

# Clasificación semántica de objetos urbanos en imágenes satelitales:

Una ontología para la interpretación de regiones urbanas basada en sus relaciones espaciales

Tesina de grado presentada  
por

Maximiliano Gabriel Cravero

al

Departamento de Ciencias de la Computación  
en cumplimiento parcial de los requerimientos  
para la obtención del grado de  
Licenciado en Ciencias de la Computación



Facultad de Ciencias Exactas, Ingeniería y Agrimensura.  
Universidad Nacional de Rosario.

Noviembre 2014

**Directora**

Dra. Cecilia H. Zanni-Merk  
cecilia.zanni-merk@unistra.fr

**Co-Directora**

Prof. Stella Marc-Zwecker  
stella@unistra.fr

ICube

(Laboratoire des sciences de  
de l'informatique et de l'imagerie)  
icube-bfo.unistra.fr

Equipo BFO

(Bioinformatique théorique,  
Fouille de données et Optimisation stochastique)

Universidad de Strasbourg

Pôle API

300 Bd S. Brant - BP 10413, 67412 Illkirch cedex, Francia

+33 (0)3 68 85 45 54

icube.unistra.fr

*Este trabajo está dedicado a mis padres y hermanos, quienes siempre me apoyaron durante toda mi vida, y particularmente, a lo largo de esta maravillosa carrera. Finalizar esta etapa tan importante para mí, me genera la necesidad y la obligación de agradecer a todas aquellas personas que siempre estuvieron conmigo cuando más las necesité. En especial a mis amigos y compañeros de carrera y estudio, los cuales sin el empuje de ellos todo hubiera sido más difícil. A Mariano, Raquel y Romina, que están y siempre estuvieron presentes brindándome su apoyo. A mi esposa Gisela, que siempre me alentó y estuvo en los buenos y malos momentos. Muchos de mis logros en esta carrera se los debo a ella. Finalmente, no quiero dejar de mencionar a mis directoras, Cecilia y Stella, quienes me ayudaron siempre que lo necesité y me guiaron por el camino correcto cuantas veces fue necesario.*

# Resumen

La información de la cobertura y el uso del suelo en objetos urbanos observados a diferentes escalas espaciales y temporales es de particular interés para planificadores urbanos y expertos geógrafos. Adquirir automáticamente esta información en entornos complejos, como por ejemplo áreas urbanas, es una tarea difícil que consume mucho tiempo si se utilizan fuentes de datos tradicionales (técnicas de agrimensura). La amplia disponibilidad de imágenes satelitales de alta resolución espacial representa una excelente oportunidad para identificar y caracterizar estos entornos. Mediante la descomposición de imágenes en objetos y regiones, es posible utilizar métodos que formalicen el conocimiento del dominio para automatizar el reconocimiento de objetos urbanos, es decir, calles, edificios, parques, etc.

Adquirir y representar conocimiento es muchas veces una tarea tediosa. Los múltiples pasos involucrados en la creación de una base de conocimiento pueden ser diferentes para un mismo dominio en estudio, lo que lleva a una heterogeneidad de la información. Sin embargo, si se hiciera correctamente, permitiría producir sistemas inteligentes basados en el uso del conocimiento adquirido. Las ontologías representan una alternativa interesante para construir una base de conocimiento. Su uso creciente en los últimos años ha permitido desarrollar diferentes lenguajes los cuales proveen diferentes facilidades para poder construir ontologías. El más utilizado, OWL 2.0, basado en lógica de descripción<sup>1</sup>, es apto para el razonamiento automático.

Este proyecto plantea el desarrollo de una ontología de objetos urbanos que ayude a los expertos en el proceso de reconocimiento de los mismos. Adicionalmente, para cumplir con los estándares utilizados por los expertos geógrafos, se utilizó la teoría del modelo RCC8<sup>2</sup> para representar relaciones espaciales entre regiones, lo cual añade mayor expresividad, mejorando el razonamiento sobre la ontología.

Para finalizar, y ante la ausencia de un software de procesamiento de imágenes que se ajuste a las necesidades del proyecto, se desarrolló una aplicación que permite que un usuario pueda delimitar las regiones de interés en una imagen satelital, simulando el comportamiento de un software especializado. Por medio de la interfaz gráfica, es posible ingresar información sobre las regiones dibujadas, la cual será incorporada a la ontología, para posteriormente, mediante el uso de un razonador lógico, inferir nuevos resultados.

