principales a *Lugar*, *Actividad*, *Recorrido* (se refiere a las rutas de senderismo), *RegionGeografica* (que permite definir regiones más específicas como ciudades, provincias, y pueblos, entre otras), y una clase llamada *Tipos*, que sirve como superclase de todas las enumeraciones que definen los valores constantes en el modelo ontológico. Un ejemplo es la enumeración que define los días de la semana, mientras que otro ejemplo es la enumeración que contiene una lista de actividades consideradas como bellas artes.

Por lo que corresponde a las clases del modelo de los recursos turísticos, destacan dos especializaciones: las clases *Lugar* y *Actividad*. Por un lado, la jerarquía de la clase *Lugar* establece una clasificación de los diversos

lugares que se han considerado como recursos turísticos, entre los que se encuentran museos, parques temáticos, parques naturales, bares, restaurantes, hoteles, monumentos históricos, estadios, centros comerciales, plazas de toros, salas de cine, casinos, teatros, zoológicos, discotecas, piscinas, centros ecuestres, autódromos, playas, campos de golf, bibliotecas y galerías de arte, entre otros. Esta clase y su jerarquía de subclases se muestran en el *treemap* de la Figura 7.7. Esta clase *Lugar* es la más acoplada en todo el modelo ontológico, una vista global de este acoplamiento se describe en la Figura 5.20 (b), y una vista detallada de *zoom semántico* de su acoplamiento se muestra en la Figura 5.22.

Por otro lado, la jerarquía de la clase *Actividad* establece una clasificación de las diversas actividades de interés para los usuarios dentro de un ámbito turístico. Así, se incluyen espectáculos (como obras de teatro, corridas de toros, espectáculos deportivos, conciertos, entre otros), también actividades deportivas a realizar por el usuario, visitas a diversos lugares, fiestas patronales o actividades gastronómicas.

La Figura 7.7 muestra la jerarquía completa con los tres submodelos, el modelo de los recursos turísticos *turisem.owl*, el modelo con la información de los dispositivos móviles *turisem_dispositivos.owl* y, por último, el modelo con las características de las personas *turisem_personas.owl*, con un total de ciento cuarenta y tres clases, de las cuales las más importantes se han descrito en los párrafos anteriores. Los colores en el *treemap* permiten distinguir a las clases simples, con el tono gris claro, de las enumeraciones o clases enumeradas, con el tono *beige*.

7.1.3. Propiedades intrínsecas

Una vez creadas las taxonomías de cada uno de los submodelos ontológicos, el siguiente paso es la definición de las propiedades intrínsecas a los conceptos, esto es, las propiedades de datos en las clases. Estas propiedades se definen acorde a la información que se quiere tener de cada uno de los conceptos en el modelo ontológico. Por ejemplo, el modelo de las personas (*turisem_personas.owl*), que contiene a la clase *Persona* como la más importante, define las propiedades *edad* y *nombrePersona* que se muestran en la Figura 5.17 en un tono verde característico de las propiedades de datos. La propiedad *nombrePersona* tiene en el rango al tipo de datos *string*, mientras que *edad* tiene un rango de tipo *integer*. En este caso, el nombre de la persona se utiliza como un identificador, mientras que la edad puede utilizarse para sugerir diversas actividades de acuerdo a este valor.

El submodelo de los dispositivos físicos (*turisem_dispositivos.owl*) contiene

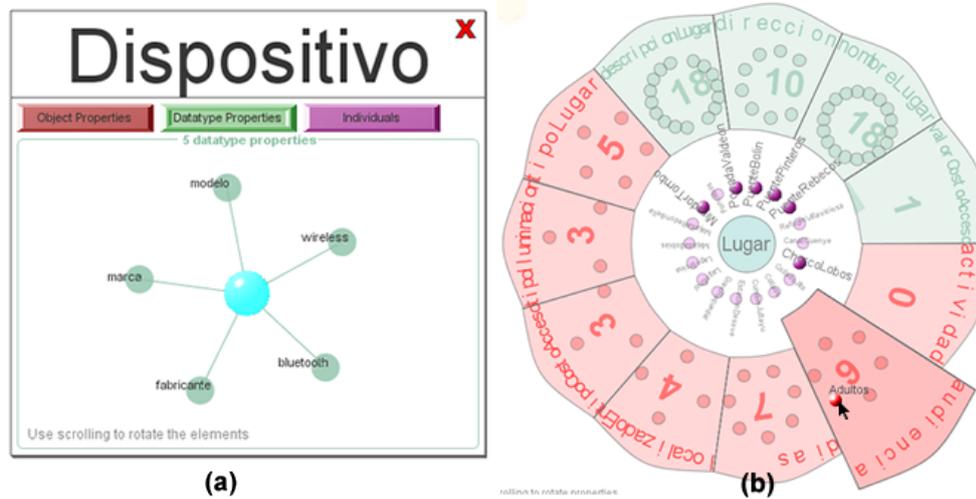


Figura 7.8: (a) muestra el *zoom* semántico sobre la clase *Dispositivo*, con cinco propiedades de datos. La Figura (b) muestra el *zoom* semántico sobre la clase *Lugar*, con siete propiedades objeto y cuatro propiedades de datos.

propiedades de datos relacionadas con los dispositivos, como el nombre del fabricante, la marca del dispositivo, el modelo, el uso de *bluetooth* o si cuenta con una tarjeta inalámbrica (*wireless*), como se muestra en la Figura 7.8 (a), con el *zoom* semántico de la clase *Dispositivo*.

Por otro lado, el submodelo de los recursos turísticos contiene la gran mayoría de propiedades de datos. Entre las clases más importantes de este modelo, se encuentra la clase *Lugar*, que se muestra en la Figura 7.8 (b). Las propiedades de datos de esta clase son *nombreLugar*, *direccion*, *valorCostoAcceso* y *descripcionLugar*; que indican el nombre o identificador del lugar, la dirección, un posible costo de acceso y una descripción. La Figura 7.8 (b), muestra el resultado de una consulta visual sobre todos los lugares enfocados para una audiencia adulta.

Otra de las clases más importantes de este modelo es la clase *Actividad*, que tiene cuatro propiedades de datos: *valorCostoActividad*, *duracionActividad*, *nombreActividad* y *descripcionActividad*. Estas propiedades se utilizan para definir el posible valor del costo para realizar la actividad, la duración estimada de la actividad, el nombre o identificador de la actividad y, finalmente, una breve descripción de la actividad. El acoplamiento de esta clase se muestra en la Figura 7.9 (a).

Dentro de este modelo de recursos, la clase con más propiedades de datos es *Recorrido*, que se muestra en la Figura 7.9 (b) con un total de diez. Esta

clase define las rutas de senderismo sobre los diferentes paisajes de Castilla y León. Dada la naturaleza de esta clase, las propiedades de datos se relacionan con la información específica del recorrido o la ruta, como son una descripción general, el máximo desnivel, la longitud total, el tiempo de recorrido, la mayor altitud, el equipo necesario y un valor en la clasificación del recorrido.

7.1.4. Acoplamiento de los modelos ontológicos

Una vez creadas las propiedades de datos, se crean las propiedades objeto que relacionan diversos conceptos entre sí. Análogamente como se describió con las propiedades de datos, en cada uno de los tres submodelos hay clases más importantes que incluyen propiedades objeto. De tal forma que el modelo de las personas (*turisem_personas.owl*), que contiene a la clase *Persona* como la más importante, define las propiedades que relacionan a esta clase con las demás, principalmente con las clases *Deporte*, *Actividad*, *Espectaculo* y *Lugar*.

La Figura 7.8 (a) muestra el *zoom* semántico de la clase *Dispositivo*. Esta clase tiene cuatro propiedades objeto: *plataformaSoftware*, *plataformaHardware*, *sistemaOperativo* y *resolucionDispositivo*. Estas propiedades tienen como finalidad permitir al modelo ontológico recabar información relevante sobre el dispositivo físico; en este caso, es fundamental conocer el sistema operativo, la resolución de la pantalla, así como aspectos específicos de las plataformas de *software* y de *hardware*.

Finalmente, por lo que respecta al submodelo *turisem.owl* de los recursos turísticos, se destacan las clases *Lugar*, *Actividad* (Figura 7.9 (a)) y *Recorrido* (Figura 7.9 (b)). La clase *Lugar* (Figura 7.8 (b)), tiene entre sus principales propiedades objeto aquellas que relacionan diferente información asociada con los lugares, como la clasificación de los tipos de lugares, los tipos de audiencia, el tipo de iluminación, los días de la semana de disponibilidad, la ubicación, así como también las actividades relacionadas con este lugar, como por ejemplo, los diversos eventos en los escenarios deportivos. Básicamente, esta clase representa un concepto general de un lugar a partir del cual se definen conceptos más específicos, por lo que las propiedades son muy generales.

En lo que respecta a la clase *Actividad*, que se muestra en la Figura 7.9 (a), destacan las propiedades que permiten clasificar los tipos de actividades, los lugares donde se desarrollan las actividades, el costo involucrado o el tipo de público al que va dirigido la actividad. Las clases *Actividad* y *Lugar* están relacionadas por las propiedades *lugarActividad* y *actividad*, que son propiedades inversas entre sí, e indican, por un lado, los lugares en los que se desarrollan las actividades y, por el otro, a cada lugar le asocian un conjunto de actividades

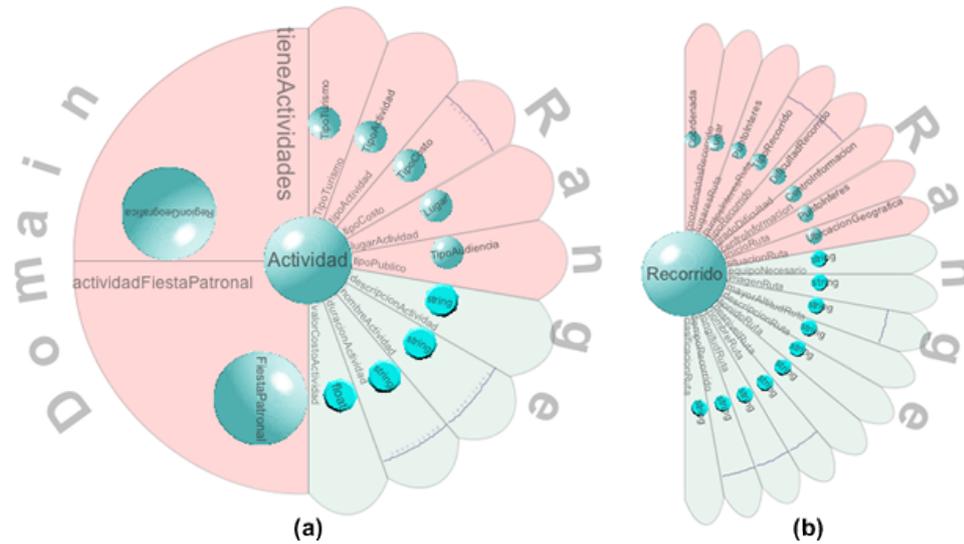


Figura 7.9: (a) *Zoom* semántico sobre la clase *Actividad*, con siete propiedades objeto y cuatro propiedades de datos. La Figura (b) muestra el *zoom* semántico sobre la clase *Recorrido*, con ocho propiedades objeto y diez propiedades de datos.

que se desarrollan en este lugar.

La clase *Recorrido*, que se muestra en la Figura 7.9 (b), define las diversas rutas de senderismo, contiene propiedades de datos que definen información de la ruta, como las coordenadas, los diversos lugares que se localizan durante el recorrido, los puntos de interés, un valor dentro de la clasificación de los tipos de recorridos, un valor en la clasificación de la dificultad de los recorridos y los centros de información ubicados dentro de la ruta.

7.1.5. Instancias de las clases

Una vez definido todo el acoplamiento de las clases en los submodelos ontológicos (las propiedades de datos y las propiedades objeto) es momento de construir las instancias de las clases, que permiten generar la base de conocimiento. Estas instancias son valores específicos para las clases de los modelos ontológicos, principalmente es información referente a lugares turísticos dentro de la provincia de Castilla y León, como restaurantes, bares, hoteles, museos, monumentos históricos, entre otros. Toda esta información ha sido proporcionada por la empresa Interra para este caso de estudio específico.

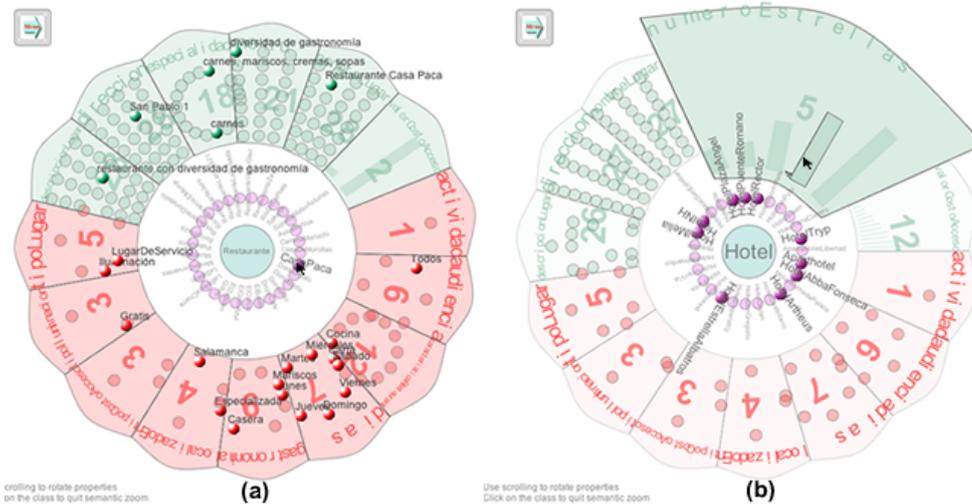


Figura 7.10: (a) *Zoom* semántico sobre la clase *Restaurante*, en donde se muestran los valores para cada una de las propiedades de una de las instancias seleccionada. La Figura (b) muestra el *zoom* semántico sobre la clase *Hotel*, en donde se muestran todas las instancias de hoteles con una clasificación de cuatro estrellas.

La Figura 7.10 (a) muestra el *zoom* semántico sobre la clase *Restaurante*⁸, mientras que la Figura 7.10 (b) muestra el *zoom* semántico sobre la clase *Hotel*⁹. Ambas imágenes muestran el uso de la técnica de consultas visuales descrita en la sección 3.2.3. Por un lado, la Figura 7.10 (a) muestra la clase *Restaurante* que tiene veintinueve instancias, en donde se resaltan todos los valores de una de las instancias que representa a un restaurante ubicado en la ciudad de Salamanca. Entre los valores de las propiedades destacan los días de la semana que abre el restaurante, las especialidades gastronómicas así como la dirección. Por otro lado, la Figura 7.10 (b) muestra la visualización de *zoom* semántico sobre la clase *Hotel*, con veintisiete instancias y la técnica de *visual querying* para la consulta de todos los hoteles de cuatro estrellas.

La Figura 7.11 muestra un fragmento de la *table lens*, en el que se muestran las dieciseis instancias de la clase *MonumentoHistorico*¹⁰. Estas instancias se refieren a diversos monumentos con carácter histórico, que se localizan princi-

⁸Un vídeo con la interacción sobre esta clase, se encuentra disponible en:

<http://www.analiticavisual.com/juan/videos/tesis/queryingRestaurante.avi>

⁹Un vídeo con la interacción sobre esta clase, se encuentra disponible en:

<http://www.analiticavisual.com/juan/videos/tesis/queryingHotel.avi>

¹⁰Un vídeo que ilustra la navegación con la *table lens* se muestra en:

<http://www.analiticavisual.com/juan/videos/tesis/tableLens.avi>

Recurso turístico	Total de instancias
Restaurantes	29
Hoteles	27
Bares	24
Lugares	18
Monumentos históricos	16
Museos	13
Parques	6
Bodegas de vino	5
Campos de golf	4
Parques temáticos	4
Escenarios de paintball	4
Casas rurales	4
Centros ecuestres	2
Estación de esquí	1

Tabla: 7.1: Un resumen del total de instancias de recursos turísticos.

palmente en la ciudad de Salamanca.

La Tabla 7.1 muestra un resumen del total de instancias de recursos turísticos definidos en el submodelo *turisem.owl*. Estas instancias, como se ha comentado previamente, se refieren a recursos turísticos dentro de la comunidad autónoma de Castilla y León, principalmente dentro de la provincia de Salamanca. Estas cantidades pueden no representar el número total de ciertos recursos, por ejemplo hoteles, bares, restaurantes, sino que simplemente se limitan a un cierto número del verdadero total disponible¹¹.

¹¹Imágenes con una mayor resolución así como vídeos se encuentran disponibles en: <http://www.analiticavisual.com/juan/visualQuerying.html>

En la fase de abstracción del dominio se ha decidido, de acuerdo a los requisitos y las especificaciones, el diseño de tres submodelos que permitan, primero, separar la funcionalidad del sistema y, segundo, mejorar los futuros procesos de gestión y una posible reutilización de estos submodelos ontológicos.

Se ha contemplado la posibilidad de reutilizar algunos modelos ontológicos previamente existentes, y en especial un modelo ontológico perteneciente a un dominio gastronómico y vinícola. Se ha desarrollado una fase de análisis, de acuerdo a las actividades propuestas y descritas en la sección 5.1.3, para evaluar la factibilidad en la reutilización de la ontología *wine.owl*. En base al análisis desarrollado, se puede concluir que la ontología *wine.owl* no incluye conceptos gastronómicos ni vinícolas para la región de España, por lo que su reutilización conllevaría la definición de estas clases. Por lo que respecta a la definición del concepto vino, sus características y propiedades se aplican de la misma forma a vinos españoles que a los no españoles, por lo que este aspecto es completamente válido para su posible reutilización. Finalmente, se ha decidido reutilizar este modelo ontológico en una segunda versión de los sistemas *Turimov* y *Ulises*, en los que se incluya una especialización gastronómica y vinícola española, como parte del sistema de recomendación turística.

Cada uno de los tres submodelos se ha creado de acuerdo al flujo de actividades propuesto para el proceso de creación, y con un nivel de abstracción bastante alto, ya que se pretende que los submodelos puedan ser altamente reutilizables. Los conceptos definidos son muy abstractos en general, por el contrario las instancias creadas son muy específicas, especialmente de la región de Salamanca y algunas otras de Castilla y León.

La evaluación de funcionalidad de OWL-VisMod con este caso de estudio, ha resultado exitosa, ya que se ha conseguido el desarrollo de las ontologías para los proyectos *Turimov* y *Ulises* de acuerdo a sus especificaciones. Tanto la evaluación de usabilidad descrita en el capítulo 6, como la evaluación de funcionalidad descrita en este capítulo, permiten concluir que los procesos de creación, gestión, análisis y reutilización definidos, así como la herramienta que soporta estos procesos, permiten el desarrollo de la Ingeniería de Ontologías de una manera eficiente, desde la perspectiva del usuario en el centro de dichos procesos y bajo un enfoque de Analítica Visual.

Capítulo 8

Resultados y conclusiones

Esta tesis se ha desarrollado dentro del ámbito de los sistemas de representación del conocimiento, un área de la Inteligencia Artificial en el que las ontologías juegan un papel fundamental, como uno de los principales mecanismos de representación formal del conocimiento. En dependencia del dominio de aplicación, los sistemas de representación del conocimiento se pueden clasificar en dos grandes vertientes. La primera se refiere a todos aquellos sistemas en los que los procesos se automatizan y los ordenadores adquieren un papel trascendental. Como ejemplos de estos sistemas destacan principalmente la Web Semántica, o sistemas en dominios específicos como la biomedicina, la genética o la biología, en donde la cantidad de información a representar es considerablemente extensa, lo que complica en demasía la intervención humana. En contraparte, para algunos otros dominios de aplicación en los que no hay tal cantidad de información, es el usuario el encargado de desarrollar todos los procesos involucrados en la representación de este conocimiento. Es específicamente en el contexto de este tipo de sistemas en los que la investigación de esta tesis se ha enfocado.