---

<sup>1</sup>Fragmento decidible de la lógica de primer orden.

<sup>2</sup>Region Connection Calculus

# Índice general

Resumen . . . . .	III
<b>1. Introducción</b>	<b>1</b>
1.1. Introducción a las Ontologías . . . . .	2
1.1.1. Lenguajes de Ontologías . . . . .	2
1.2. Ontologías OWL . . . . .	3
1.2.1. Modelado con OWL 2 . . . . .	4
<b>2. Una ontología de objetos urbanos</b>	<b>10</b>
2.1. Diccionario de objetos urbanos . . . . .	10
2.2. Construyendo la jerarquía . . . . .	12
2.3. Algunas decisiones de diseño . . . . .	16
2.4. Modelado de relaciones y atributos . . . . .	17
2.5. Modelado de las clases . . . . .	23
2.5.1. Decisiones de diseño sobre la ontología . . . . .	30
<b>3. Relaciones espaciales aplicadas a la ontología</b>	<b>45</b>
3.1. El modelo RCC . . . . .	45
3.2. Implementación de relaciones RCC8 en OWL . . . . .	47
3.2.1. Reglas SWRL . . . . .	47
3.2.2. Primitivas CM8 . . . . .	48
3.2.3. Reificación de las relaciones RCC8 . . . . .	49
3.2.4. Desventajas de la reificación de relaciones . . . . .	50
3.2.5. Utilizando reglas SWRL para extender el modelo . . . . .	50
3.3. Relaciones RCC8 en la ontología de objetos urbanos . . . . .	52
<b>4. Una interfaz para la ontología de objetos urbanos</b>	<b>55</b>
4.1. Tecnologías utilizadas . . . . .	55
4.2. Desarrollo . . . . .	56
4.3. Ejemplo de uso . . . . .	65
<b>5. Conclusiones y trabajos futuros</b>	<b>75</b>
5.1. Conclusiones . . . . .	75
5.2. Trabajos futuros . . . . .	76

# Índice de figuras

1.1. Representación de individuos . . . . .	5
1.2. Representación de propiedades . . . . .	6
2.1. Una ficha típica del diccionario . . . . .	11
2.2. Tabla de principales relaciones para objetos urbanos . . . . .	14
2.3. Taxonomía de objetos urbanos . . . . .	15
2.4. Reificación de la relación de distancia . . . . .	20
2.5. Figura rectangular . . . . .	22
2.6. Figura cuadrada . . . . .	23
2.7. Figura sin forma reconocible . . . . .	23
2.8. Descripción del objeto <i>Pavillon</i> . . . . .	24
2.9. Equivalencia entre las clases <i>Pavillon</i> y <i>Chemin</i> . . . . .	28
2.10. Ejemplo de Inconsistencia . . . . .	29
2.11. Clase <i>Pavillon</i> en Protégé . . . . .	30
2.12. Clasificación del individuo “TEST_1” como miembro de <i>Curs d’Eau</i> . . . . .	32
2.13. Clasificación del individuo “TEST_1” como miembro de <i>Curs d’Eau</i> y de <i>Canal</i> . . . . .	33
2.14. Clasificación del individuo “TEST_2” como miembro de <i>Jardin</i> . . . . .	34
2.15. Clasificación de <i>Jardin</i> como subclase de <i>Cimetiere</i> . . . . .	35
2.16. Clasificación del individuo “TEST_2” como miembro de <i>Cimetiere</i> . . . . .	35
2.17. Ficha para el objeto <i>Bassin Artificiel</i> . . . . .	37
2.18. Propuesta para objetos identificables en múltiples resoluciones . . . . .	38
2.19. Clase <i>Bassin</i> definida en bloques . . . . .	39
2.20. Modificación de la taxonomía de <i>Surface Vegetal</i> . . . . .	40
2.21. Descripción del objeto <i>Parc</i> . . . . .	41
2.22. Separación de la clase <i>Parc</i> en tres subclases . . . . .	42
2.23. Taxonomía final para objetos simples . . . . .	43
2.24. Taxonomía final para objetos compuestos . . . . .	44
3.1. Representación gráfica de las relaciones RCC8 . . . . .	46
3.2. Reificación de relaciones espaciales . . . . .	49
3.3. Látice de relaciones espaciales RCC8 con primitivas CM8 . . . . .	51
3.4. Taxonomía parcial de la unión de las ontologías . . . . .	53

4.1. Paneles de la aplicación . . . . .	57
4.2. Dibujado de polígonos . . . . .	57
4.3. Formulario de características de un polígono . . . . .	60
4.4. Ventana de importación de segmentación . . . . .	63
4.5. Diagrama de flujo de una ejecución típica . . . . .	64
4.6. Aplicación cargada con imagen de resolución HR2 . . . . .	65
4.7. Segmentación de la imagen de resolución HR2 . . . . .	66
4.8. Formulario de características del polígono P4 . . . . .	67
4.9. Tipo inferido para el polígono P4 . . . . .	67
4.10. Imagen de resolución THR1 cargada en la nueva capa . . . . .	70
4.11. Segmentación de la imagen de resolución THR1 . . . . .	71
4.12. Formulario de características del polígono P10 . . . . .	71
4.13. Tipo inferido para el polígono P10 . . . . .	72

# Capítulo 1

## Introducción

La interpretación de imágenes es una tarea difícil que puede ser definida como la extracción de la semántica de una imagen. Consiste en la obtención de información espacial útil y temática sobre los objetos en consideración utilizando el conocimiento y la experiencia humana [2]. En este dominio, se observan diferencias entre la interpretación “visual” y la interpretación semántica de los píxeles producto de las diferencias entre los niveles de abstracción.

La semántica depende del conocimiento del dominio y del contexto, pero no siempre se encuentra explícitamente contenida en la imagen. Este problema es llamado *semantic gap*<sup>1</sup> y se define como la falta de coherencia entre la información de bajo nivel (aquella extraída automáticamente de las imágenes mediante software de reconocimiento) y la información de alto nivel (analizada por expertos urbanistas y geógrafos). Para reducir el “gap” semántico, se han desarrollado métodos de análisis de imágenes basados en objetos que hacen uso del conocimiento de dominio ([3], [4]). Estos métodos involucran la segmentación de imágenes en regiones homogéneas y la caracterización espectral (firma espectral, índice de Morton), espacial (adyacencia, inclusión, composición) y topológica (forma) de objetos. La clasificación basada en regiones logra obtener mejores resultados que la clasificación basada en píxeles [5]. Tan sólo una minoría de trabajos se han enfocado en el conocimiento del dominio para la clasificación de objetos urbanos [8], y el principal problema que han tenido ha sido la formalización del conocimiento y su explotación basado en ontologías.

Construir una ontología de objetos urbanos es una tarea difícil debido a que la mayoría de las veces, el conocimiento es implícito y se encuentra en manos de los expertos en el dominio. Este trabajo utiliza el mismo enfoque que [1] al usar ontologías, pero, a diferencia de éste, el conocimiento obtenido para desarrollar la ontología será extraído de un diccionario existente de objetos geográficos desarrollado por expertos, y la formalización y normalización de este conocimiento será modelizado con un lenguaje de ontologías basado en lógica que permitirá razonar sobre el modelo con el objetivo de verificar la consistencia e inferir nueva información sobre el mismo.

---

<sup>1</sup>Traducido en español como brecha semántica

## 1.1. Introducción a las Ontologías

Dentro de las variadas definiciones que existen, la más aceptada y citada en el ámbito de las ciencias de la computación es la descrita por Thomas Gruber ([6, 7]): “Una ontología puede definirse como una especificación explícita de una conceptualización”<sup>2</sup>. Define los términos usados para describir y representar un área de conocimiento. Son utilizadas por personas, bases de datos y agentes que necesitan compartir información sobre un dominio en estudio<sup>3</sup>.

Una ontología necesita especificar los siguientes conceptos:

- **Clases** en el dominio de interés.
- **Relaciones** existentes entre las clases.
- **Propiedades** o **Atributos** que las clases pueden tener.

Los lenguajes utilizados para especificar ontologías permiten una abstracción de las estructuras de datos y de las implementaciones. En la práctica, los lenguajes que se utilizan tienen un poder de expresividad similar al de la lógica de primer orden. Por esta razón, se dice que las ontologías se encuentran a nivel “semántico”, mientras que los esquemas de base de datos se encuentran a nivel “lógico” o “físico”. Debido a su independencia de los modelos de bajo nivel, las ontologías son utilizadas para la integración de bases de datos heterogéneas, permitiendo la interoperabilidad entre sistemas dispares, y para la especificación de interfaces para servicios independientes basados en conocimiento.