Como planteamiento general de este trabajo de tesis se ha cuestionado la viabilidad para la propuesta de procesos que permitan un desarrollo eficiente de la Ingeniería de Ontologías. Estos procesos consisten básicamente de un flujo de actividades para la creación, gestión, análisis y reutilización de ontologías y bases de conocimiento, bajo el esquema del usuario como punto central en todos estos. De acuerdo al planteamiento del enfoque centrado en el usuario, la propuesta de soluciones se ha basado en el uso de un área de conocimiento denominada Analítica Visual, dadas las similitudes existentes entre estos dos esquemas. Este área de conocimiento tiene como uno de sus principales hitos el uso de mecanismos de visualización e interacción con el usuario, que utilizan

al máximo las capacidades cognitivas de este para crear, gestionar y analizar modelos de conocimiento.

Dada la problemática de las actuales herramientas, descrita principalmente a lo largo del capítulo 4, este trabajo de tesis ha planteado soluciones que mejoran considerablemente las propuestas actuales. Los principales resultados obtenidos durante el desarrollo de esta tesis se describen en los siguientes párrafos.

8.1. Resultados

El primero de los resultados obtenidos es la definición de las tareas y los flujos de trabajo para los principales procesos involucrados en la Ingeniería de Ontologías: creación, gestión, análisis y reutilización de modelos ontológicos. Estos flujos de trabajo se basan en las principales actividades definidas en metodologías existentes, como DOGMA y Methontology, dos propuestas que se han tomado como base en este trabajo de tesis. Dentro de estos flujos de trabajo, destacan principalmente tres grupos de actividades: una primera fase de abstracción del dominio del problema y el planteamiento de soluciones, una fase de análisis de un modelo ontológico, y una fase de implementación del modelo ontológico. La primera fase de abstracción del dominio del problema corresponde su desarrollo completamente al usuario, ya que involucra principalmente a los procesos de comprensión del problema y el diseño de la solución. Las fases de implementación y análisis de los modelos ontológicos incluyen la participación del usuario como el actor principal, pero requieren el soporte de diversos mecanismos y herramientas que permitan su desarrollo de forma eficiente. Es justo en estas dos fases en las que se ha centrado el trabajo de esta tesis.

Un segundo resultado de este trabajo es la implementación de diversos mecanismos de visualización e interacción, propuestos específicamente para resolver la problemática de este dominio de aplicación. El primer objetivo ha sido la corrección de los problemas existentes en las actuales herramientas, tales como la saturación de información, la poca expresividad visual, así como la falta de mecanismos de interacción robustos. Un segundo objetivo ha sido la presentación de nuevas propuestas que permitan al usuario el desarrollo eficiente de los principales procesos involucrados en la Ingeniería de Ontologías.

Una de las principales soluciones propuestas es la definición de los flujos de trabajo en los procesos. Cada una de estas actividades se asocia específicamente a una o más de las técnicas implementadas, y entre las ventajas que esto conlleva se puede mencionar, primeramente, la separación de la funcionalidad

en los procesos, lo que resulta en actividades más específicas y sencillas de realizar con visualizaciones menos saturadas de información. Otra ventaja es que cada una de las técnicas, tanto de visualización como de interacción, se ha definido específicamente para una determinada tarea, lo que resulta en un mecanismo muy intuitivo adecuado a conseguir un objetivo muy específico.

Todo este conjunto de mecanismos, implementados especialmente para cubrir aspectos específicos, define la herramienta denominada OWL-VisMod, que puede describirse de forma general como la implementación de diversas técnicas de visualización e interacción basadas en el enfoque de Analítica Visual, que soportan el desarrollo de los principales procesos de la Ingeniería de Ontologías.

Desde la perspectiva del contexto global de la herramienta, la navegación entre las diversas visualizaciones se desarrolla de acuerdo a los flujos de trabajo propuestos. Esta navegación comienza desde los niveles conceptuales más abstractos y se desarrolla hacia aspectos muy específicos y detallados dentro de los modelos ontológicos. Este flujo de navegación se desarrolla a través de los mecanismos interactivos implementados, mediante los cuales el usuario puede recorrer los diferentes niveles de abstracción de la información representada.

Con la finalidad de obtener retroalimentación de los usuarios y de estimar el rendimiento y funcionalidad de la herramienta, se han desarrollado dos tipos de evaluaciones: una primera evaluación de usabilidad y una segunda evaluación de funcionalidad con un caso de estudio. La primera evaluación de usabilidad se ha llevado a cabo con la participación de un grupo de usuarios, los cuales realizaron un conjunto de actividades para el desarrollo de las fases de análisis y la creación de ontologías, de acuerdo a los flujos de trabajo propuestos. El principal objetivo de esta primera evaluación fue obtener una retroalimentación de los usuarios acerca de los mecanismos de visualización e interacción implementados. Debido a que la herramienta sigue un enfoque centrado en el usuario, esta retroalimentación representa un aspecto crucial.

En términos generales, los usuarios han evaluado la herramienta satisfactoriamente en cuanto a las visualizaciones e interacción. La retroalimentación obtenida ha sido fundamental para corregir y mejorar diversos aspectos. Esta evaluación también permitió la detección de algunos errores de programación, que posteriormente fueron resueltos.

La segunda evaluación consistió en el desarrollo de los modelos ontológicos para un sistema de recomendación para dos proyectos llamados Turimov y Ulises, el primero con un alcance que abarca la comunidad de Castilla y León, mientras que el segundo tiene un alcance mayor. El sistema de recomendación basado en el uso de dispositivos móviles utiliza como modelo de datos tres on-

tologías. Este caso de estudio, abarcó la construcción de estos modelos ontológicos, llamados *turisem.owl*, *turisem_dispositivos.owl* y *turisem_personas.owl*.

El principal objetivo en este caso de estudio ha sido la evaluación de la funcionalidad de la herramienta, esto es, evaluar la capacidad de OWL-visMod para el desarrollo de modelos ontológicos para un ambiente de producción. El diseño y la creación de los submodelos ontológicos se ha desarrollado como parte de este caso de estudio, el cual abarcó también la definición de una base de conocimiento con información específica en el contexto del dominio de turismo para la comunidad de Castilla y León.

El desarrollo de este caso de estudio ha sido satisfactorio, de acuerdo a las especificaciones originales de los proyectos Turimov y Ulises. También ha permitido la detección de algunos aspectos que se mejoraron, así como algunos otros no contemplados durante la fase del diseño original y que fueron posteriormente implementados. La evaluación funcional de la herramienta ha sido exitosa, ya que los modelos ontológicos se han desarrollado de acuerdo a las especificaciones del sistema. El anexo C muestra la carta de conformidad de la empresa Interra, con respecto a los modelos elaborados con la herramienta como caso de estudio.

8.2. Conclusiones

El primer objetivo de esta tesis ha sido la definición de los diversos flujos de trabajo de los procesos involucrados en la Ingeniería de Ontologías. Estos flujos de actividades constituyen la base del diseño de las técnicas de visualización implementadas en OWL-VisMod, que se han tomado a partir de las propuestas más relevantes en el ámbito de las metodologías ontológicas. Estas actividades son básicas y necesarias para los procesos de creación, gestión, análisis y reutilización de los modelos ontológicos.

La creación de ontologías es el proceso mediante el cual un nuevo modelo ontológico se crea, a partir de un cierto dominio de aplicación. Este proceso contiene dos fases generales; por un lado la creación del modelo conceptual, y por otro lado la implementación de este modelo en una ontología o base de conocimiento. La primera de estas fases corresponde a un proceso de abstracción sobre el dominio de aplicación, mientras que la segunda fase requiere específicamente el uso de herramientas que permitan la implementación de este modelo. Es específicamente en este aspecto en el que la propuesta de este trabajo adquiere relevancia, ya que permite **la implementación de los modelos ontológicos de forma intuitiva, en base a un modelado de datos basado en el uso de técnicas de visualización e interacción.**

Otro proceso en el que esta propuesta incide de forma directa y útil es el análisis de ontologías. El análisis de modelos ontológicos o bases de conocimiento es un proceso que resulta complejo en demasía, especialmente para modelos creados por terceras partes. **Las técnicas de visualización permiten una representación visual de los datos, lo que facilita la abstracción y el entendimiento del modelo, en comparación con el uso de técnicas no visuales. La interacción sobre estas técnicas permite al usuario adentrarse sobre el modelo y obtener diversas vistas para mejorar la abstracción y el entendimiento de los datos representados.**

Por lo que respecta a los procesos de gestión y reutilización de ontologías existentes, ambos procesos se basan en los antes mencionados de análisis y creación. Por un lado, la gestión requiere de hacer modificaciones a los modelos existentes, principalmente, la adición de nuevos componentes, lo que involucra un proceso de creación. Por otro lado, la reutilización involucra una primera fase de análisis de los modelos actuales, con el objeto de tomar una decisión acerca de la posible reutilización de dichos modelos existentes.

Como respuesta a la hipótesis planteada, se han definido una serie de flujos de actividades sobre los procesos (creación, gestión, análisis y reutilización), que son soportados por la implementación de la herramienta OWL-VisMod. En conclusión, **este trabajo de tesis ha definido los procesos y flujos de actividades básicos necesarios para el desarrollo de la Ingeniería de Ontologías.**

El segundo objetivo planteado corresponde con el diseño e implementación de la herramienta que soporta los procesos antes mencionados. **Esta implementación primero, ofrece mejores prestaciones con respecto a las propuestas actuales, en cuanto a visualización e interacción; y segundo, se ha validado en cuanto a su capacidad para desarrollar los procesos de creación, gestión, análisis y reutilización.**

Las estrategias en el diseño de la herramienta OWL-VisMod han sido por un lado, la separación de la funcionalidad entre las diversas técnicas implementadas; y, por otro lado, el diseño de ciertas técnicas específicas para el dominio de la Ingeniería de Ontologías. Estas estrategias han permitido establecer una secuencia claramente definida en cuanto a los flujos de trabajo y sus actividades, y en consecuencia, se evita la saturación de información al usuario, como en el caso de otras herramientas.

Otras estrategias abarcan la implementación de técnicas de visualización robustas, especialmente enfocadas para la representación de grandes cantidades de información; también la inclusión de técnicas muy específicas para analizar los detalles de los elementos de datos, así como funcionalidad no solo para mo-

delar ontologías, sino también bases de conocimiento a través de la creación de instancias de los conceptos. En conclusión, el diseño de la herramienta define claramente la secuencia de actividades en los procesos, a través de la navegación de las técnicas visuales. El resultado es que los procesos se realizan de forma más clara e intuitiva por el usuario, lo que facilita el desarrollo de estas actividades. Por otro lado, la robustez de las visualizaciones mejora significativamente los procesos de modelado de ontologías y bases de conocimiento, con respecto a otras propuestas.

Aparte de las técnicas de visualización, otro factor crucial es la interacción del usuario con la herramienta. En el diseño de OWL-VisMod se ha considerado una interacción robusta, con un conjunto de funcionalidades que no se limitan simplemente a visualizar la información, sino que permiten al usuario entre otras cosas, tener múltiples vistas y perspectivas de los modelos ontológicos, así como navegar a través de los diferentes niveles de abstracción. A diferencia de otras propuestas que implementan técnicas de interacción muy pobres, OWL-VisMod basa completamente su diseño tanto en una visualización robusta, como en una interacción potente con el usuario, que permita a este abstraer conocimiento y modelar nuevo conocimiento sobre los modelos ontológicos de forma muy intuitiva.

En conclusión, se puede afirmar que todas las técnicas visuales y de interacción propuestas, se han diseñado completamente enfocadas en las necesidades de las actividades en los flujos de trabajo de los procesos. De forma más detallada, **las visualizaciones más generales se han utilizado con éxito en otros dominios de aplicación con características similares, mientras que las visualizaciones más detalladas se han definido específicamente para resolver la problemática en este dominio de aplicación en concreto.**

Otro de los principales objetivos en este trabajo de tesis es la aplicación de la Analítica Visual en el área de Ingeniería de Ontologías, un campo en el que no se había utilizado previamente. El desarrollo de OWL-VisMod demuestra que el enfoque de la Analítica Visual puede aplicarse con éxito en los procesos involucrados en el ciclo de vida de las ontologías. El uso de técnicas de visualización e interacción basadas en el enfoque del usuario como parte central en los procesos antes mencionados, le permiten a este, desarrollar las actividades de forma más simple y eficiente, lo que resulta en modelos ontológicos de mejor calidad y con menos esfuerzo por parte del usuario. El soporte a las actividades y a los procesos mediante herramientas robustas que explotan al máximo las capacidades cognitivas del usuario, simplifican en gran medida, el análisis y modelado de las ontologías y las bases de conocimiento.

Este enfoque se basa en un esquema en el que el usuario utiliza estos mecanismos para modelar y crear nuevo conocimiento, u obtener conocimiento de modelos previamente creados. Para lograr este objetivo, el área de investigación de la Analítica Visual se fundamenta principalmente en el uso de la capacidad cognitiva humana. Todas las técnicas, tanto de visualización como de interacción que se basan en este paradigma, están diseñadas en su totalidad para utilizar de la forma más eficientemente posible, esta capacidad cognitiva. Principalmente se basan en el uso interfaces visuales dotadas de una gran capacidad de interacción persona-ordenador, que sacan el máximo provecho de la capacidad de procesamiento de los ordenadores actuales, para permitir al usuario desarrollar un proceso de abstracción de un cierto dominio de aplicación.

OWL-VisMod aprovecha al máximo la capacidad cognitiva humana mediante la interacción visual de modelos ontológicos sobre algún dominio de aplicación. Esta capacidad se ha venido aplicando en diversas áreas como la Ingeniería de *Software*, sin embargo, no se había aplicado previamente al campo de la Ingeniería de Ontologías. Este trabajo de tesis ha demostrado que estas técnicas pueden aplicarse con éxito en el área de la Ingeniería de Ontologías, lo que enriquece considerablemente los procesos de análisis y modelado de ontologías y bases de conocimiento.

La investigación realizada ha permitido, primero, detectar la problemática principal de las actuales herramientas y, segundo, en base a los problemas detectados, el planteamientos de soluciones muy específicas al dominio de la Ingeniería de Ontologías. OWL-VisMod consiste de un conjunto de técnicas muy especializadas e intuitivas, que utilizan la capacidad de procesamiento de los ordenadores y permiten al usuario utilizar su capacidad cognitiva visual para el desarrollo de las actividades dentro de los procesos básicos de la Ingeniería de Ontologías.

Como conclusión general, se tiene que OWL-VisMod es el resultado de una serie de propuestas de diversas técnicas de visualización e interacción, que soportan los principales procesos de la Ingeniería de Ontologías bajo un esquema de Analítica Visual centrado en el usuario, quien se encarga de modelar y analizar las bases de conocimiento.

8.3. Publicaciones relacionadas con la tesis

Entre los resultados obtenidos se pueden listar diversas publicaciones en congresos internacionales y revistas, que soportan el trabajo de investigación

de esta tesis al someterlo al escrutinio de diversos investigadores en el área. Esta lista se detalla a continuación:

Revistas JCR

1. García, J., García-Peñalvo, F. J., Therón, R., Ordóñez de Pablo, P. "Usability Evaluation of a Visual Modelling Tool for OWL Ontologies". *Journal of Universal Computer Science*. 17(9): 1299-1313. May 2011. ISSN: 0948-695X. Available online at http://www.jucs.org/jucs_17_9. [91].
2. García-Peñalvo, F. J., García, J., Therón, R. "Analysis of the OWL ontologies: A survey". *Scientific Research and Essays*. 6(20): 4318-4329. Academic Journals. 19 September, 2011. ISSN 1992-2248. Available online at: <http://www.academicjournals.org/sre/contents/2011Cont/19Sept.htm>. [99].

Revistas

1. García, J., Gómez Aguilar, D. A., González, A. García Peñalvo, F. J., Therón, R. "A middleware framework to create data structures for a visual analytics object oriented approach". *International Journal of Knowledge and Learning (IJKL)*, 6(2/3): 256-267. Inderscience. 2010. ISSN (Online): 1741-1017 - ISSN (Print): 1741-1009. DOI: 10.1504/IJKL.2010.034757 [83].

Capítulos de libro y Postproceedings

1. García, J., Therón, R., García, F. J. "Visualization of Large Software Projects by using Advanced Techniques". In T. Sobh (Ed.), *Innovations and Advances in Computer Sciences and Engineering*. Pages 325-330. Springer: Netherland. ISBN 978-90-481-3657-5 (Print) 978-90-481-3658-2 (Online). DOI 10.1007/978-90-481-3658-2. 2010 [94].
2. García, J., García, F. J., Therón, R. "Visualising Semantic Coupling among Entities in an OWL Ontology". In M. Á., Sicilia, C. Kop, F. Sartori (Eds.), *Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Science*. 4th International Workshop, ONTOSE 2010, held at CAiSE 2010, Hammamet, Tunisia,

June 7-8, 2010, Revised Selected Papers. Pages 90-106. Series: Lecture Notes in Business Information Processing, Vol. LNBIP 62. Berlin Heidelberg: Springer Verlag. ISSN 1865-1348. ISBN-10 3-642-16495-1 ISBN-13 978-3-642-16495-8. DOI: 10.1007/978-3-642-16496-5_7. 2010 [85].