### 1.1.1. Lenguajes de Ontologías

Existen diferentes lenguajes para especificar ontologías los cuales proveen diferentes facilidades. Por ejemplo, entre los menos expresivos, podemos encontrar:

- Para modelado conceptual: Diagramas Entidad Relación, Diagramas UML (Unified Modeling Language).
- El Resource Description Framework (RDF) y su extensión RDF Esquema (RDF-S).

Entre los más expresivos, basados en lógica se encuentran:

- Lógica de descripción y el estándar OWL (Web Ontology Language).
- Datalog y lenguajes basados en reglas.
- Gráfos conceptuales.

---

<sup>2</sup>Una conceptualización se define como una vista simplificada, abstracta, de un dominio a representar.

<sup>3</sup>Dominio se entiende como un área temática específica o área de conocimiento, como puede ser medicina, fabricación de herramientas, bienes raíces, reparación de automóviles, gestión financiera, etc

- Lógica de predicados.

El Resource Description Framework [9], como su nombre lo indica, es un framework utilizado para describir información sobre recursos en la World Wide Web. Los recursos pueden ser cualquier cosa, incluyendo documentos, personas, objetos físicos, y conceptos abstractos. Fue pensado para situaciones en las que se necesita que la información en la Web sea procesada por aplicaciones en lugar de tan sólo ser presentadas a los humanos.

RDF Schema [10] es una extensión semántica de RDF. Provee mecanismos para describir grupos de recursos relacionados y relaciones entre estos recursos. Dichos recursos son utilizados para determinar las características de otros recursos tales como el dominio y el rango de propiedades. Con RDF-S, pueden definirse clases que puedan tener múltiples subclases y super clases, y pueden definirse propiedades, las cuales pueden tener subpropiedades, dominios y rangos. En este sentido, RDF-S es un simple lenguaje de ontologías.

Entre las deficiencias del lenguaje, se destaca la incapacidad para representar disjunción de clases. Esto significa que, por ejemplo, no es posible especificar que dos clases como “Persona” y “Auto” no comparten características. Otro ejemplo de limitación del lenguaje es no poder especificar que un cuarteto de cuerdas tiene exactamente 4 músicos como miembros.

Construido sobre RDF, **OWL**[11] fue diseñado para compensar la falta de un lenguaje que permita formalmente describir la semántica de las clases y propiedades presentes en documentos web.

Este nuevo estándar posee un conjunto de operadores más extenso que sus predecesores como intersección, unión y negación, más vocabulario para describir propiedades y clases, relaciones entre clases (por ej. disjunción), cardinalidad (por ej. exactamente 4), igualdad, tipos de datos de propiedades más ricos, características de las propiedades (por ej. simetría) y clases enumeradas. Está basado en Lógica de Descripción, el cual es un fragmento decidible de la lógica de primer orden, y por tanto apto para razonamiento automático, lo que permite computar automáticamente jerarquía de clases y comprobar inconsistencias en una ontología.

Para este trabajo, la elección de OWL como lenguaje de ontologías se debe, entre otras cosas, a las nuevas facilidades que adiciona para crear modelos lógicos consistentes. Esto permite la utilización de razonadores lógicos para comprobar consistencia e inferir nueva información que no está literalmente presente en el modelo, pero sí implicada por la semántica.

## 1.2. Ontologías OWL

El lenguaje de ontologías OWL fue diseñado para formular, intercambiar y razonar sobre el conocimiento de un dominio de interés. Antes de avanzar con este documento, es necesario introducir algunas nociones fundamentales sobre cómo el conocimiento es representado en OWL:

- **Axiomas:** Las afirmaciones básicas que una ontología OWL expresa.

- **Entidades:** Elementos utilizados para referirse a objetos en el dominio.
- **Expresiones:** Combinaciones de entidades utilizadas para formar descripciones complejas a partir de más simples.

Si consideramos que el conocimiento está conformado por piezas elementales, usualmente referidas como afirmaciones o proposiciones, formular conocimiento se vuelve un poco más simple. Sentencias como “Está lloviendo”, “María es mujer” o “Juan y María están casados”, son típicos ejemplos de tales proposiciones. De hecho, toda ontología OWL es esencialmente una colección de esas “piezas básicas de conocimiento”. Las proposiciones o afirmaciones que se hacen en una ontología son llamados **axiomas** en OWL, y la ontología afirma que los axiomas son verdaderos.