Contribuciones en Congresos

1. García, J., Therón, R., García F. J. “Visualization of Large Ontologies by Using Advanced Techniques”. In Web Mining and Semantic Web. I Workshop, MiWebSe 2008 (27 October - 9 November, 2008, Online Conference). Ma. N. Moreno García, A. B. Gil González, V. F. López Batista (Eds.). Pages 1-10. Salamanca: Departamento de Informática y Automática de la Universidad de Salamanca. ISBN 978-84-691-5945-3. 2008 [94].
2. García, J., Therón, R., García, F. J. “Visualization of Large Software Projects by using Advanced Techniques”. In Proceedings of the Second International Conference on Systems, Computing Sciences & Software Engineering (SCSS 08). Held in the framework of the International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering (CISSE 2008) (5-12 December 2008). Online Conference. DVD Version. 2008 [95].
3. García, J., González Torres, A., Gómez Aguilar, D., Therón, R., García Peñalvo, F. J. “A Visual Analytics Tool for Software Project Structure and Relationships among Classes”. In 10th International Symposium, SG 2009, Proceedings (Salamanca, Spain, May 28-30, 2009). Published in A. Butz, B. Fisher, M. Christie, A. Krüger, P. Olivier, R. Therón (Eds.) Smart Graphics, Series: Lecture Notes in Computer Science. VOL. LNCS 5531. Berlin Heidelberg: Springer Verlag. Pages 203-212. ISSN 0302-9743. ISBN-103-642-02114-X. ISBN-13 978-3-642-02114-5. 2009 [92].
4. García, J. Gómez, D. A., González, A., García, F. J., Therón, R. “A Middleware Framework to Create and Manage Data Structures for Visual Analytics”. In Best Practices for the Knowledge Society. Knowledge, Learning, Development and Technology for All. Second World Summit on the Knowledge Society, WSKS 2009, Chania, Crete, Greece, September 16-18, 2009. Proceedings (Chania, Crete, Greece, September 16-18, 2009). M. D. Lytras, P. Ordoñez de Pablos, E. Damiani, D. Avison, A. Naeve, D. G. Horner (Eds.). Series: Communications in Computer and Information Science. Berlin, Heidelberg: Springer. Vol. CCIS 49. Pages

- 466-473. ISSN 1865-0929 (Print) 1865-0937 (Online). ISBN-10 3-642-04756-4. ISBN-13 978-3-642-04756-5 (Print). ISBN-13 978-3-642-04757-2 (Online). DOI 10.1007/978-3-642-04757-2_50. 2009 [100].
5. García, J., García, F. J., Therón, R. "Defining Coupling Metrics among Classes in an OWL Ontology". In Trends in Applied Intelligent Systems. 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2010, Cordoba, Spain, June 1-4, 2010, Proceedings, Part II (Córdoba, Spain, June 1-4, 2010). N. García-Pedrajas, F. Herrera, C. Fyfe, J. Manuel Benítez, M. Ali (Eds.): Series: Lecture Notes in Artificial Intelligence. Subseries: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer. Vol. LNAI 6097. Pages 12-17. ISSN 0302-9743 (Print) 1611-3349 (Online). ISBN-13: 978-3-642-13024-3. ISBN-10: 3-642-13024-0. 2010 [88].
 6. García, J., García Peñalvo, F. J., Therón, R. "Visualising Semantic Coupling Among Entities in an OWL Ontology". In ONTOSE 2010 - Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Sciences, held with CAiSE 2010, (Hammamet, Tunisia), June 7-8, 2010 [85].
 7. García, J., García, F. J., Therón, R. "Analizando el acoplamiento entre las clases de una ontología OWL". En las Actas de las XV Jornadas de Ingeniería del Software y Bases de Datos - JISBD 2010, celebradas en el marco del III Congreso Español de Informática - CEDI 2010. (Valencia, 7-10 de septiembre de 2010). E. Teniente, S. Abrahao (Eds.). Madrid: Ibergarceta Publicaciones S.L., Páginas 207-216. ISBN 978-84-92812-51-6. 2010 [87].
 8. García, J., García Peñalvo, F. J., Therón, R. "A Survey on Ontology Metrics". In Knowledge Management, Information Systems, E-Learning, and Sustainability Research. Third World Summit on the Knowledge Society, WSKS 2010, Corfu, Greece, September 22-24, 2010 Proceedings, Part I (Corfu, Greece, September 22-24, 2010). M. D. Lytras, P. Ordoñez De Pablos, A. Ziderman, A. Roulstone, H. Maurer, J. B. Imber (Eds.): Series: Communications in Computer and Information Science. Berlin, Heidelberg: Springer. Vol. CCIS 111. Pages 22-27. ISSN 1865-0929. ISBN-13: 978-3-642-16317-3. ISBN-10: 3-642-16317-3. DOI: 10.1007/978-3-642-16318-0_4. 2010 [93].
 9. García, J., Therón, R., García, F. J. "Semantic Zoom: A Details on Demand Visualisation Technique for Modelling OWL Ontologies". In Highlights in Practical Applications of Agents and Multiagent Systems, 9th

International Conference on Practical Applications of Agents and Multiagent Systems - PAAMS 2011 (Salamanca, Spain, April 6-8, 2011). J. Bajo Pérez, J. M. Corchado, Ma. N. Moreno, V. Julián, P. Mathieu, J. Canada-Bago, A. Ortega, A. Fernández Caballero (Eds.): Series: Advances in Intelligent and Soft Computing. Berlin, Heidelberg: Springer. Vol. AISC 89. Pages 85-92. ISSN 1867-5662. ISBN 978-3-642-19916-5. e-ISBN 978-3-642-19917-2. DOI 10.1007/978-3-642-19917-2. 2011 [96].

10. García, J., García, F. J., Therón, R. "Modelling Relationships among Classes as Semantic Coupling in OWL Ontologies". In Proceedings of the 2011 International Conference on Information & Knowledge Engineering, IKE 2011, held at The 2011 World Congress in Computer Science, Computer Engineering, and Applied Computing - WORLDCOMP'11 (July 18-21, 2011, Las Vegas, Nevada, USA). H. R. Arabnia, R. R. Hashemi (Eds.). Vol. 1, pages 22-28. CSREA Press. ISBN Press: 1-60132-188-0. CD-ROM ISBN 1-60132-169-4. 2011 [86].
11. García, J., García, F. J., Therón, R. "Evaluating the usability of OWL-VisMod: A modelling tool for OWL ontologies". En las Actas del XII Congreso Internacional de Interacción Persona-Ordenador - Interacción 2011. (Lisboa, Portugal, 2-6 de septiembre de 2011). N. Garay, J. Abascal (Eds.). Madrid: Ibergarceta Publicaciones S.L., Páginas 297-300. ISBN 978-84-9281-234-9. 2011 [89].
12. García, J., García-Peñalvo, F. J., Therón, R.. "Through the Data Modelling Process of Turimov, an Ontology-based Project for Mobile Intelligent Systems". Proceedings of 10th International Conference on Practical Applications of Agents and Multi-Agent Systems, 2012 [90]. EN PRENSA

8.4. Líneas de trabajo futuro

Este trabajo de tesis se ha basado en los principales y más básicos procesos involucrados en la Ingeniería de Ontologías, sin embargo, existen algunos otros procesos que han quedado fuera de esta investigación por sobrepasar los objetivos establecidos. Las líneas de investigación futuras se relacionan principalmente con estos diversos procesos que no han sido incluidos en este trabajo.

Entre estas líneas, se puede mencionar primeramente, un aspecto que ha sido discutido en varias metodologías como Methontology [71], y que se refiere al proceso de documentación de ontologías. A pesar de que se ha mencionado

este proceso en algunas metodologías, la realidad es que hay muy poco trabajo relacionado con la documentación de los modelos ontológicos. La falta de mecanismos de documentación de ontologías complica en demasía el proceso de reutilización de estos modelos ontológicos existentes, por lo que pueden resultar muy benéficas las nuevas propuestas en esta dirección.

Otra línea de investigación se refiere a la alineación de ontologías, un proceso utilizado principalmente para la reutilización de ontologías y la integración de bases de conocimiento heterogéneas. La alineación básicamente consiste en determinar las correspondencias entre los conceptos de dos o más ontologías. Sobre todo, la futura línea de investigación se basa en el proceso de alineación de ontologías mediante técnicas y mecanismos de visualización, siguiendo el mismo enfoque de la Analítica Visual.

Una tercera línea de investigación se refiere a dos conceptos similares, por un lado al mapeo y por otro lado la mezcla de ontologías. El mapeo de ontologías es un mecanismo mediante el cual múltiples ontologías se utilizan desde diversas aplicaciones. Esta interacción permite el intercambio de información semántica, especialmente en ambientes distribuidos como la Web Semántica. En el mapeo de ontologías, las correspondencias entre dos ontologías se almacenan de forma separada de las propias ontologías, por lo que no forman parte de las propias ontologías. Por el contrario, en la mezcla de ontologías se crea una nueva ontología como la unión de las ontologías fuente. La ontología resultante captura todo el conocimiento de las ontologías originales.

Se pretende que las líneas de investigación descritas (la documentación, la alineación, el mapeo y la mezcla de ontologías) sean enfocadas desde la perspectiva de la Analítica Visual, con la propuesta e implementación de diversas técnicas de visualización e interacción con el usuario, de la misma forma en que se han propuesto las soluciones en este trabajo de tesis.

Los objetivos serían similares a los descritos en el presente trabajo y conllevarían primero, una revisión exhaustiva del estado del arte para la identificación de la problemática existente. Como segundo aspecto, la propuesta de soluciones basadas en el enfoque de la Analítica Visual, y la implementación de estas soluciones en la herramienta. También es necesaria la evaluación de las propuestas para su validación, especialmente con usuarios que provean una retroalimentación de diversos aspectos que puedan mejorarse, o, incluso, algunos otras sugerencias nuevas que puedan implementarse y que no hubieran sido consideradas en el diseño original.

Apéndice A

Manual de usuario

1. Crear una nueva ontología
2. Abrir una ontología existente
3. Guardar una ontología
4. Configurar una conexión a base de datos
5. Agregar submodelos al modelo actual
6. Protocolo FTP (*File Transfer Protocol*) para transferencia de archivos
7. Menú visual
8. Desplegar el código XML
9. Desplegar la ayuda
10. Uso del difuminado del fondo
11. Búsqueda de clases
12. Desplegar estadísticas
13. Reiniciar las visualizaciones
14. Cambiar de visualización
15. Modificar el esquema de la ontología
16. Crear una clase como unión o intersección
17. Navegación en los niveles del *treemap*

18. Crear clases enumeradas
19. Eliminar clases, propiedades o instancias
20. Opciones de interacción de la *table lens*
21. Opciones de interacción de la vista de acoplamiento
22. Interacción con las listas circulares

A.1. Crear una nueva ontología

Para crear una nueva ontología presionar la opción *File -> Create Ontology*. Esta opción, que se muestra con un rectángulo resaltado en color rojo en la Figura A.1 (a), crea un nuevo modelo ontológico vacío.

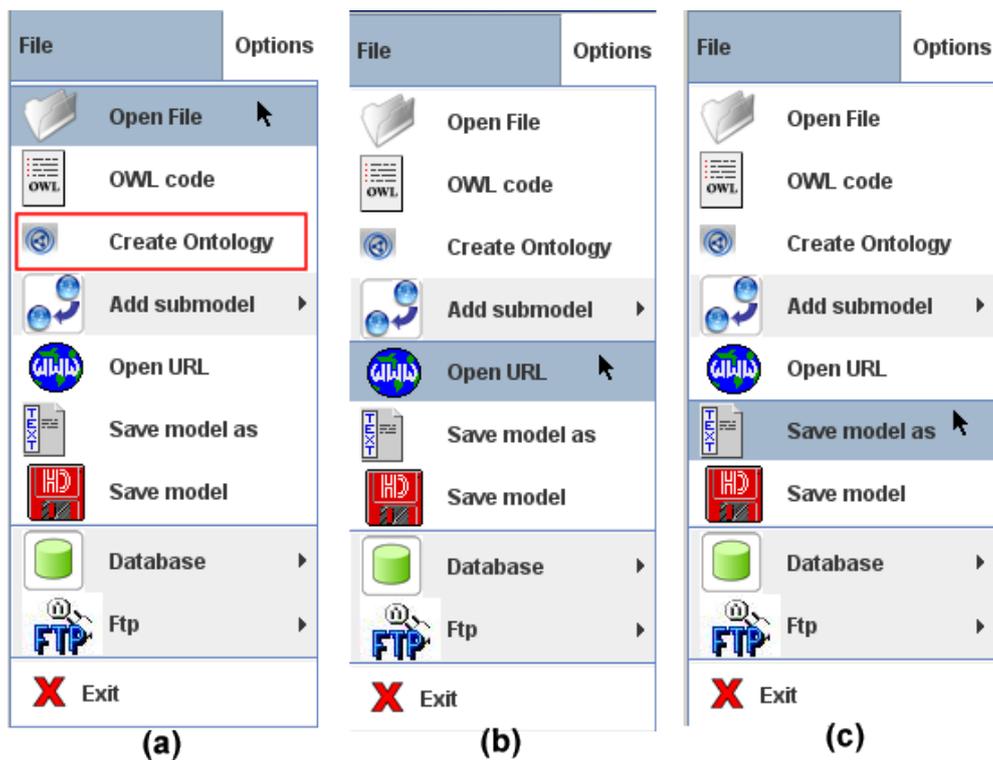


Figura A.1: Las opciones del menú *File* que contienen funciones para (a) abrir un archivo, (b) abrir una dirección URL y (c) guardar un modelo en un archivo con extensión “.owl”, y también para crear una nueva ontología.

A.2. Abrir una ontología existente

Para abrir una ontología existente hay tres opciones, dependiendo de la ubicación del modelo ontológico. La primera opción es abrir directamente un archivo con extensión “.owl” localizado en el propio ordenador del usuario, con la opción *File -> Open File*. Esta opción se muestra en la Figura A.1 (a).

La segunda opción es abrir directamente una dirección URL con el modelo ontológico, con la opción *File -> Open URL*. Esta opción se muestra en la Figura A.1 (b), y obviamente debe existir una conexión a internet en el ordenador del usuario.

La tercera opción es abrir una ontología desde una base de datos, con la opción *File -> Database -> Open model from DB*. Esta opción se ilustra en un rectángulo de color rojo en la Figura A.2, para lo que se requiere haber configurado previamente una conexión a la base de datos. Para más información sobre la configuración de una conexión a la base de datos, consultar la sección A.4.

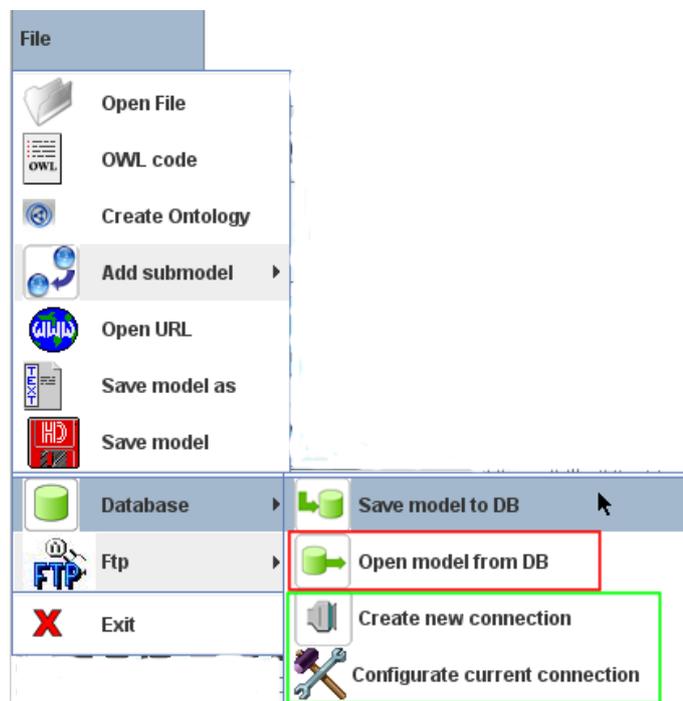


Figura A.2: Las opciones del menú *File -> Database* que contiene las diferentes opciones para la funcionalidad de la base de datos.

A.3. Guardar una ontología

Para guardar una ontología hay dos posibles opciones: la primera es guardarla en un archivo en formato XML con extensión “.owl”, y la segunda opción es guardarla en una base de datos. La opción para guardar en un archivo es *File ->Save model as*, y se ilustra en la Figura A.1 (c). En esta opción hay que especificar la ruta en la que se quiere almacenar el archivo con la ontología, así como el nombre del archivo con extensión “.owl”.

La opción para guardar el modelo ontológico en una base de datos es *File ->Database ->Save model to DB*. Esta opción se ilustra en la Figura A.2, para lo que se requiere haber configurado previamente una conexión a la base de datos. Para más información sobre la configuración de una conexión a la base de datos, consultar la sección A.4.

A.4. Configurar una conexión a base de datos

Las conexiones a las bases de datos se realizan mediante el uso de la API JDBC (*Java Database Connectivity*). La configuración de una base de datos se almacena en un archivo denominado “properties.xml”, el cual requiere la siguiente información:

- El nombre de la base de datos.
- La contraseña para conexión con la base de datos.
- La dirección url del ordenador huésped de la base de datos.
- El nombre del motor de base de datos.
- El nombre de usuario para establecer la conexión con la base de datos.
- El puerto para la conexión con la base de datos.

OWL-VisMod permite crear diferentes conexiones para guardar los modelos ontológicos, como se muestra en el recuadro en color verde de la Figura A.2, con las opciones *File ->Database ->Create new connection* y *File ->Database ->Configure current connection*. La estructura de un archivo de configuración es la siguiente:

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

```
<!DOCTYPE properties SYSTEM "http://java.sun.com/dtd/properties.dtd">
<properties>
<comment>the properties file for the database connection</comment>
<entry key='database_name'>ontoDB</entry>
<entry key='database_password'>usuarioMySQL</entry>
<entry key='database_host'>localhost</entry>
<entry key='database_engine'>MySQL</entry>
<entry key='database_login'>user</entry>
<entry key='database_port'>3306</entry>
</properties>
```

A.5. Agregar submodelos al modelo actual

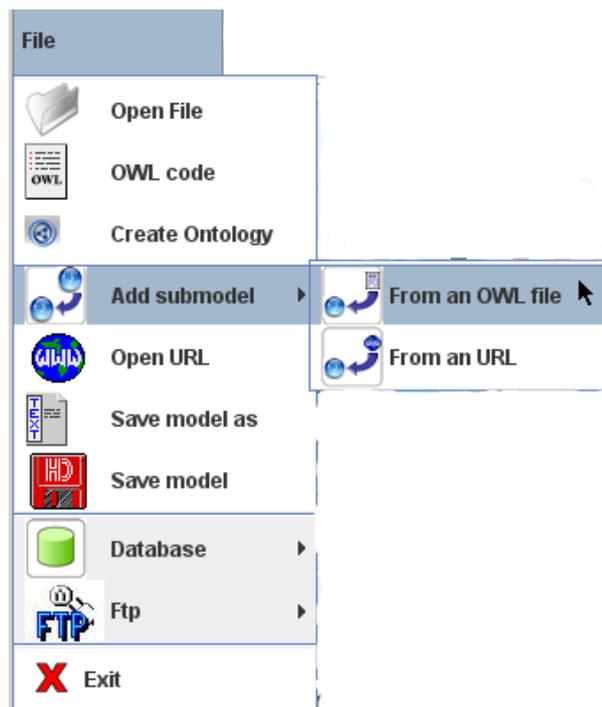


Figura A.3: Las opciones del menú *File ->Add submodel* para agregar un submodelo al actual modelo ontológico.

Una funcionalidad de OWL-VisMod es la posibilidad de agregar submodelos al actual modelo ontológico, esta funcionalidad tiene dos opciones: agregar un submodelo desde un archivo con extensión “.owl”, o desde una dirección URL. La primera opción *File ->Add submodel ->From an OWL file* agrega

un submodelo ontológico desde un archivo con extensión “.owl” ubicado en el ordenador del usuario.

La segunda opción *File ->Add submodel ->From an URL* agrega un submodelo ontológico desde una dirección URL, para lo cual se requiere que el ordenador tenga una conexión a internet. Ambas opciones se ilustran en la Figura A.3.

A.6. Protocolo FTP (*File Transfer Protocol*) para transferencia de archivos

OWL-VisMod ha implementado el protocolo para transferencia de archivos FTP, como se muestra en la Figura A.4. La funcionalidad incluye las opciones de enviar y recibir archivos a través de Internet mediante este protocolo. Esta funcionalidad se ha implementado para la publicación de los archivos con las ontologías en los diversos servidores, con las opciones *File ->Ftp ->Send files via FTP* y *File ->Ftp ->Retrieve files via FTP*.