Las proposiciones generalmente se refieren a objetos del mundo en representación y los describe poniéndolos en categorías o diciendo algo sobre su relación. Los átomos que constituyen estas afirmaciones, ya sean objetos (“Juan”, “María”), categorías (“mujer”) o relaciones (“casados”), son **entidades**. En OWL, se denota a los objetos como *Individuos* (Individuals), a las categorías como *Clases* (Classes) y a las relaciones como *Propiedades* (Properties).

Las entidades pueden combinarse en **expresiones** utilizando *constructores*. Como ejemplo básico, las clases “Mujer” y “Profesor” pueden combinarse en conjunción para describir la clase de profesores que son mujeres. Este ejemplo describe lo que se conoce como *OWL Class Expression* o expresión formada por clases OWL, que a su vez puede ser usada en proposiciones o en otras expresiones.

Pese a las grandes ventajas que OWL posee para modelar ontologías, el lenguaje tiene todavía algunas carencias. Problemas relacionados mayormente con la expresividad y la sintaxis necesitan ser corregidos para contar con un lenguaje de ontologías totalmente confiable. OWL 2 ([12],[13]) se desarrolló con esta intención e introdujo nuevas características, algunas de estas son *syntactic sugar*<sup>4</sup>, (por ejemplo, unión disjunta de clases) mientras que otras ofrecen más expresividad:

- Tipos de datos más ricos, rango de datos.
- Restricciones de cardinalidad calificadas<sup>5</sup>.
- Propiedades asimétricas, reflexivas y disjuntas.

### 1.2.1. Modelado con OWL 2

Luego de estas introducciones, es momento de entrar en detalle sobre el modelado de ontologías con OWL 2. Pero antes de continuar, es fundamental tener conocimiento que

---

<sup>4</sup>Término utilizado para referirse a los añadidos a la sintaxis de un lenguaje de programación que no afectan a su funcionalidad, pero que facilitan expresar algunas construcciones de una forma más clara o concisa, o en un estilo alternativo.

<sup>5</sup>Las restricciones de cardinalidad se utilizan para limitar el número de relaciones en las que un individuo puede participar mediante una determinada propiedad, independientemente del tipo del individuo. Cuando estas restricciones se limitan sobre un tipo en particular se denominan calificadas.

OWL trabaja con lo que se conoce como **hipótesis del mundo abierto**. La cual establece que si un hecho no está presente en la base de conocimiento, no puede darse por falso, ya que podría no ser conocido por el observador. Para que un hecho sea falso tiene que estar explícitamente especificado. Por otra parte, la **hipótesis del mundo cerrado**, establece que un hecho es falso si no se conoce si es verdadero. A lo largo de este documento se hará hincapié en esta hipótesis varias veces ya que es una parte fundamental del modelado de ontologías OWL.

A continuación se hará una breve introducción de las características que los componentes de OWL 2 ofrecen para modelar ontologías<sup>6</sup>.

## Individuos

Los individuos representan objetos en un dominio. Muchas veces es útil expresar la igualdad o la diferencia entre dos o más individuos de un dominio. En OWL los nombres no se asumen como únicos, por lo tanto no se supone que nombres diferentes representen entidades diferentes. Para subsanar esto, el lenguaje provee mecanismos que permiten expresar la igualdad o diferencia de entidades. Esto permite por ejemplo, que si dos individuos se establecen como iguales, mediante un razonador lógico toda la información inferida en un individuo también se inferirá en el otro.

La figura 1.1 muestra la representación de individuos en algún dominio.



Figura 1.1: Representación de individuos

## Propiedades

Las propiedades o roles se clasifican en *Object Properties* y en *Datatype Properties*<sup>7</sup>, dependiendo de si relacionan objetos con objetos u objetos con algún valor de un tipo de

---

<sup>6</sup>Para profundizar en el tema y obtener ejemplos de uso, se recomienda la lectura de [12].

<sup>7</sup>Nota: A lo largo de este documento nos referiremos a estas propiedades como *Object Properties* y *Datatype Properties*, ya que su traducción al español es un poco incómoda para la lectura.

datos en particular. Sea cual sea el tipo de la propiedad, estas pueden ser vistas como relaciones binarias, ya que siempre relacionan dos elementos.