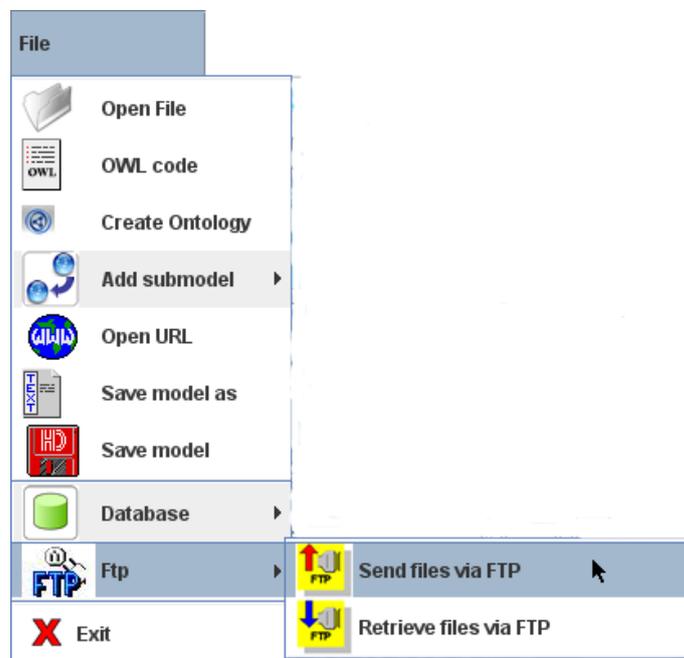


Figura A.4: Las opciones del menú la funcionalidad del protocolo para transferencia de archivos FTP.

A.7. Menú visual

El menú visual se muestra en la Figura A.5, y contiene las siguientes funciones de izquierda a derecha: crear una subclase de la clase enfocada o seleccionada, modificar la clase enfocada o seleccionada, eliminar la clase enfocada o seleccionada, crear una propiedad objeto, crear una datatype property, crear una instancia, crear una enumeración y finalmente, crear una clase como unión o intersección de otras clases. Este menú representa una opción más rápida para esta funcionalidad en las visualizaciones.



Figura A.5: Menú visual que contiene las principales funciones, como crear, modificar o eliminar clases, propiedades, instancias, enumeraciones o clases definidas como unión o intersección.

A.8. Desplegar el código XML

Una funcionalidad de OWL-VisMod, es la posibilidad de desplegar el código XML generado por la ontología OWL. Esta opción *File -> OWL code* se muestra en la Figura A.6 (a), y la Figura A.6 (b) muestra la ventana que contiene el código OWL.

A.9. Desplegar la ayuda

Un aspecto fundamental para los usuarios es desplegar la ayuda sobre las visualizaciones, con la opción *Help -> Help* que se muestra en la Figura A.7 (a).

La opción *Help -> About* que se muestra en la Figura A.7 (b), despliega información sobre la versión de la herramienta.

A.10. Uso del difuminado del fondo

La técnica de difuminar el fondo de las visualizaciones se utiliza para mantener el contexto, especialmente cuando se utiliza una vista en detalle como

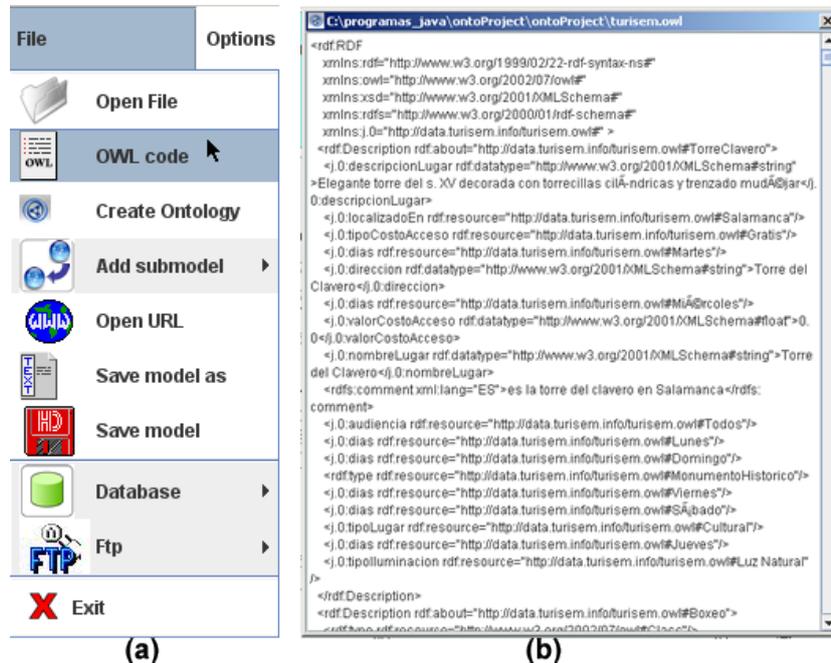


Figura A.6: El código OWL de la ontología *turisem.owl* se muestra con la opción *File -> OWL code*.

el Foco+Contexto, o el *zoom* semántico. Sin embargo, esta técnica consume recursos de cómputo como memoria y procesamiento, y puede ralentizar las visualizaciones, especialmente cuando se despliegan grandes modelos ontológicos. Por esta razón, y con la finalidad de mejorar el rendimiento de las visualizaciones, opcionalmente se puede deshabilitar esta técnica, mediante la opción *Visualisations -> Blurred background* como se muestra en la Figura A.7 (c).

A.11. Búsqueda de clases

La búsqueda de clases en una taxonomía es un aspecto crucial dentro del proceso de análisis de un modelo ontológico. La opción *Options -> Search* permite al usuario buscar una clase por nombre, y resalta todos los resultados de las clases que coincidan con el nombre introducido.

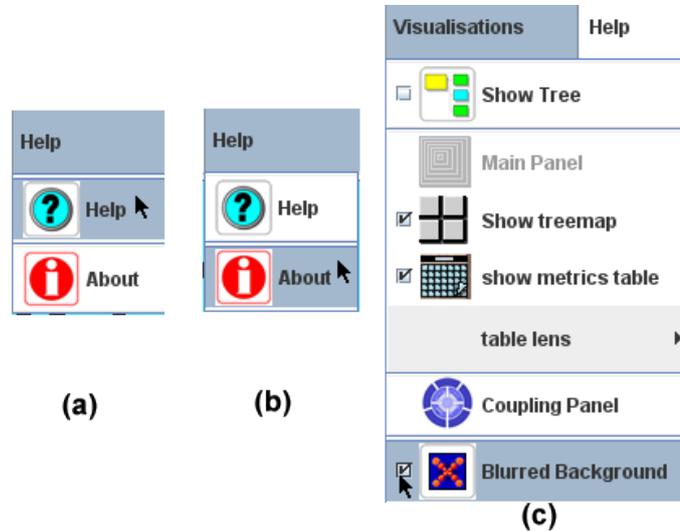


Figura A.7: Las figuras (a) y (b) muestran las opciones *Help* ->*Help* y *Help* ->*About*. La figura (c) muestra la opción que permite habilitar o deshabilitar la opción del difuminado del fondo. Esta técnica puede deshabilitarse para optimizar el rendimiento de las visualizaciones, especialmente para grandes ontologías.

A.12. Desplegar estadísticas

OWL-VisMod despliega dos tipos de estadísticas; por un lado, las estadísticas individuales de las clases en la ontología que se muestran con la visualización *table lens* y, por otro lado, estadísticas generales sobre el número total de clases en el modelo ontológico y el total de modelos agregados. La opción para desplegar estas estadísticas globales se muestran en la Figura A.8 (b).

A.13. Reiniciar las visualizaciones

Para reiniciar las visualizaciones a su estado inicial, se utiliza la opción del menú *Options* ->*Reset values*, que se muestra en la Figura A.8 (a).

A.14. Cambiar de visualización

Para cambiar de visualización se utiliza la opción *Visualisations*, que permite cambiar la visualización activa entre todas las posibles vistas globales,

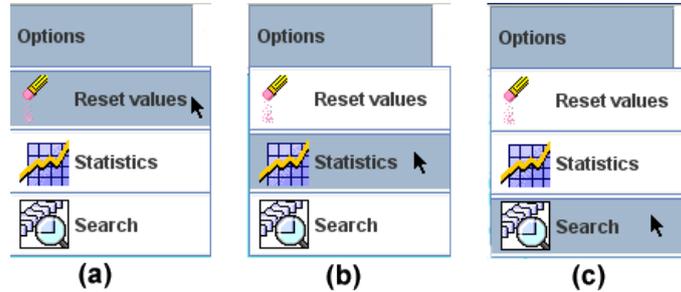


Figura A.8: La opción del menú *Options*, permite reiniciar las visualizaciones a los valores por omisión (Figura (a)), mostrar estadísticas del modelo ontológico cargado (Figura (b)) y la búsqueda de clases en la taxonomía del modelo ontológico (Figura (c)).

que son el árbol de jerarquías (Figura A.9 (a)), el *treemap* (Figura A.9 (b)), la tabla con las métricas (*table lens*) y la vista con el acoplamiento global (Figura A.9 (c)).

Por omisión, OWL-VisMod despliega la vista jerárquica del *treemap*, junto con la *table lens*, para combinar vistas enlazadas de taxonomía y métricas de las clases. La opción *Visualisations -> Show tree* que se muestra en la Figura A.9 (a), cambia la vista del *treemap* al árbol de jerarquías, y también puede incluir la *table lens*. Las vistas jerárquicas árbol y el *treemap*, pueden mostrarse en la pantalla completa, o pueden enlazarse con la vista del *table lens*. Incluso esta última vista (*table lens*), también puede desplegarse sin la vista jerárquica.

Por otro lado, la visualización del acoplamiento global reemplaza a las visualizaciones taxonómicas, esta opción *Visualisations -> Coupling panel* se muestra en la Figura A.9 (c).

A.15. Modificar el esquema de la ontología

El esquema de una ontología se refiere a las clases, las propiedades y las instancias. La Figura A.10 muestra las diferentes opciones para modificar el esquema, estas opciones se describen a continuación:

- *OWL Lite Schema -> create subclass.-* crea una subclase de la clase enfocada o seleccionada.
- *OWL Lite Schema -> edit class.-* permite editar la clase enfocada o seleccionada.

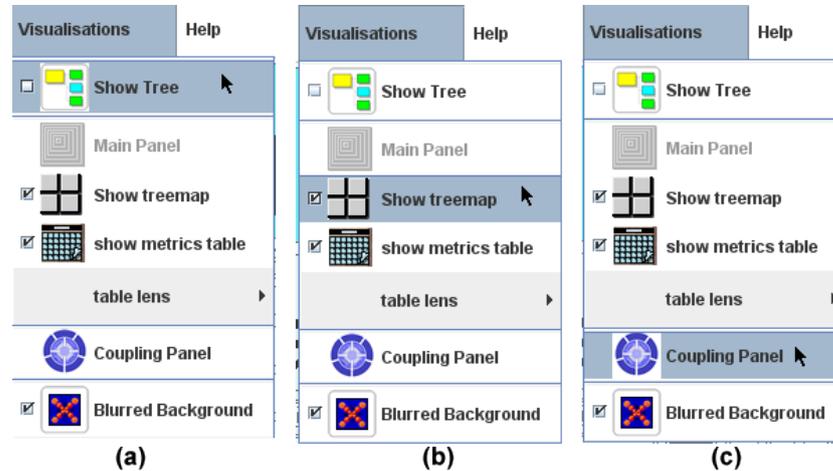


Figura A.9: La opción del menú para cambiar las diferentes visualizaciones disponibles en OWL-VisMod. Estas visualizaciones son el árbol de jerarquía (Figura (a)), el *treemap* (Figura (b)), la *table lens* y la vista del acoplamiento global (Figura (c)).

- *OWL Lite Schema ->create object property.-* crea una nueva propiedad objeto en la clase enfocada o seleccionada.
- *OWL Lite Schema ->create datatype property.-* crea una nueva propiedad de datos en la clase enfocada o seleccionada.
- *OWL Lite Schema ->add superclass.-* agrega una superclase a la clase enfocada o seleccionada.
- *OWL Lite Schema ->create individual.-* agrega una nueva instancia a la clase enfocada o seleccionada.

A.16. Crear una clase como unión o intersección

Una clase puede definirse como una unión o intersección de otras clases en el modelo ontológico, en el lenguaje OWL-DL. De esta forma, las clases pueden definirse como componentes lógicos de otras clases, bajo la definición de unión e intersección de la lógica y teoría de conjuntos. La opción *OWL Lite Intersection Class* se ilustra en la Figura A.11 (a).

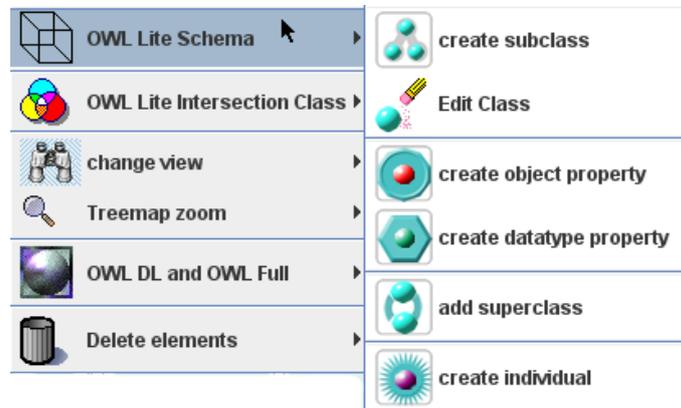


Figura A.10: Las diferentes opciones del esquema de una ontología, para crear clases, propiedades o instancias. Este submenú se invoca al presionar click con el botón derecho del mouse, sobre el área de la visualización.

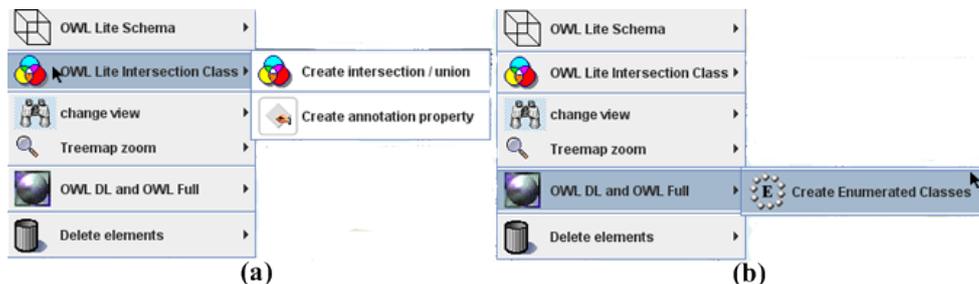


Figura A.11: (a) El lenguaje OWL-DL permite crear clases definidas como la unión o la intersección de otras clases existentes en el modelo. (b) El lenguaje OWL-DL permite definir clases enumeradas.

A.17. Navegación en los niveles del *treemap*

La navegación a través de los diversos niveles del *treemap* se hace con la opción *Treemap zoom* y la selección de los niveles inferior y superior que deben desplegarse en el *treemap*. Esta opción se invoca con el click derecho del *mouse* sobre la superficie del *treemap*. Esta navegación también puede hacerse con la combinación de teclas <SHIFT>+ <UP>y <SHIFT>+ <DOWN>.

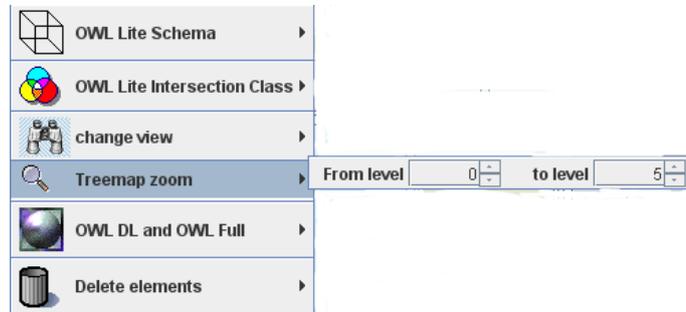


Figura A.12: La navegación a través de los diversos niveles del *treemap* se realiza con la opción *Treemap zoom*, invocada con el click derecho del *mouse*, colocado sobre la visualización.

A.18. Crear clases enumeradas

Las clases enumeradas se han incluido en el lenguaje OWL-DL, y son clases que se describen mediante la enumeración de sus elementos, que son instancias de la propia clase. La opción del menú para crear clases enumeradas se muestra en la Figura A.11 (b), y se invoca a través del click derecho del *mouse* sobre el área de la clase enfocada o seleccionada, que servirá como superclase de la nueva enumeración.

A.19. Eliminar clases, propiedades o instancias

La eliminación de clases del modelo ontológico requiere posicionar el cursor sobre la clase que se quiere eliminar, ya sea en la visualización del *treemap*, o en alguna de las visualizaciones de *zoom* semántico. Después, presionar el botón derecho del *mouse* para desplegar el menú de opciones, y seleccionar la opción *Delete elements ->Delete Class*, como se muestra en la Figura A.13 (a).

Para eliminar alguna propiedad de una clase, ya sea una propiedad objeto o una propiedad de datos, es necesario estar en la vista de detalle de la clase que contiene a la propiedad, mediante la vista de *zoom* semántico. Una vez en esta vista, es necesario posicionar el cursor sobre la propiedad que se desea eliminar, presionar click derecho sobre el *mouse* para desplegar el menú, y después seleccionar la opción *Delete elements ->Delete Property*, como se muestra en la Figura A.13 (b).

Análogamente, para eliminar una instancia de una clase, se repite el procedimiento descrito en el párrafo anterior y se selecciona la opción *Delete elements*

-> *Delete Individual*, como se muestra en la Figura A.13 (b).

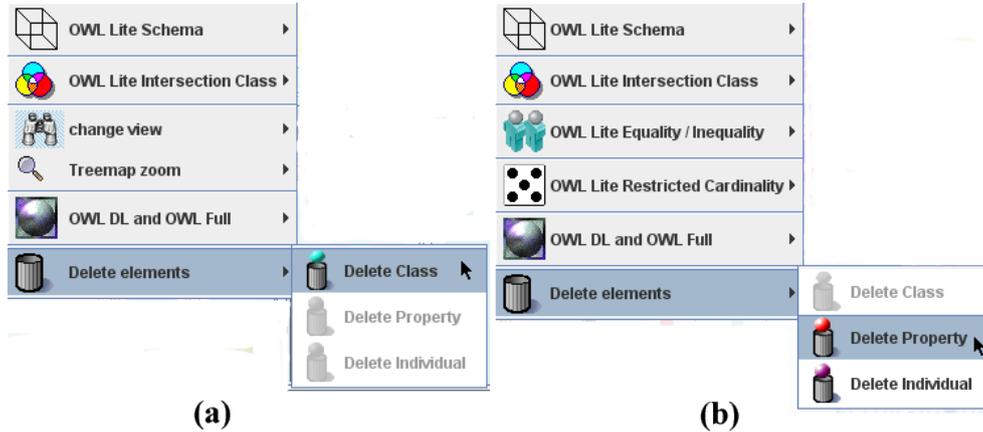


Figura A.13: La opción *Delete elements* permite eliminar clases, propiedades e instancias del modelo ontológico. La figura (a) muestra la opción que permite eliminar una clase, mientras que la figura (b) muestra las opciones para eliminar ya sea una propiedad o una instancia.

A.20. Opciones de interacción con la *table lens*

Cuando se ha desplegado la visualización *table lens*, existen diversas opciones de configuración para el análisis de las métricas representadas. Una de estas opciones es la posibilidad de cambiar los colores de los histogramas que representan los valores numéricos en las columnas. Esta opción se invoca con el botón derecho del *mouse* sobre el área de la *table lens*, acción que despliega el menú del cual se selecciona la opción *change colour*, y posteriormente la columna que se quiere cambiar de color, como se ilustra en la Figura A.14 (a). Posteriormente, aparece una ventana para seleccionar el nuevo color que será aplicado a los valores representados en la *table lens*. El uso de colores distintivos para los valores de las columnas, permite muy fácilmente identificar cada una de las columnas y distinguirlas entre sí. Cada columna contiene un color distintivo diferente de los demás, pero no existe restricción alguna en el uso de los colores en la tabla.