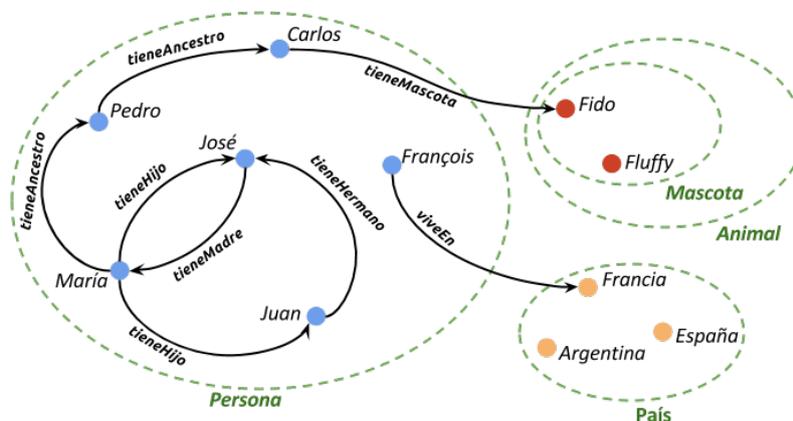


Figura 1.2: Representación de propiedades

Entre las prestaciones que ofrecen las propiedades se destacan:

- Jerarquía de propiedades. Al igual que las clases, las propiedades pueden estructurarse en jerarquías. Al utilizar este tipo de relación de subsunción, es posible mediante el uso de razonadores inferir nuevas relaciones entre objetos o entre objetos y valores de datos.
- Afirmación negativa. En oportunidades es necesario expresar que dos individuos (o un individuo y un valor) no se encuentran conectados por medio de una determinada propiedad. Esto es lo que se denomina una *afirmación negativa de una propiedad* y resulta especialmente útil en OWL ya que de acuerdo a la hipótesis del mundo abierto, cualquier cosa es posible a menos que se indique lo contrario.
- Restricción de dominio y rango. Restricciones de este tipo son útiles, por ejemplo, para inferir membresía de los participantes de una propiedad a las clases del dominio o del rango de la misma. Notar que este tipo de restricciones no conforma una limitación al conocimiento, sino que permite a los razonadores inferir conocimiento nuevo. Esto significa, por ejemplo, que dada una propiedad  $P$  que relaciona objetos de tipo  $A$  con objetos de tipo  $B$ , y un axioma que exprese una relación entre un individuo de clase  $A$  con otro de una tercera clase  $C$ , sería suficiente para que un razonador infiera que el individuo de clase  $C$  también pertenece a la clase  $A$ , lo cual no sería incorrecto. Para que una inconsistencia tenga lugar,  $A$  y  $C$  deberían haberse definido como disjuntas desde un principio, ya que de lo contrario, la hipótesis del mundo abierto no garantiza que esto ocurra.

En el ejemplo de la figura 1.2, podríamos limitar el dominio de la propiedad **viveEn** a la clase *Persona* y el rango a la clase *País*. De esta forma sabríamos que cada vez que un país se relaciona con un individuo mediante la propiedad **viveEn**, este será de tipo *Persona* y no de tipo *Mascota*.

- Dentro de la descripción de una propiedad, además de especificar rango y dominio, es posible precisar a qué propiedades es equivalente, con cuáles propiedades es disjunta y de qué propiedades es inversa. Continuando con el ejemplo de la figura 1.2, la propiedad **tieneMadre** puede ser considerada como la propiedad inversa de **tieneHijo** y viceversa.
- Caracterización de propiedades. Una propiedad puede ser calificada como:
  - *Simétrica*. Si  $P$  es una propiedad simétrica que relaciona a un individuo  $x$  con un individuo  $y$ , entonces  $P$  también relaciona a  $y$  con  $x$ . La propiedad **tieneHermano** de la figura 1.2, que afirma que *Juan* tiene como hermano a *José*, es una propiedad simétrica. Por lo tanto, si esta propiedad se definiera como tal, un razonador podría inferir que *José* también tiene como hermano a *Juan*.
  - *Asimétrica*. Si  $P$  es una propiedad asimétrica que relaciona a un individuo  $x$  con un individuo  $y$ , entonces  $P$  no puede relacionar a  $y$  con  $x$ . Las propiedades **tieneHijo** y **tieneMadre** de la figura 1.2 son ejemplos de propiedades asimétricas.
  - *Reflexiva*. Una propiedad  $P$  se dice reflexiva si relaciona a un individuo consigo mismo.
  - *Irreflexiva*. Una propiedad  $P$  se dice irreflexiva si relaciona a un individuo con otros individuos diferentes de él. Todas las propiedades representadas en la figura 1.2 son irreflexivas dado que ninguna puede relacionar a un individuo consigo mismo.
  - *Funcional*. Si  $P$  es una propiedad funcional, un individuo  $x$  se relaciona mediante  $P$  con a lo sumo un único individuo  $y$ . La propiedad **tieneMadre** de la figura 1.2 es una propiedad funcional dado que una persona no puede tener más de una madre biológica.
  - *Funcional inversa*. Si  $P$  es una propiedad funcional inversa, entonces la inversa de  $P$  es funcional. Si consideramos a la propiedad **tieneMadre** como la inversa de la propiedad **tieneHijo**, dado que **tieneMadre** es funcional, podemos decir que la propiedad **tieneHijo** es funcional inversa.
  - *Transitiva*. Si  $P$  es una propiedad transitiva que relaciona a un individuo  $x$  con un individuo  $y$ , y a  $y$  con un individuo  $z$ , entonces  $P$  también relaciona a  $x$  con  $z$ . La propiedad **tieneAncestro** es un ejemplo de propiedad transitiva. Si se definiera como tal, un razonador podría inferir que *Carlos* es ancestro de *María* conociendo que *Carlos* es ancestro de *Pedro* y que *Pedro* es ancestro de *María*.