La principal interacción con esta técnica visual, es el recorrido de las diversas filas. Cuando hay demasiadas filas que no alcanzan a ser desplegadas en el pantalla, es necesario utilizar la función de *scroll* del dispositivo de entrada o *mouse*. Esta interacción permite desplazar las filas ya sea hacia abajo o hacia arriba según sea el caso.

Otra de las opciones es la posibilidad de ocultar ciertas columnas que no son del interés del usuario en determinado momento. Esta opción tiene la ventaja de permitir administrar el espacio de las columnas relevantes, y se muestra en la Figura A.14 (b). Esta funcionalidad también puede invocarse presionando el botón derecho del mouse sobre el encabezado de la columna que se quiere ocultar. La misma interacción sobre el borde visible cuando la columna está oculta, ocasiona que la columna vuelva a mostrarse completamente.

Otra funcionalidad agregada a la *table lens*, es la posibilidad de filtrar datos de acuerdo a un cierto criterio o rango numérico. Esta opción se muestra en la Figura A.14 (c). El filtrado de datos se realiza mediante la definición de una cota superior y otra inferior, que establecen un rango de valores. Todos los elementos cuyo valor se ubique dentro de este rango de valores definido, son mostrados por la *table lens*, mientras que el resto de valores se ocultan.

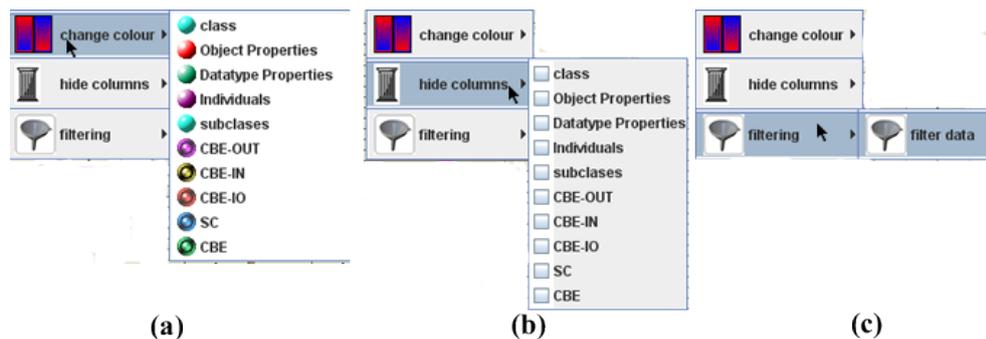


Figura A.14: La principal funcionalidad del *table lens* incluye cambios de colores para las columnas (figura (a)), ocultar determinadas columnas (figura (b)) y el filtrado de datos (figura (c)).

A.21. Opciones de interacción con la vista de acoplamiento global

La vista que muestra el acoplamiento global de la ontología descrita en la sección 5.3.3.3 tiene implementada diversa funcionalidad que enriquece esta técnica y el análisis del acoplamiento del modelo ontológico. La primera funcionalidad implementada es el filtrado de clases de acuerdo a las métricas de acoplamiento propuestas y descritas en la sección 5.1.4. Esta opción *Filter -> classes* se muestra en la Figura A.15. El filtrado de clases se realiza mediante la definición de una cota superior y otra inferior, que establecen un rango de valores para alguna de las métricas de acoplamiento definidas. Todas las clases

cuyo valor de la métrica seleccionada se ubique dentro de este rango de valores definido, se muestran en la vista de acoplamiento global, mientras que el resto de las clases se ocultan. Esta funcionalidad permite el análisis de las clases de acuerdo al criterio de las métricas de acoplamiento.

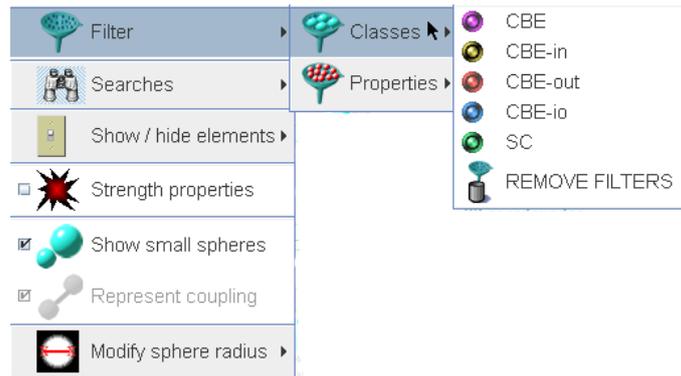


Figura A.15: El filtrado de clases permite analizar las clases en un cierto rango de acuerdo a las métricas de acoplamiento.

La opción *Filter -> Properties* que se muestra en la Figura A.16, permite filtrar las propiedades objeto, las propiedades de datos y todas aquellas propiedades que no tienen definido un valor en el dominio o en el rango. Estas propiedades con valores indefinidos pueden ocultarse para evitar la saturación de la visualización con todas las aristas. Esta opción de filtrado también ocultará las clases que como resultado de este filtro, resulten sin propiedades que deban ser mostradas, por lo que esta acción afecta tanto a las propiedades que se van a ocultar como a las clases.

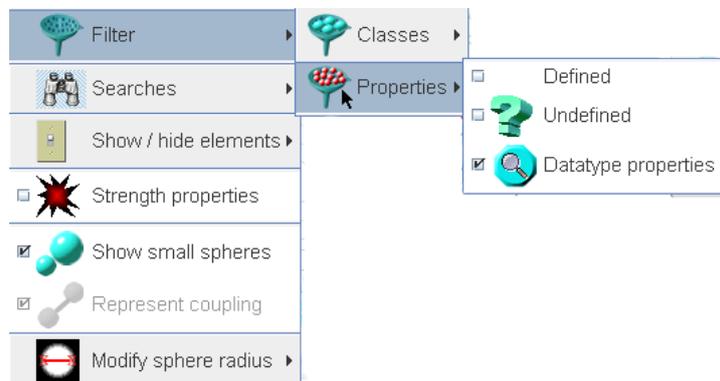


Figura A.16: El filtrado de propiedades permite ocultar todas aquellas aristas que no sean del interés del usuario.

Otra funcionalidad implementada es la búsqueda de clases o propiedades específicas en la vista de acoplamiento. Esta búsqueda se realiza con la opción *Searches* del menú que se despliega al presionar el botón derecho del *mouse*, como se muestra en la Figura A.17 (a).

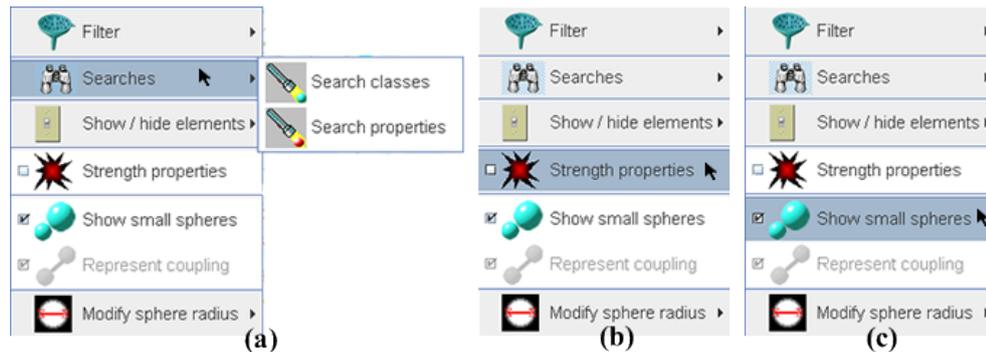


Figura A.17: Las opciones en la vista de acoplamiento global de los modelos ontológicos. La figura (a) muestra la opción para realizar búsquedas de clases o propiedades. La figura (b) muestra la opción para tensar las aristas que representan las propiedades, y la figura (c) muestra la opción para cambiar el tamaño de las esferas que representan a las clases acopladas.

La vista de acoplamiento global descrita en la sección 5.3.3.3, utiliza una distribución radial de las clases acopladas, con aristas que cruzan y unen a las clases. Cuando hay demasiadas aristas, estas pueden “tensarse” para mejorar la representación visual del acoplamiento. Esta opción se muestra en la Figura A.17 (b), y se invoca presionando el botón derecho del *mouse* sobre la superficie de la vista de acoplamiento.

Esta visualización también permite cambiar el tamaño de las esferas que representan las clases, así como la posición de los nombres de las clases en las mismas esferas. Estas opciones pueden configurarse en dependencia del número de clases que haya en la vista; para un mayor número de clases, lo mejor es la representación de las esferas pequeñas, mientras que por el contrario, para un menor número de clases, puede usarse la representación de las esferas grandes. Esta opción se muestra en la Figura A.17 (c), y contiene dos opciones, la primera es la selección de la representación de esferas grandes, y la segunda opción define esferas pequeñas.

La opción *Modify sphere radius* modifica el radio de la circunferencia en la que se despliegan las clases acopladas. Esta funcionalidad es útil en dependencia de la cantidad de clases en la ontología, ya que para un número mayor el diámetro de la circunferencia puede agrandarse, mientras que para el caso contrario, para una cantidad relativamente pequeña de clases, el diámetro de

la circunferencia puede disminuirse. Esta opción se invoca con el menú que se despliega presionando el botón derecho del *mouse*, sobre el área de la visualización, como se muestra en la Figura A.18.

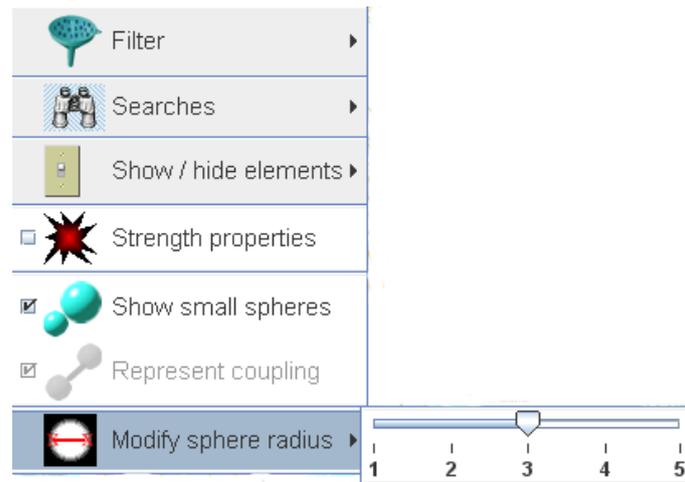


Figura A.18: La opción que modifica el radio de la circunferencia con las clases, en la vista de acoplamiento global de los modelos ontológicos.

A.22. Interacción con las listas circulares

La visualización del acoplamiento global despliega por un lado la lista con las clases acopladas, y por otro lado, la lista con las propiedades que definen el acoplamiento. Debido a que el número de clases y propiedades puede ser demasiado grande para ser desplegados todos los valores, la solución implementada es el uso de listas en forma circular. Este tipo de listas de forma circular, despliegan los elementos en grupos de cuarenta elementos, y hay dos mecanismos para cambiar los elementos desplegados mediante una interacción que “rota” la lista circular. Esta “rotación” de los elementos de la lista, se puede hacer mediante el control del menú visual, o mediante la función de “scroll up” o “scroll down” para rotar en un sentido o en otro las listas. Estas listas se ilustran en la Figura A.19; la Figura (a) muestra la lista de las clases y la Figura (b) muestra las propiedades, en donde el color identifica el tipo de propiedad, rojo para propiedades objeto y verde para propiedades de datos.

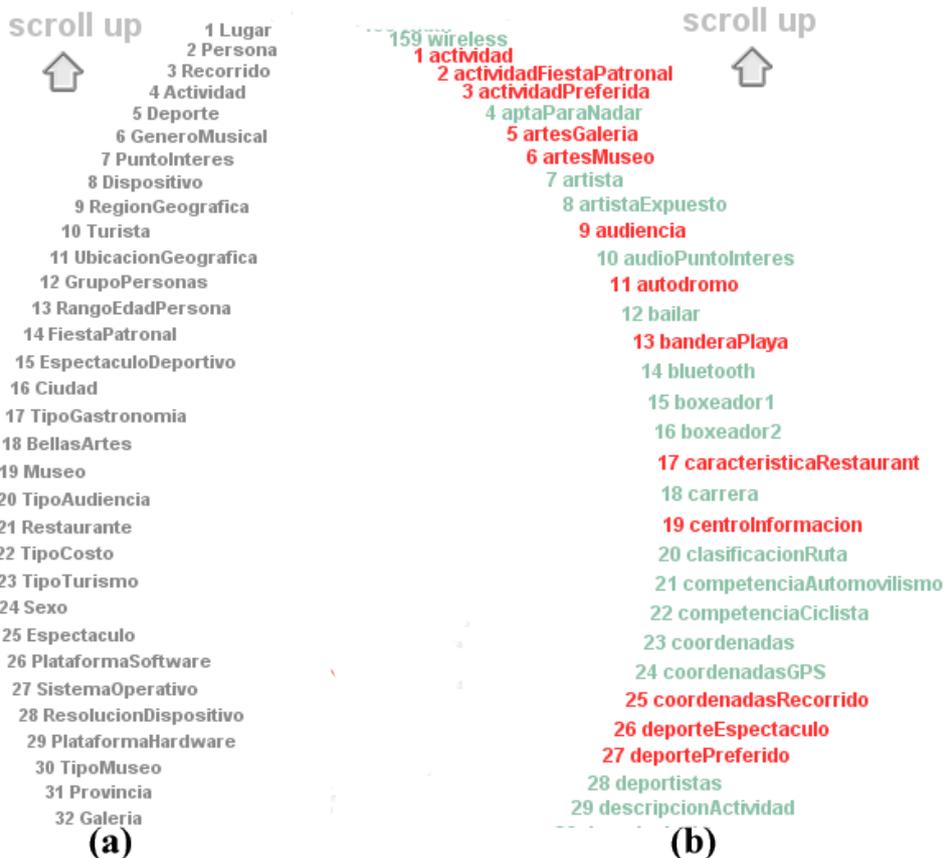


Figura A.19: Las listas circulares que muestran las clases (figura (a)) y las propiedades (figura (b)) de los modelos ontológicos. En la lista de propiedades, el color identifica el tipo de la propiedad, el color rojo identifica a las propiedades objeto, mientras que el color verde identifica a las propiedades de datos.

A.23. Menús visuales para clases y propiedades

La visualización de acoplamiento global contiene dos menús visuales con las opciones más comunes de acuerdo a la funcionalidad. La Figura A.20 muestra estos dos menús visuales, uno de los cuales es para las clases y el otro para las propiedades.

La Figura A.20 (a) muestra la opción para rotar la lista de clases que se localiza del lado izquierdo de la visualización, y que como se ha descrito en la sección A.22, también se puede interactuar y rotar esta lista con el “scroll” del *mouse*. La Figura A.20 (b) muestra la opción para realizar el filtrado de clases

de acuerdo a las métricas de acoplamiento, como se ha descrito en la sección A.21. La Figura A.20 (c) muestra la opción para realizar búsquedas de clases en la visualización de acoplamiento.

La Figura A.20 (d) muestra la opción para “rotar ” la lista con los nombres de las propiedades. La Figura A.20 (e) muestra la opción para realizar un filtrado de propiedades, como se ha descrito en la sección A.21. La Figura A.20 (f) muestra la opción que permite al usuario realizar búsquedas de propiedades de acuerdo al nombre. La Figura A.20 (g) muestra la opción para crear una nueva propiedad objeto (*object property*) en el modelo ontológico. Finalmente, la Figura A.20 (h) muestra la opción para crear una propiedad de datos (*datatype property*) en el modelo ontológico. Estas últimas dos opciones, despliegan una ventana en la que el usuario introduce la información de las propiedades que se van a crear.

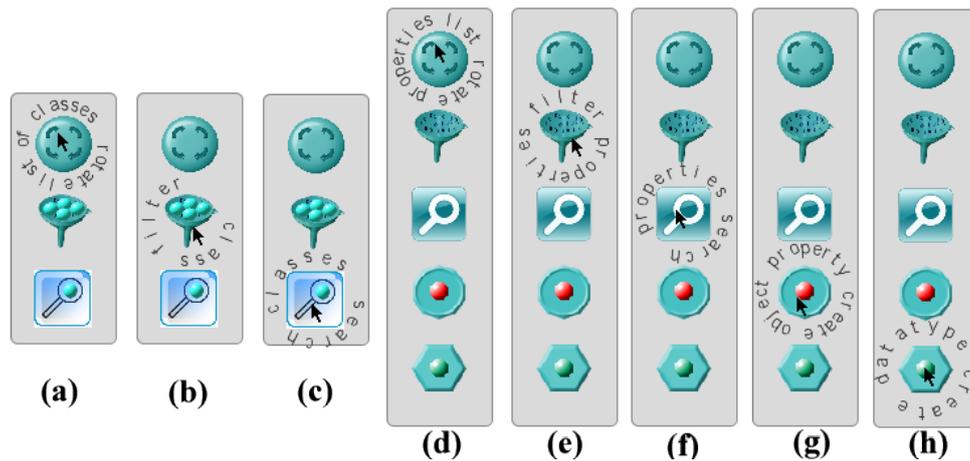


Figura A.20: Las figuras (a), (b) y (c) muestran el menú visual para la funcionalidad de las clases; rotar la lista, filtrar clases o la búsqueda de clases respectivamente. Las Figuras (d), (e), (f), (g) y (h) muestran el menú visual para la funcionalidad de las propiedades; rotar la lista, filtrar propiedades, búsqueda de propiedades, crear propiedades objeto y finalmente crear propiedades de datos respectivamente.

Apéndice B

Glosario de términos y acrónimos

Este glosario recopila los términos y acrónimos más comunes en el desarrollo de este trabajo de tesis.

- API (Application Programming Interface).- Un conjunto de funciones, procedimientos o métodos que ofrece una cierta biblioteca para ser utilizado por otro componente de software como una capa de abstracción.
- FTP (File Transfer Protocol).- Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol).
- GUI (Graphic User Interface).- Acrónimo que se refiere a los componentes gráficos que permiten la interacción de un usuario con un sistema de cómputo.
- HTML (HyperText Markup Language).- Es un lenguaje de marcado basado en etiquetas para la elaboración de páginas web.
- IEEE.- Es un consorcio internacional dedicado a la innovación tecnológica, a través de publicaciones, conferencias, estándares tecnológicos y diversas actividades educativas y profesionales.
- ISO (International Organization for Standardization).- Es una organización internacional para la definición de estándares de calidad.
- Métrica de *software*.- Es una medida de cierta propiedad o pieza de *software*. Las métricas se utilizan para proporcionar un estimado principalmente numérico, sobre cierto aspecto de un sistema de *software*.

- **OMG (Object Management Group).**- Es un consorcio internacional dedicado al desarrollo de los estándares de tecnologías orientadas a objetos.
- **OWL (Web Ontology Language).**- Es una recomendación del consorcio W3C para un lenguaje diseñado para usarse por aplicaciones que necesitan procesar el contenido de información, y facilita la interoperabilidad entre los equipos de cómputo.
- **PDA (Personal Digital Assistant).**- Un asistente personal digital, es un dispositivo móvil que permite administrar información personal, y cuenta con diversas capacidades tecnológicas como conexión a internet, audio, vídeo, telefonía, entre otras.
- **RDF (Resource Description Framework).**- Es un modelo estándar para el intercambio de datos en la Web, aprobado por el consorcio W3C.
- **RFC (Request for Comments).**- Son documentos que definen las propuestas de los protocolos que se usan en Internet.
- **SPARQL.**- Es un lenguaje para consultas sobre modelos ontológicos, recomendación del consorcio W3C.
- **SQL (Structured Query Language).**- Es un lenguaje declarativo de acceso a bases de datos, diseñado para administrar datos en bases de datos relacionales.
- **UML (Unified Modeling Language).**- Es el lenguaje unificado para el modelado de sistemas de software, respaldado por el OMG.
- **URI (Uniform Resource Identifier).**- Es un identificador uniforme de recurso, una cadena de caracteres corta que identifica inequívocamente un recurso, se utilizan para identificar los elementos en las ontologías.
- **W3C (World Wide Web Consortium).**- Es un consorcio internacional que regula el uso de la Web, a través de diversas directrices y especificaciones técnicas, que se hacen públicas como estándares.
- **Widget.**- Es una pequeña aplicación o programa, que generalmente se ejecuta por algún cierto motor de widgets.
- **XML (eXtensible Markup Language).**- Es un lenguaje de marcado que se basa en el uso de etiquetas, que se utiliza para el intercambio de datos. Se utiliza principalmente para el intercambio de información entre aplicaciones y sistemas de cómputo.

Apéndice C

Carta de colaboración empresa Interra

Como se ha mencionado previamente, el caso de estudio se desarrolló en colaboración con la empresa Interra. Se anexa una carta de conformidad por parte de la empresa, con respecto al acuerdo de colaboración suscrito.

Durante los meses que duró el diseño del modelo de datos para los proyectos Turimov y ULISES, hubo un constante intercambio de información con los analistas de la empresa para que los modelos ontológicos cumplieran con todas las especificaciones.



interra
ingeniería y recursos
Santo Domingo nº 2, 1º Dcha
38003 Santa Cruz de Tenerife
Tel. (+34) 922 276 710
Fax. (+34) 922 275 345



Parque Científico USAL, M2 Sur 02
C/del Adaja s.n. Villamayor
37185 Salamanca
Tel. y Fax (+34) 923 338 058

Web: www.interra.es Mail: proyectos@interra.es
proyectos.id@interra.es administracion.id@interra.es

D. Germán Hernández Durán, con D.N.I. 42883169Z, con domicilio a efectos de notificaciones en C/ Adaja s/n, Parque Científico de la Universidad de Salamanca, Edificio M2, Planta 02 Sur, del término municipal de Villamayor, código postal 37185, en representación de la empresa Interra, Ingeniería y Recursos, SLU (en adelante INTERRA), con C.I.F. B35573385

MANIFIESTA:

PRIMERO. Que en cumplimiento de las obligaciones derivadas del acuerdo de colaboración de fecha 7 de junio de 2011, suscrito de una parte por INTERRA, y de otra parte por el Doctorando D. Juan Francisco García Navarro, con NIE X7431972M, miembro del grupo de investigación GRiAL de la Universidad de Salamanca, el citado doctorando colaboró en la realización del modelado ontológico de los datos de los proyectos Turimov y ULISES.

SEGUNDO. Que esta colaboración ha permitido al doctorando por un lado, el uso de dos proyectos para la elaboración del caso de estudio de su tesis doctoral, y por otro lado, a INTERRA estrechar vínculos con los grupos de investigación de la Universidad de Salamanca, y asimismo, beneficiarse de la investigación que se realiza en estos grupos.

TERCERO. Que dicha colaboración ha consistido en la definición y creación de los diversos modelos ontológicos necesarios para el desarrollo de los proyectos Turimov y ULISES, los cuales han sido entregados y validados por INTERRA, manifestando la empresa su entera satisfacción con el resultado del convenio de colaboración.

Lo que se declara a los efectos oportunos,

En Salamanca a, 24 de noviembre de 2011

INTERRA, Ingeniería y Recursos, s.l.
C.I.F. B-35573385
proyectos@interra.es

Fdo.: Germán Hernández Durán

Bibliografía

- [1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *Journal IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, 2005.
- [2] Katifori Akrivi, Torou Elena, Vassilakis Costas, Lepouras Georgios, Halatsis Constantin, and Daradimos Elias. Historical archive ontologies - requirements, modeling and visualization. *IEEE*, 1:1 – 11, 2002.
- [3] Harith Alani. Tgviztab: An ontology visualisation extension for protégé. *Proceedings of Knowledge Capture (K-Cap'03), Workshop on Visualization Information in Knowledge Engineering*, 2003.
- [4] Harith Alani and Christopher Brewster. Metrics for ranking ontologies. *4th International EON Workshop, 15th International World Wide Web Conference*, 2006.
- [5] Harith Alani, Christopher Brewster, and Nigel Shadbolt. Ranking ontologies with aktiverank. *5th International Semantic Web Conference (ISWC)*, 2006.
- [6] Walter alexander Jungmeister and David Turo. Adapting treemaps to stock portfolio visualization. *Technical Report, University of Maryland, Center for Automation Research, Computer Vision Laboratory*, 1:10–16, 1992.
- [7] Edward Allen and Taghi Khoshgoftaar. Measuring coupling and cohesion: An information-theory approach. *Proceedings of Sixth International Software Metrics Symposium*, 1:119–127, 1999.
- [8] Mauricio Barcellos Almeida. A proposal to evaluate ontology content. *Applied Ontology*, 4:245–265, 2009.

- [9] Keith Andrews and Helmut Heidegger. Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. *Proceedings of IEEE Symposium on Information Visualization (InfoVis)*, 1:9–12, 1998.
- [10] Gennady Andrienko, Natalia Andrienko, Piotr Jankowski, Daniel Keim, Menno-Jan Kraak, Alan MacEachren, and Stefan Wrobel. Geovisual analytics for spatial decision support: Setting the research agenda. *International Journal of Geography Information Science*, 21:839–857, January 2007.
- [11] Margaret Anne-Storey, Mark Musen, John Silva, Casey Best, Neil Ernst, Ray Ferguson, and Natasha Noy. Jambalaya: Interactive visualization to enhance ontology authoring and knowledge acquisition in protégé. *in Protégé. Workshop on Interactive Tools for Knowledge Capture (K-CAP)*, 1:1–9, 2001.
- [12] Grigoris Antoniou and Frank Van Harmelen. A semantic web primer. *The MIT Press*, second edition:1–227, 2008.
- [13] Daniel Archambault, Tamara Munzner, and David Auber. Grouse: Feature-based, steerable graph hierarchy exploration. *Eurographics/IEEE-VGTC Symposium on Visualization (2007)*, 14(4):900–913, 2007.
- [14] Julio Arpírez, Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Webode in a nutshell. *AI Magazine*, 24(3):37–47, 2003.
- [15] Julio C. Arpírez, Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Webode: a scalable workbench for ontological engineering. *Proceedings of the First International Conference on Knowledge Capture (K-CAP)*, 1:6–13, 2001.
- [16] John Avery and John Yearwood. Supporting evolving ontologies by capturing the semantics of change. *Proceedings of AusWeb*, 1:1–8, 2004.
- [17] Ketan Babaria. Using treemaps to visualize gene ontologies. *University of Maryland*, 1:1–10, 2001.
- [18] Eric Baehrecke, Niem Dang, Ketan Babaria, and Ben Shneiderman. Visualization and analysis of microarray and gene ontology data with treemaps. *BMC Bioinformatics*, 1:12–14, june 2004.
- [19] Sikha Bagui and Richard Earp. *Database Design Using Entity-Relationship Diagrams*. CRC Press, 2011.

- [20] Ian Bailey. *Intelligence-Based Systems Engineering*, chapter Enterprise Ontologies: Better Models of Business, pages 327–342. Springer-Verlag, 2011.
- [21] John Bateman. Upper modeling: organizing knowledge for natural language processing. *Proceedings of the Fifth International Workshop on Natural Language Generation*, 1:54–61, 1990.
- [22] Sean Bechhofer, Ian Horrocks, Carole Goble, and Robert Stevens. Oiled: a reason-able ontology editor for the semantic web. *Joint German/Austrian conference on Artificial Intelligence (KI 01), Lecture Notes in Artificial Intelligence*, 2174:396–408, 2001.
- [23] Howard Beck and Helena Sofia Pinto. Overview of approach, methodologies, standards, and tools for ontologies. Technical report, University of Florida, Universidade Técnica de Lisboa, 2002.
- [24] Benjamin Bederson, Ben Shneiderman, and Martin Wattenberg. Ordered and quantum treemaps: Making effective use of 2d space to display hierarchies. *Transactions on Graphics (TOG)*, pages 833–854, 2002.
- [25] Palash Bera. *Using ontologies in the context of knowledge management systems*. PhD thesis, The University of British Columbia (Canada), 2007.
- [26] Tim Berners-Lee, James Hendler, and Ora Lassila. The semantic web. *Scientific American*, 2001.
- [27] Jacques Bertin. Graphics and graphic information processing. *Gruyter Press*, 1:1–273, 1977.
- [28] Alberto Del Bimbo, Marco Bertini, and CarloTorniai. Multimedia ontologies for video digital libraries. *European Digital Library*, 66:37–38, 2006.
- [29] Jorge Biolchini, Paula Gomes Mian, Ana Candida Cruz Natali, and Guilherme Horta Travassos. Systematic review in software engineering. Technical report, Systems Engineering and Computer Science Department, COPPE / UFRJ, 2005.
- [30] Renaud Blanch and Eric Lecolinet. Navigation techniques for zoomable treemaps. *Proceedings of UIST*, 1:49–50, 2006.
- [31] Renaud Blanch and Éric Lecolinet. Browsing zoomable treemaps: Structure-aware multi-scale navigation techniques. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1248–1253, 2007.

- [32] Elizabeth Sugar Boese. *An Introduction to Programming with Java Applets*. Jones and Bartlett, 2010.
- [33] John Bondy and Siva Murty. *Graph Theory with Applications*. Elsevier Science, 1976.
- [34] Elena Paslaru Bontas, Malgorzata Mochol, and Robert Tolksdorf. Case studies on ontology reuse. *Proceedings of 5th International Conference on Knowledge Management*, 1:1–8, 2005.
- [35] Alessio Bosca, Dario Bonino, and Paolo Pellegrino. Ontosphere: more than a 3d ontology visualization tool. *SWAP*, pages 1–15, 2005.
- [36] Erol Bozsak, Marcc Ehrig, Siegfried Handschuh, Siegfried H, Alexander Maedche, Andreas Hotho, Er Maedche, Boris Motik, Daniel Oberle, Christoph Schmitz, Nenad Stojanovic, Rudi, Steffen Staab, Ljiljana Stojanovic, and Valentin Zacharias. Kaon - towards a large scale semantic web. *Proceedings of EC-Web 2002, LNCS*, 1:304–313, 2002.
- [37] Janez Brank, Marko Grobelnik, and Dunja Mladenic. A survey of ontology evaluation techniques. *Proceedings of the Conference on Data Mining and Data Warehouses*, 1:166–170, 2005.
- [38] Mark Bruls, Kees Huizing, and Jarke Van Wijk. Squarified treemaps, data visualization. *In Proceedings of the joint Eurographics and IEEE TCVG Symposium on Visualization, Springer*, pages 33–42, 2000.
- [39] Stuart Card and Jock Mackinlay. The structure of the information visualization design space. *Proceedings of Infovis*, 1:92–101, 1997.
- [40] Stuart Card, Jock Mackinlay, and Benn Shneiderman. *Readings in Information Visualization- Using Vision to Think*. Morgan Kaufmann, 1998.
- [41] Núria Casellas. Ontology evaluation through usability measures. *Proceedings of the Confederated International Workshops on On the Move to Meaningful Internet Systems*, 1:594–603, 2009.
- [42] Vinay Chaudhri, Adam Farquhar, Richard Fikes, Peter Karp, and James Rice. Open knowledge base connectivity 2.0.3. Technical report, Knowledge Systems Laboratory, Stanford University, 1998.
- [43] Chaomei Chen. *Visualizing the Semantic Web: Xml-Based Internet and Information Visualization*. Springer-Verlag New York, 2002.
- [44] Chaomei Chen. *Information Visualization: beyond the horizon*. Springer-Verlag, 2006.

- [45] Shyam Chidamber and Chris Kemerer. A metrics suite for object oriented design. *IEEE Transactions on Software Engineering*, 20(6):476–493, 1994.
- [46] Luca Chittaro. Information visualization and its application to medicine. *Artificial Intelligence in Medicine*, 22:81–88, 2001.
- [47] Clare Churcher. *Beginning Database Design*. Apress, 2007.
- [48] Ricardo Colomo-Palacios, Angel Garcia-Crespo, Pedro Soto-Acosta, Marcos Ruano-Mayoral, and Diego Jimenez-Lopez. A case analysis of semantic technologies for r & d intermediation information management. *International Journal of Information Management*, 30(5):465–469, 2010.
- [49] Oscar Corcho, Mariano Fernández-López, and Asunción Gómez-Pérez. Methodologies, tools and languages for building ontologies. where is their meeting point?. *Data & Knowledge Engineering*, 1:41–64, 2003.
- [50] Oscar Corcho, Mariano Fernández-López, Asunción Gómez-Pérez, and Angel López-Cima. *Law and the Semantic Web. Legal Ontologies, Methodologies, Legal Information Retrieval, and Applications*, chapter Building legal ontologies with METHONTOLOGY and WebODE. Springer-Verlag, 2005.
- [51] Oscar Corcho and Asunción Gómez-Pérez. A roadmap to ontology specification languages. *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management*, 1:80–96, 2000.
- [52] Randall Davis, Howard Shrobe, and Peter Szolovits. What is a knowledge representation? *American Association for Artificial Intelligence*, 14:17–33, 1993.
- [53] Alberto del Bimbo, Marco Mugnaini, Pietro Pala, and Franco Turco. Visual querying by color perceptive regions. *Pattern Recognition*, 31(9):1241–1253, 1998.
- [54] Roberto Therón Francisco José García Peñalvo Diego Alonso Gómez Aguilar, Miguel Ángel Conde González. Reveling the evolution of semantic content through visual analysis. *Proceedings of the 11th IEEE International Conference on Advanced Learning Technologies (ICALT)*, 1:450–454, 2011.
- [55] Dragan Djuric, Dragan Gasevic, and Vladan Devedzc. Ontology modeling and mda. *Journal of Object Technology*, 4(1):109–128, 2005.

- [56] Julia Dmitrieva and Fons J. Verbeek. Multi-view ontology visualization. *Proceedings 11th International Protege Conference*, 1:1–4, 2009.
- [57] Julia Dmitrieva and Fons J. Verbeek. Node-link and containment methods in ontology visualization. *Proceedings of OWL: Experiences and Directions*, 1:1–8, 2009.
- [58] Mike O Docherty. *Object-oriented analysis and design: understanding system development with UML 2.0*. John Wiley & Sons, Ltd, 2005.
- [59] Pedro Maroun Eid. Tviz: A taxonomy visualization tool. Master’s thesis, Concordia University, 2005.
- [60] John Ellson, Emden R. Gansner, Eleftherios Koutsofios, Stephen C. North, and Gordon Woodhull. Graphviz and dynagraph - static and dynamic graph drawing tools. pages 127–148, 2003.
- [61] Lars Engel, Michael C. Jaeger, and Gero Muhl. Search and evaluation of ontologies for semantic web services in the internet. *Proceedings of IADIS International Conference on WWW/Internet*, 1:255–260, 2005.
- [62] Yuri Engelhardt. *The Language of Graphics*. PhD thesis, Institute for Logic, Language and Computation, University of Amsterdam, 2003.
- [63] Amineh Fadhil. Ontovql: Ontology visual query language. Master’s thesis, Concordia University, Montreal, Quebec, Canada, 2008.
- [64] Amjad Farooq, Syed Ahsan, and Abad Shah. Web-ontology design quality metrics. *Journal of American Science*, 6(11):52–58, 2010.
- [65] Amjad Farooq, Junaid Arshad, Aslam Muhamamd, and Saba Sana. Inconsistencies in ontologies. *Journal of American Science*, 7(7):800–803, 2011.
- [66] Adam Farquhar, Richard Fikes, and James Rice. The ontolingua server: A tool for collaborative ontology construction. *10th Knowledge Acquisition for Knowledge-Based Systems*, 1:1–19, 1996.
- [67] Dieter Fensel, Jürgen Angele, Stefan Decker, Michael Erdmann, Hans-Peter Schnurr, Steffen Staab, Rudi Studer, and Andreas Witt. On2broker: Semantic-based access to information sources at the www. 1999.
- [68] Dieter Fensel, Frank Harmelen, Y. Ding, M. Klein, and P. Mika. On-to-knowledge: Final report. Technical report, Semantic Technology Institute (STI), University of Innsbruck, Austria, 2007.

- [69] Dieter Fensel, Ian Horrocks, Frank Van Harmelen, S. Decker, M. Erdmann, and Mca Klein. Oil in a nutshell. *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling, and Management*, 1937:1–16, 2000.
- [70] Dieter Fensel, Ian Horrocks, Frank van Harmelen, Deborah L. McGuinness, and Peter Patel-Schneider. Oil: An ontology infrastructure for the semantic web. *IEEE Intelligent Systems*, 16 (2):38–45, 2001.
- [71] Mariano Fernández, Asunción Gómez, and Natalia Juristo. Methontology: From ontological art towards ontological engineering. *Proceedings of the Ontological Engineering AAAI-97 Spring Symposium Series*, 1:33–40, 1997.
- [72] Mariano Fernández-López, Asunción Gómez-Pérez, and María Rojas-Amaya. Ontology s crossed life cycles. *Proceedings of the 12th European Workshop on Knowledge Acquisition, Modeling and Management*, 1937:65–79, 2000.
- [73] Eileen Ferrance. Action research. Technical report, Northeast and Islands Regional Educational Laboratory At Brown University, 2000.
- [74] Daniel Fesenmaier, Karl Wober, and Hannes Werthner, editors. *Destination Recommendation Systems*. CAB International, 2006.
- [75] L. Fong, S.D. Larson, and A. Gupta. An ontology-driven knowledge environment for subcellular neuroanatomy: Owl: Experiences and directions. *CEUR Workshop Proceedings, ISSN 1613-0073*, 2007.
- [76] Frederico Fonseca and Max Egenhofer. Ontology-driven geographic information systems. *Proceedings of 7th ACM Symposium on Advances in Geographic Information Systems*, 1:14–19, 1999.
- [77] Frederico Fonseca, Max Egenhofer, Peggy Agouris, and Gilberto Camara. Using ontologies for integrated geographic information systems. *Transactions in GIS*, 6(3):231–257, 2002.
- [78] Thomas Franke. Enhancing an online regional tourism consulting system with extended personalized services. *Information Technology & Tourism*, 5:135–150, 2003.
- [79] Natalya Fridman and Deborah McGuinness. Ontology development 101: A guide to creating your first ontology. Technical report, Knowledge Systems, AI Laboratory, Stanford University (KSL-01-05), 2001.

- [80] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [81] Aldo Gangemi, Carola Catenacci, Massimiliano Ciaramita, and Jos Lehmann. Modelling ontology evaluation and validation. *The Semantic Web Research and Applications*, 3:140–154, 2006.
- [82] Carlos García, José Castellanos-Garzón, and Carlos Blanco. Analyzing gene expression data on a 3d scatter plot. *Proceedings of Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*, 87:349–356, 2011.
- [83] Juan García, Diego Alonso Gómez Aguilar, Antonio González, Francisco José García-Peñalvo, and Roberto Therón. A middleware framework to create data structures for a visual analytics object oriented approach. *International Journal of Knowledge and Learning (IJKL)*, 6(2/3):256–267, 2010.
- [84] Juan García, Francisco García, and Roberto Therón. Defining coupling metrics among classes in an owl ontology. *In Trends in Applied Intelligent Systems. 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2010, Springer*, 2010.
- [85] Juan García, Francisco García, and Roberto Therón. Visualising semantic coupling among entities in an owl ontology. *Proceedings of ONTOSE 2010, Ontology, Conceptualization and Epistemology for Information Systems, Software Engineering and Service Science, Springer*, 2010.
- [86] Juan García, Francisco García, and Roberto Therón. Modelling relationships among classes as semantic coupling in owl ontologies. *Proceedings of the 2011 International Conference on Information & Knowledge Engineering, IKE 2011,*, 1:22–28, 2011.
- [87] Juan García, Francisco García-Peñalvo, and Roberto Therón. Analizando el acoplamiento entre las clases de una ontología owl. *Actas de las XV Jornadas de Ingeniería del Software y Bases de Datos*, 1:207–216, 2010.
- [88] Juan García, Francisco García-Peñalvo, and Roberto Therón. Defining coupling metrics among classes in an owl ontology. *Proceedings of the 23rd international conference on Industrial engineering and other applications of applied intelligent systems*, 2:12–17, 2010.

- [89] Juan García, Francisco García-Peñalvo, and Roberto Therón. Evaluating the usability of owl-vismod: A modelling tool for owl ontologies. *Proceedings XII Congreso Internacional de Interacción Persona-Ordenador Interacción 2011*, 1:297–300, 2011.
- [90] Juan García, Francisco J. García-Peñalvo, and Roberto Therón. Through the data modelling process of turimov, an ontology-based project for mobile intelligent systems. *10th International Conference on Practical Applications of Agents and Multi-Agent Systems*, 2012.
- [91] Juan García, Francisco J. García-Peñalvo, Roberto Therón, and Patricia Ordoñez de Pablos. Usability evaluation of a visual modelling tool for owl ontologies. *Journal of Universal Computer Science*, 17(9):1299–1313, 2011.
- [92] Juan García, Antonio González, Diego Gómez, Roberto Therón, and Francisco García. A visual analytics tool for software project structure and relationships among classes. *Proceedings of 9th. International Symposium on Smart Graphics*, 1, 2009.
- [93] Juan García, Francisco García Peñalvo, and Roberto Therón. A survey on ontology metrics. *In Knowledge Management, Information Systems, E-Learning, and Sustainability Research. Third World Summit on the Knowledge Society, WSKS*, 111:22–27, 2010.
- [94] Juan García, Roberto Therón, and Francisco García. Visualisation of large ontologies by using advanced techniques. *Proceedings of I Workshop Web Mining and Semantic Web*, 1:1–10, 2008.
- [95] Juan García, Roberto Therón, and Francisco García. Visualisation of large software projects by using advanced techniques. *Innovations and Advances in Computer Sciences and Engineering*, 1:325–330, 2008.
- [96] Juan García, Roberto Therón, and Francisco García. Semantic zoom: A details on demand visualisation technique for modelling owl ontologies. *Highlights in Practical Applications of Agents and Multiagent Systems*, AISC 89:85–92, 2011.
- [97] Elena García-Barriocanal, Miguel Ángel Sicilia, and Salvador Sánchez-Alonso. Usability evaluation of ontology editors. *Knowledge Organization*, 32(1):1–9, 2005.
- [98] Elena García-Barriocanal, Miguel A. Sicilia, León A. González Sotos, and José R. Hilerá González. An ontology-based approach for designing web usability evaluation questionnaires. *Proceedings of the International Conference on Web Engineering*, 1:186–189, 2003.

- [99] Francisco-José García-Peñalvo, Juan García, and Roberto Therón. Analysis of the owl ontologies: A survey. *Scientific Research and Essays*, 6(20):4318–4329, 2011.
- [100] Juan Garcia, Diego A. Gomez, Antonio Gonzalez, Francisco J. Garcia, and Roberto Theron. A middleware framework to create and manage data structures for visual analytics. *Proceedings of Second World Summit on the Knowledge Society (WSKS)*, 49(3):466–473, 2009.
- [101] Angel Garcia-Crespo, Ricardo Colomo-Palacios, Juan Miguel Gomez-Berbis, and Belen Ruiz-Mezcua. Semo: a framework for customer social networks analysis based on semantics. *Journal of Information Technology*, 25 (2):178–188, 2010.
- [102] Alexander G. Gee, Georges G. Grinstein, Yu Min, and Li Hongli. Dynamic and interactive dimensional anchors for spring-based visualizations. Technical report, University of Massachusetts Lowell, Dept. of Computer Science, Lowell, 2005.
- [103] John Gennari, Mark Musen, Ray Ferguson, William Grosso, Monica Crubezy, Henrik Eriksson, Natalya Noy, and Samson Tu. The evolution of protege: An environment for knowledge-based systems development. *Stanford Medical Informatics*, 58(1):89–123, 2003.
- [104] Hiranmay Ghosh, Gaurav Harit, and Subhasis Chaudhury. Using ontology for building distributed digital libraries with multimedia contents. *World Digital Library*, 1(2):83–100, 2008.
- [105] Diego Gómez-Aguilar, Roberto Therón, and Francisco García-Peñalvo. Semantic spiral timelines used as support for e-learning. *Journal of Universal Computer Science (j-jucs)*, 15(7):1526–1545, 2009.
- [106] Asunción Gómez-Pérez. Evaluation of taxonomic knowledge in ontologies and knowledge bases. *Proceedings of the Banff Knowledge Acquisition for Knowledge-Based Systems*, 2:1 – 18, 1999.
- [107] Asunción Gómez-Pérez. Evaluating ontologies: Cases of study. special issue on verification and validation of ontologies. *IEEE Intelligent Systems and their Applications*, 16(3):391 – 409, 2001.
- [108] Asunción Gómez-Pérez. Evaluation of ontologies. *International Journal of Intelligent Systems*, 16(3), 2001.
- [109] Asunción Gómez-Pérez and Oscar Corcho. Ontology specification languages for the semantic web. *IEEE Intelligent Systems*, 17 (1):54–60, 2002.

- [110] Asunción Gómez-Pérez, Mariano Fernández, and Oscar Corcho. *Ontological Engineering*. Springer-Verlag, 2003.
- [111] Asunción Gómez-Pérez, Mariano Fernández-López, and Oscar Corcho. *Ontological Engineering, with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web*. Springer-Verlag, 2004.
- [112] Asunción Gómez-Pérez, Mari-Carmen Suárez, and Boris Villazón. Neon methodology for building ontology networks: Ontology specification. Technical report, NeOn Project, 2008.
- [113] Antonio González-Torres, Roberto Therón, Francisco José García Peñalvo, Michel Wermelinger, and Yijun Yu. Maleku: An evolutionary visual software analysis tool for providing insights into software evolution. *icsm 2011: Proceedings of IEEE 27th International Conference on Software Maintenance ICSM*, 1:594–597, 2011.
- [114] Bernardo Cuenca Grau, Ian Horrocks, Yevgeny Kazakov, and Ulrike Sattler. Modular reuse of ontologies: Theory and practice. *Journal of Artificial Intelligence Research*, 31:273–318, 2008.
- [115] Thomas Gruber. Toward principles for the design of ontologies used for knowledge sharing. *Kluwer*, 43:907–928, August 1993.
- [116] Thomas Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, pages 199–220, 1993.
- [117] Michael Gruninger. Designing and evaluating generic ontologies. *Proceedings of the 12th European Conference of Artificial Intelligence*, 1:53–64, 1996.
- [118] Nicola Guarino. Understanding, building and using ontologies. *International Journal of Human-Computer Studies*, 46(2-3):293–310, 1997.
- [119] Nicola Guarino and Pierdaniele Giaretta. Ontologies and knowledge bases. *Knowledge Acquisition*, Volume: 1:25–32, 1995.
- [120] Nicola Guarino and Chris Welty. Evaluating ontological decisions with ontoclean. *Communications of the ACM*, 45(2):61–65, 2002.
- [121] Nicola Guarino and Chris Welty. An overview of ontoclean. *The Handbook on Ontologies*, 1:151–159, 2004.
- [122] Jorge Gutiérrez, Miguel Ruiz, Roberto Herrera, Enrique Cabello, Steve Legrand, and Dave Elliman. Ontology languages for the semantic web: A never completely updated review. *Knowledge-Based Systems*, 19:489–497, 2006.

- [123] Peter Haase, Holger Lewen, Rudi Studer, and Michael Erdmann. The neon ontology engineering toolkit. In *Proceedings of the 17th International World Wide Web Conference*, 2008.
- [124] Maja Hadzic, Pornpit Wongthongtham, Tharam Dillon, and Elizabeth Chang, editors. *Ontology-Based Multi-Agent Systems*, chapter Ontology Design Approaches, pages 75–91. Springer-Verlag, 2009.
- [125] Terry Halpin. Object-role modeling: an overview. Technical report, Microsoft Corporation, 2002.
- [126] Terry Halpin. Object-role modeling version 2 (orm 2). *Encyclopedia of Database Systems*, 1:60–69, 2009.
- [127] Hans-Jorg Happel and Stefan Seedorf. Applications of ontologies in software engineering. *Proceedings of 2nd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2006)*, 1:1–14, 2006.
- [128] Frank Harary, Robert Norman, and Dorwin Cartwright. *Structural Models: An Introduction to the Theory of Directed Graphs*. John Wiley & Sons Inc, 1965.
- [129] Helwig Hauser. Generalizing focus+context visualization. Technical report, Scientific Visualization: Extracting Information and Knowledge from Scientific Data Sets, 2003.
- [130] Jeff Heflin, James Hendler, and Sean Luke. Shoe: A blueprint for the semantic web. *Spinning the Semantic Web*, 1:1–19, 2003.
- [131] Pippa Hemingway and Nic Brereton. What is a systematic review? *Hayward Medical Communications*, 2:1–8, 2009.
- [132] Ivan Herman, Guy Melancon, and Scott Marshall. Graph visualization and navigation in information visualisation: a survey. *IEEE Transactions on Visualisation and Computer Graphics*, 6(1):24–43, 2000.
- [133] Martin Hitz and Behzad Montazeri. Measuring coupling and cohesion in object-oriented systems. *Proceedings of International Symposium on Applied Corporate Computing*, 1:25–35, 1995.
- [134] Rami Hodrob and Mustafa Jarrar. Orm to owl 2 dl mapping. *Proceedings of the International Conference on Intelligent Semantic Web - Applications and Services*, 1:131–137, 2010.
- [135] Danny Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on visualization and computer graphics*, 12(5):741–748, 2006.

- [136] Tonya Hongsermeier and Vipul Kashyap. A knowledge management platform for translational medicine. In *AMIA 2005 Symposium Proceedings*, 2005.
- [137] Matthew Horridge, Simon Jupp, Georgina Moulton, Alan Rector, Robert Stevens, and Chris Wroe. A practical guide to building owl ontologies using protégé 4 and co-ode tools. Technical report, University of Manchester, 2007.
- [138] Matthew Horridge, Holger Knublauch, Alan Rector, Robert Stevens, and Chris Wroe. A practical guide to building owl ontologies using the protege-owl plugin and co-ode tools edition 1.0. 2004.
- [139] Ian Horrocks, Ulrike Sattler, and Stephan Tobies. Practical reasoning for expressive description logics. *Proceedings of LPAR '99 Proceedings of the 6th International Conference on Logic Programming and Automated Reasoning*, 1:161–180, 1999.
- [140] Watts S. Humphrey. *Managing the software process*. SEI Series in Software Engineering, 1989.
- [141] Petra Isenberg and Danyel Fisher. Collaborative brushing and linking for co-located visual analytics of document collections. *Eurographics/IEEE-VGTC Symposium on Visualization*, 28(3):1031–1038, 2009.
- [142] Ivar Jacobson and Stefan Bylund. *The Road to the Unified Software Development Process*. Cambridge University Press, 2000.
- [143] Mustafa Jarrar. *Towards Methodological Principles for Ontology Engineering*. PhD thesis, Vrije Universiteit Brussel, 2005.
- [144] Mustafa Jarrar, Jan Demey, and Robert Meersman. On using conceptual data modeling for ontology engineering. *Journal on Data Semantics*, 2800:185–207, 2003.
- [145] Mustafa Jarrar and Andriy Lisovoy. Dogmamodeler - a tool for ontology specification. Technical report, STARLab VUB, 2003.
- [146] Mustafa Jarrar and Robert Meersman. Formal ontology engineering in the dogma approach. *Proceedings of the International Conference on Ontologies, Databases and Applications of Semantics*, 2519:1238–1254, 2002.
- [147] Mustafa Jarrar and Robert Meersman. Scalability and knowledge reusability in ontology modeling. *Proceedings of International conference on the infrastructure for e-Business, e-Education, e-Science and e-Medicine*, 1:12–20, 2002.

- [148] Mustafa Jarrar and Robert Meersman. Ontology engineering -the dogma approach. *Advances in Web Semantics I*, 4891:7–34, 2008.
- [149] Yuhui Jin, Stefan Decker, and Gio Wiederhold. Ontowebber: Model-driven ontology-based web site management. *The 1st International Semantic Web Working Symposium (SWWS'01), Stanford University*, 2001:1–19, 2001.
- [150] Mathias John, Christian Tominski, and Heidrun Schumann. Visual and analytical extensions for the table lens. *Proceedings of Visualization and Data Analysis (SPIE)*, 6809:680907–680907–12, 2008.
- [151] Brian Johnson and Ben Shneiderman. Treemaps: a space filling approach to the visualization of hierarchical information structures. *Proceedings of the 2nd. IEEE Visualization Conference*, pages 284–291, 1991.
- [152] Yannis Kalfoglou and Marco Schorlemmer. Ontology mapping: the state of the art. *The Knowledge Engineering Review*, 18 (1):1–31, 2003.
- [153] Aditya Kalyanpur, Bijan Parsia, Evren Sirin, Bernardo Cuenca, and James Hendler. Swoop: A web ontology editing browser. *Journal of Web Semantics (JWS)*, 4(2):1–20, 2005.
- [154] Dimitris Kanellopoulos. An ontology-based system for intelligent matching of travellersneeds for group package tours. *International Journal of Digital Culture and Electronic Tourism*, 1(1):76–99, 2008.
- [155] Even-André Karlsson. *Software Reuse: A holistic approach*. John Wiley & Sons Ltd., 1995.
- [156] Peter Karp, Vinay Chaudhri, and Jerome Thomere. Xol: An xml-based ontology exchange language. Technical report, Artificial Intelligence Center SRI International, 1999.
- [157] Vipul Kashyap, Cartic Ramakrishnan, and Thomas C Rindflesch. Towards (semi-)automatic generation of bio-medical ontologies. In *AMIA 2003 Symposium Proceedings*, page 886, 2003.
- [158] Vipul Kashyapa, Alfredo Morales, and Tonya Hongsermeiera. On implementing clinical decision support: Achieving scalability and maintainability by combining business rules and ontologies. In *AMIA 2006 Symposium Proceedings*, pages 414–419, 2006.
- [159] Akrivi Katifori, Constantin Halatsis, George Leopuras, Costas Vassilakis, and Eugenia Giannopoulou. Ontology visualization methods- a survey. volume 39, pages 1–43. *ACM Computing*, 2007.

- [160] Akrivi Katifori, Elena Torou, Costas Vassilakis, Georgios Lepouras, and Constantin Halatsis. A comparative study of four ontology visualization techniques in protege: Experiment setup and preliminary results. *Information Visualization, IEEE*, 1:1–7, 2006.
- [161] Mikhail Kazakov and Habib Abdulrab. DL-workbench: a metamodeling approach to ontology manipulation. *Proceedings of EON Conference*, 1:10–26, 2003.
- [162] Mikhail Kazakov and Habib Abdulrab. A meta-modeling approach to ontological engineering: DL-workbench platform. *Metainformatics*, 3002:17–33, 2004.
- [163] Daniel Keim. Visual exploration of large data sets. *Communications of the ACM*, 44:38–44, 2001.
- [164] Daniel Keim, Gennady Andrienko, Jean-Daniel Fekete, Carsten Görg, Jörn Kohlhammer, and Guy Melançon. Visual analytics: Definition, process, and challenges. pages 154–175. 2008.
- [165] Daniel Keim, Jörn Kohlhammer, Geoffrey Ellis, and Florian Mansmann. *Mastering the Information Age Solving Problems with Visual Analytics*. Eurographics Association, 2010.
- [166] Daniel Keim, F. Mansmann, J. Schneidewind, and T. Schreck. Monitoring network traffic with radial traffic analyzer. *IEEE Symposium on Visual Analytics Science and Technology*, pages 123–128, 2006.
- [167] Daniel A. Keim, Florian Mansmann, Daniela Oelke, and Hartmut Ziegler. Visual analytics: Combining automated discovery with interactive visualizations. In *DS 08: Proceedings of the 11th International Conference on Discovery Science*, pages 2–14, Berlin, Heidelberg, 2008. Springer-Verlag.
- [168] Andreas Kerren, John Stasko, Jean-Daniel Fekete, and Chris North. *Information Visualization - Human-Centered Issues and Perspectives*. Springer-Berlag, 2008.
- [169] Purvesh Khatri and Sorin Draghici. Ontological analysis of gene expression data: current tools, limitations, and open problems. *Bioinformatics*, 21(18):3587–3595, 2005.
- [170] M. Rahamatullah Khondoker and Paul Mueller. Comparing ontology development tools based on an online survey. *Proceedings of the World Congress on Engineering 2010*, 1, 2010.

- [171] Barbara Kitchenham. Procedures for performing systematic reviews. Technical report, Keele University and NICTA, 2004.
- [172] Holger Knublauch, Ray Fergerson, Natalya Noy, and Mark Musen. The protege owl plugin: An open development environment for semantic web applications. *Proceedings of The Semantic Web (ISWC)*, 3298:229–243, 2004.
- [173] Cris Kobryn, Grady Booch, Ivar Jacobson, and Jim Rumbaugh, editors. *UML distilled: a brief guide to the standard object modeling language*, volume 3. Addison-Wesley, 2004.
- [174] Haridimos Kondylakis, Giorgos Flouris, and Dimitris Plexousakis. Ontology and schema evolution in data integration: Review and assessment. In *Proceedings of the Confederated International Conferences, CoopIS, DOA, IS, and ODBASE 2009 on On the Move to Meaningful Internet Systems: Part II*, OTM 2009, pages 932–947, Berlin, Heidelberg, 2009. Springer-Verlag.
- [175] Robert Kosara, Silvia Miksch, and Helwig Hauser. Focus+context taken literally. *IEEE Computer Graphics and Applications*, 22(1):22–29, 2002.
- [176] Kouji Kozaki, Yoshinobu Kitamura, and Riichiro Mizoguchi. Developing ontology-based applications using hozo. *Proceedings of the Fourth IASTED International Conference on Computational Intelligence*, 1:10–16, 2005.
- [177] John Lamping and Ramana Rao. The hyperbolic browser : A focus + context technique for visualizing large hierarchies. *Journal of Visual Languages and Computing*, 7:33 – 55, 1996.
- [178] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 1:401–408, 1995.
- [179] Jonathan Lazar, Jinjuan Feng, and Harry Hochheiser. *Research Methods in Human-Computer Interaction*. John Wiley and Sons, Ltd., Publication, 2010.
- [180] Bongshin Lee, Cynthia Sims Parr, Dana Campbell, , and Benjamin B. Bederson. How users interact with biodiversity information using tax-ontree. *Proceedings of Advanced Visual Interfaces (AVI)*, 1:320–327, 2004.

- [181] Douglas Lenat and R Guha. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Addison-Wesley Longman Publishing Co, 1989.
- [182] Zhanjun Li, Maria Yang, and Karthik Ramani. A methodology for engineering ontology acquisition and validation. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 23:37–51, 2009.
- [183] Thorsten Liebig and Olaf Noppens. Ontotrack: Fast browsing and easy editing of large ontologies. *Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools*, 1:47–56, 2003.
- [184] Thorsten Liebig and Olaf Noppens. Ontotrack: Combining browsing and editing with reasoning and explaining for owl lite ontologies. *Proceedings of the 3rd International International Semantic Web Conference*, 1:244–258, 2004.
- [185] Robert Lintern and Margaret-Anne Storey. Jambalaya express: ontology visualization-on-demand. In *Proceedings of the 8th International Protégé Conference*, pages 1–3, 2005.
- [186] Miron Livny, Raghu Ramakrishnan, Kevin Beyer, Guangshun Chen, Donko Donjerkovic, Shilpa Lawande, Jussi Myllymaki, and Kent Wenger. Devise: integrated querying and visual exploration of large datasets. *SIGMOD Rec*, 26(2):301–312, 1997.
- [187] Adolfo Lozano-Tello and Asunción Gómez-Pérez. Ontometric: A method to choose the appropriate ontology. *Journal of Database Management*, 15(2):1–18, 2004.
- [188] Sean Luke and Jeff Heflin. Shoe 1.01. proposed specification. Technical report, SHOE Project technical report, University of Maryland, 2000.
- [189] Alexander Maedche, Boris Motik, Ljiljana Stojanovic, Rudy Studer, and Raphael Volz. Managing multiple ontologies and ontology evolution in ontologging. *Intelligent Information Processing*, 1:51–63, 2002.
- [190] Carlos Marcondes, Maria Mendon, Luciana Malheiros, Leonardo Costa, and Tatiana Santos. Ontologies as the new bases of scientific knowledge. *Perspectivas em Ciencia da Informacion*, 13:20 – 39, 12 2008.
- [191] Carlos Marcondes and Marilia Rocha. Ontological and conceptual bases for a scientific knowledge model in biomedical articles. *Electronic Journal of Communication, Information & Innovation in Health*, 3:19–30, 2009.

- [192] Fernandez Mariano. Overview of methodologies for building ontologies. 1999.
- [193] Fernandez Mariano and Asuncion Gomez. A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies. 2002.
- [194] Deborah McGuinness, Richard Fikes, James Hendler, and Lynn Andrea Stein. Daml+oil: An ontology language for the semantic web. *IEEE Intelligent Systems*, 17 (5):72–80, 2002.
- [195] Deborah Mcguinness, Richard Fikes, Lynn Stein, and James Hendler. Daml-ont: An ontology language for the semantic web. In *Dagstuhl Seminars*, pages 65–93, 2003.
- [196] Alexander Mikroyannidis. Toward a social semantic web. *Database Management & Information Retrieval*, 40:113–115, 2007.
- [197] Riichiro Mizoguchi. Tutorial on ontological engineering: Advanced course of ontological engineering. *New Generation Computing*, 22(2):198–220, 2004.
- [198] Gonzalo Méndez. *Especificación de Requisitos según el estándar de IEEE 830*. Facultad de Informática, Universidad Complutense de Madrid, 2008.
- [199] David Modjeska. Navigation in electronic worlds: A research review. Technical report, Computer Systems Research Group, University of Toronto, 1997.
- [200] Daniel Moody. The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35 (6), 2009.
- [201] Enrico Motta. An overview of the ocml modelling language. In *Proceedings KEML'98: 8th Workshop on Knowledge Engineering Methods & Languages*, pages 21–22, 1998.
- [202] Jakob Nielsen. Finding usability problems through heuristic evaluation. *Proceedings of the Computer-Human Interaction Conference*, 1:373–380, 1992.
- [203] Jakob Nielsen. *Usability Engineering*. Morgan Kaufmann, 1993.
- [204] Jakob Nielsen. Enhancing the explanatory power of usability heuristics. *Proceedings ACM CHI Conference*, 1:152–158, 1994.

- [205] Héctor Oscar Nigro and Sandra Elizabeth González Císaro, editors. *Ontologies-Driven Web Mining: Concepts and Techniques*. IGI Global (formerly Idea Group Inc.), 2011.
- [206] Olaf Noppens and Thorsten Liebig. Interactive visualization of large owl instance sets. In *Proc. of the Third Int. Semantic Web User Interaction Workshop (SWUI 2006)*, 2006.
- [207] Chris North. Multiple views and tight coupling in visualization: A language, taxonomy, and system. In *Proc. CSREA CISST 2001 Workshop of Fundamental Issues in Visualization*, pages 626–632, 2001.
- [208] Chris North, Nathan Conklin, Kiran Indukuri, and Varun Saini. Visualization schemas and a web-based architecture for custom multiple-view visualization of multiple-table databases. *Information Visualization*, 1(3/4):211–228, 2002.
- [209] Chris North and Ben Shneiderman. Snap-together visualization: a user interface for coordinating visualizations via relational schemata. In *Proceedings of the working conference on Advanced visual interfaces*, pages 128–135, 2000.
- [210] Natalya F. Noy and Deborah L. McGuinness. *Ontology Development 101: A Guide to Creating Your First Ontology*. Stanford Knowledge Systems Laboratory, March 2001.
- [211] Leo Obrst, Benjamin Ashpole, Werner Ceusters, Inderjeet Mani, Steve Ray, and Barry Smith. The evaluation of ontologies: Toward improved semantic interoperability. Technical report, US National Institute of Standards and Technology, 2006.
- [212] Jeff Offutt, Mary Jean Harrold, and Priyadarshan Kolte. A software metric system for module coupling. *The Journal of Systems and Software*, 20(3):295–308, 1993.
- [213] Annika Ohgren and Kurt Sandkuhl. Towards a methodology for ontology development in small and medium-sized enterprises. In *Proceedings of IADIS International Conference on Applied Computing*, pages 369–376, 2005.
- [214] Mamoru Ohta, Kouji Kozaki, and Riichiro Mizoguchi. An extension of an environment for building/using ontologies "hozo" toward practical ontology engineering. *Proceedings of the IASTED International Conference if Software Engineering and Applications*, 1:373–379, 2010.

- [215] Anthony Orme, Haining Yao, and Letha Etzkorn. Coupling metrics for ontology-based systems. *Software, IEEE*, 2006.
- [216] Ekaterina Ovchinnikova, Tonio Wandmacher, and Kai-Uwe Kühnberger. Solving terminological inconsistency problems in ontology design. *IBIS - Interoperability in Business Information Systems*, 1(1):1–16, 2006.
- [217] Bostjan Pajntar, Marko Grobelnik, and Dunja Mladenic. Contextualized visualization of ontologies and ontology networks. *Proceedings of SIKDD*, 1:1–4, 2009.
- [218] Viktoria Pammer and Stefanie Lindstaedt. Ontology evaluation through assessment of inferred statements: Study of a prototypical implementation of an ontology questionnaire for owl dl ontologies. *Proceedings of Knowledge Science, Engineering and Management (KSEM)*, 1:394–405, 2009.
- [219] Jeff Z. Pan and Ian Horrocks. Reasoning in the shoq(dn) description logic. *Proceedings of the International Workshop on Description Logics*, 1:15–25, 2002.
- [220] Cynthia Sims Parr, Bongshin Lee, Dana Campbell, and Benjamin B. Bederson. Visualizations for taxonomic and phylogenetic trees. *Bioinformatics*, 20(17):2997–3004, 2004.
- [221] Bijan Parsia, Taowei Wang, and Jennifer Golbeck. Visualizing web ontologies with cropcircles. *ISWC*, pages 1 – 8, 2005.
- [222] Alan Perlis. *Software Reusability: Concepts and Models*. ACM Press, 1989.
- [223] Catherine Plaisant, Jesse Grosjean, and Benjamin Bederson. Spacetree: Supporting exploration in large node link tree, design evolution and empirical evaluation. *IEEE Symposium on Information Visualization, (InfoVis)*, 1:57–64, 2002.
- [224] Roberto Poli. Ontological methodology. *International Human-Computer Studies*, 56:639–664, 2002.
- [225] Rubén Prieto-Díaz and Guillermo Arango. *Domain Analysis and Software Systems Modeling*. IEEE, 1991.
- [226] Ramana Rao and Stuart K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In ACM, editor, *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. ACM, 1994.

- [227] Steven Reiss. A visual query language for software visualization. *Proceedings of the IEEE 2002 Symposia on Human Centric Computing Languages and Environments*, 1:80–88, 2002.
- [228] Walky Rivadeneira and Benjamin Bederson. A study of search result clustering interfaces: Comparing textual and zoomable interfaces. *University of Maryland HCIL Tech. Rep. HCIL-2003-36*, 21:1–5, 2003.
- [229] Dietmar Rosner and Manfred Stede. Generating multilingual documents from a knowledge base: the techdoc project. *Proceedings of the 15th conference on Computational linguistics*, 1:339–343, 1994.
- [230] Winston Royce. Managing the development of large software systems. *Proceedings of IEEE WESCON*, 1:1–9, 1970.
- [231] Jeffrey Saffer, Vicki Burnett, Guang Chen, and Peter Van der Spek. Visual analytics in the pharmaceutical industry. *Computer Graphics and Applications, IEEE*, 24:10–15, 2004.
- [232] Rodrigo Santamaría and Roberto Therón. Visualization of intersecting groups based on hypergraphs. *IEICE TRANSACTIONS on Information and Systems*, 7:1957–1964, 2010.
- [233] Óscar Corcho, Asunción Gómez-Pérez, Rafael González-Cabero, and Carmen Suárez-Figueroa. Odeval: a tool for evaluating rdf(s), daml+oil, and owl concept taxonomies. *Proceedings of IFIP Conference on Artificial Intelligence Applications and Innovations*, 1:369–382, 2004.
- [234] Jean Scholtz. Usability evaluation. Technical report, National Institute of Standards and Technology (NIST), 2004.
- [235] Helen Sharp, Yvonne Rogers, and Jenny Preece. *Interaction design : beyond human-computer interaction*. Wiley & Sons Ltd., second edition, 2007.
- [236] Ben Shneiderman. Extreme visualization: squeezing a billion records into a million pixels. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 3–12, 2008.
- [237] Harri Siirtola. Combining parallel coordinates with the reorderable matrix. In *Proceedings of the conference on Coordinated and Multiple Views In Exploratory Visualization*, 2003.
- [238] Elena Simperl. Reusing ontologies on the semantic web: A feasibility study. *Data & Knowledge Engineering*, 68:905–925, 2009.

- [239] Evren Sirin, Bijan Parsia, Bernardo Cuenca Grau, Aditya Kalyanpur, and Yarden Katz. Pellet: A practical owl-dl reasoner. *Web Semantics: Science, Services and Agents on the World Wide Web*, 5(2):51 – 53, 2007.
- [240] Chakkrit Snae and Michael Brueckner. Ontology-driven e-learning system based on roles and activities for thai learning environment. *Interdisciplinary Journal of Knowledge and Learning Objects*, 3:1–17, 2007.
- [241] Peter Spyns, Robert Meersman, and Mustafa Jarrar. Data modelling versus ontology engineering. *SIGMOD Rec.*, 31:12–17, 2002.
- [242] John Stasko and Eugene Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. *Infovis 2000*, 1:57–70, 2000.
- [243] Robert Stevens, Mikel Egana Aranguren, Katy Wolstencroft, Ulrike Sattler, Nick Drummond, Matthew Horridge, and Alan Rector. Using owl to model biological knowledge. *International Journal of Human-Computer Studies*, 2007.
- [244] Evelyn Stiller and Cathie LeBlanc. *Project-Based Software Engineering: An Object-Oriented Approach*. Addison-Wesley, 2002.
- [245] Vijayan Sugumaran and Veda Storey. Ontologies for conceptual modeling: their creation, use and management. *Data & Knowledge Engineering*, 42(3):251–271, 2002.
- [246] Bongwon Suh and Benjamin Bederson. Ozone: a zoomable interface for navigating ontology information. *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI)*, 1:139 – 143, 2002.
- [247] Eiichi Sunagawa, Kouji Kozaki, Yoshinobu Kitamura, and Riichiro Mizoguchi. A framework for organizing role concepts in ontology development tool: Hozo. *Roles, an Interdisciplinary Perspective: Ontologies, Programming Languages, and Multiagent Systems*, 1:136–143, 2005.
- [248] Kaustubh Supekar, Chintan Patel, and Yugyung Lee. Characterizing quality of knowledge on semantic web. *Proceedings of AAAI Florida AI Research Symposium (FLAIRS)*, 1:20–26, 2004.
- [249] York Sure, Juergen Angele, and Steffen Staab. Ontoedit: Guiding ontology development by methodology and inferencing. In *Coop-IS/DOA/ODBASE*, pages 29–31, 2002.
- [250] York Sure, Steffen Staab, Rudi Studer, and Ontoprise Gmbh. On-to-knowledge methodology (otkm). In *Handbook on Ontologies, International Handbooks on Information Systems*, pages 117–132. Springer, 2003.

- [251] Kamidi Suresh, Sanjay Kumar Malik, Nupur Prakash, and SAM Rizvi. Role of ontology editors : Ontology design. *Proceedings of IPCV08*, 2008.
- [252] Mari-Carmen Suárez-Figueroa, Saartje Brockmans, Aldo Gangemi, Asunción Gómez-Pérez, Jos Lehmann, and Holger Lewen. Neon modelling components. neon deliverable d5.1.1, neon project. Technical report, NeOn project, 2007.
- [253] Samir Tartir and Budak Arpinar. Ontology evaluation and ranking using ontoqa. *Proceedings of the International Conference on Semantic Computing*, 2007.
- [254] Samir Tartir, Budak Arpinar, Michael Moore, Amit Sheth, and Boanerges Aleman-meza. Ontoqa: Metric-based ontology quality analysis. *Scientific Literature Digital Library and Search Engine*, pages 1–9, 2005.
- [255] Samir Tartir, Budak Arpinar, and Amit Sheth. Ontological evaluation and validation. *Theory and Applications of Ontology: Computer Applications*, 1:115–130, 2010.
- [256] Alexandru Telea and Lucian Voinea. Case study: Visual analytics in software product assessments. In *VISSOFT'09*, pages 65–72, 2009.
- [257] Soon Tee Teoh and Kwan-Liu Ma. Rings : A technique for visualizing large hierarchies. *Proceedings of the 10th International Symposium on Graph Drawing*, 1:268–275, 2002.
- [258] Roberto Therón. Visual analytics of paleoceanographic conditions. In Pak Chung Wong and Daniel A. Keim, editors, *IEEE VAST*, pages 19–26. IEEE, 2006.
- [259] James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005.
- [260] Christian Tominski, James Abello, Frank Van Ham, and Heidrun Schumann. Fisheye tree views and lenses for graph visualization. In *Proceedings of the conference on Information Visualization*, pages 17–24, Washington, DC, USA, 2006. IEEE Computer Society.
- [261] Christian Tominski, P. Schulze-Wollgast, and Heidrun Schumann. Visual methods for analyzing human health data. *Encyclopedia of Healthcare Information Systems*, 1:1357–1364, 2008.
- [262] Mike Uschold. Building ontologies towards a unified methodology. *Proceedings of Expert Systems*, 1:1–20, 1996.

- [263] Mike Uschold and Michael Gruninger. Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11(2):93–136, 1996.
- [264] Mike Uschold and Martin King. Towards a methodology for building ontologies. *Proceedings of Workshop on Basic Ontological Issues in Knowledge Sharing*, 1:1–15, 1995.
- [265] Mike Uschold, Martin King, Stuart Moralee, and Yannis Zorgios. The enterprise ontology. *The Knowledge Engineering Review*, 13:31–89, 1995.
- [266] David Vallet, Miriam Fernández, and Pablo Castells. An ontology-based information retrieval model. *Proceedings of Extended Semantic Web Conference (ESWC 2005)*, 1:455–470, 2005.
- [267] Michel Vanden-Bossche, Peter Ross, Ian Maclarty, Bert Van Nuffelen, and Nikolay Pelov. Ontology driven software engineering for real life applications. *Proceedings of 3rd International Workshop on Semantic Web Enabled Software Engineering (SWESE 2007)*, 1:61–66, 2007.
- [268] Pepijn Visser and Trevor Bench-Capon. On the reusability of ontologies in knowledge-system design. *Proceedings of the Seventh Int. Workshop on Database and Expert Systems Applications*, 1:256 – 261, 1996.
- [269] Raphael Volz, Daniel Oberle, Steffen Staab, and Rudi Studer. Ontobroker and ontoedit adaptation. Technical report, Institute AIFB, University of Karlsruhe, 2003.
- [270] Denny Vrandečić. *Ontology Evaluation*. PhD thesis, Karlsruher Institute of Technology, 2010.
- [271] Denny Vrandečić and York Sure. How to design better ontology metrics. *The Semantic Web: Research and Applications, Springer-Berlag*, pages 311–325, 2007.
- [272] Yair Wand, Veda Storey, and Ron Weber. An ontological analysis of the relationship construct in conceptual modeling. *ACM Transactions on Database Systems*, 24(4):494–528, 2000.
- [273] David Wang and Bijan Parsia. Cropcircles: Topology sensitive visualization of owl class hierarchies. In *International Semantic Web Conference*, volume 4273 of *Lecture Notes in Computer Science*, pages 695–708. Springer, 2006.
- [274] Colin Ware. *Information Visualization: perception for design*. Elsevier, second edition, 2004.

- [275] Ron Weber. Conceptual modelling and ontology: Possibilities and pitfalls. *Journal of Database Management*, 14 (3):1–20, 2003.
- [276] Moritz Weiten. Ontostudio 3: Professional tool for knowledge architects. Technical report, Enterprise GmbH, 2009.
- [277] Moritz Weiten. *Semantic Knowledge Management*, chapter OntoSTUDIO: as a Ontology Engineering Environment, pages 51–61. Springer-Verlag, 2009.
- [278] Douglas West. *Introduction to Graph Theory*. Prentice Hall, 2001.
- [279] Jarke Van Wijk and Huub Van de Wetering. Cushion treemaps: Visualization of hierarchical information. *Proceedings of the IEEE Symposium on Information Visualization (InfoVis 99)*. *IEEE Computer Society*, pages 73–78, 1999.
- [280] Hong Yul Yang, Ewan Tempero, and Rebecca Berrigan. Detecting indirect coupling. *Proceedings of the 2005 Australian conference on Software Engineering*, 1:212–221, 2005.
- [281] Haining Yao, Anthony Mark Orme, and Letha Etzkorn. Cohesion metrics for ontology design and application. *Journal of Computer Science*, vol. 1:107–113, 2005.
- [282] Ma Yinglong, Jin Beihong, and Yulin Feng. Semantic oriented ontology cohesion metrics for ontology-based systems. *The Journal of Systems and Software, Elsevier*, 83(1):143–152, 2009.
- [283] Anna Zhdanova and Uwe Keller. Choosing an ontology language. *World Academy of Science, Engineering and Technology*, 4:47–50, 2005.
- [284] Dalu Zhang Zhe YANG and Chuan YE. Evaluation metrics for ontology complexity and evolution analysis. *IEEE International Conference on e-Business Engineering (ICEBE'06)*, 1:162–170, 2006.