## Clases e Instancias

En este contexto, un individuo que es miembro de una clase se dice que es una instancia de esa clase. Un individuo que es miembro de más de una clase es una instancia de cada una de esas clases.

Las clases OWL son interpretadas como conjuntos que contienen individuos (instancias de esa clase), como en el ejemplo de la figura 1.2, los cuales son elementos del dominio de interés. Se describen utilizando una descripción formal, lógica-matemática, que precisa los requerimientos que un individuo debe cumplir para ser miembro.

Pueden organizarse en una jerarquía del tipo superclase-subclase, conocida también como taxonomía. La relación del tipo superclase-subclase está basada en el principio de herencia. Una subclase hereda la definición completa de sus superclases (o clases padre) y agrega alguna información extra que la hace más específica que su padre. Podemos decir entonces que si  $A$  es subclase de  $B$ , si un individuo  $a \in A$ , entonces  $a \in B$ . Esta relación también es llamada comúnmente *is a* (en español *es un*), en alusión a que todo miembro de una clase *es un* miembro de la clase padre de esta, pero no necesariamente al revés. Por ejemplo, en la figura 1.2 tenemos a la clase *Animal* representada como superclase de la clase *Mascota*. Como consecuencia, podemos asegurar entonces que *Fluffy*, además de ser una mascota, también es un animal. De la misma forma, también podemos afirmar que no todo animal será considerado como mascota.

Además de la relación de subclase, en OWL tenemos la relación de *equivalencia*, para especificar que dos clases se consideran semánticamente equivalentes, y la relación de *disjunción* para expresar justamente lo contrario. Información como esta es parte del conocimiento de trasfondo y por tanto debe estar explícito para que un sistema de razonadores pueda hacer uso efectivo del mismo.

En los lenguajes basados en lógica de descripción como OWL, las clases pueden ser **Primitivas** o **Definidas**. Las clases **Primitivas** sólo tienen condiciones necesarias, del tipo que definen a una superclase. Las clases **Definidas** tienen condiciones necesarias y suficientes, como las que se usan para definir las clases equivalentes.

Para definir una clase, ya sea **Primitiva** o **Definida**, se deben crear *restricciones*. Esto se logra combinando clases mediante constructores lógicos **and**, **or**, **not** (análogos a los operadores de intersección, unión y complemento de la teoría de conjuntos), mediante **cuantificadores existenciales y universales** sobre las propiedades, utilizando **restricciones de cardinalidad** sobre las propiedades o combinando todas las anteriores de manera adecuada.

Una restricción conforma lo que se denomina una *Clase Anónima*. A diferencia de las clases comunes, una clase anónima no tiene nombre y está compuesta por aquellos individuos que cumplen con esa restricción. Por ejemplo, cuando una clase se define de manera equivalente con algunas restricciones, implícitamente se la está declarando como equivalente a esas clases anónimas que la definen.

Una restricción de propiedad formada por una cuantificación existencial define la clase de individuos que están conectados por medio de una propiedad en particular con **al menos** un individuo miembro de alguna otra clase. Este tipo de restricciones es útil para capturar conocimiento incompleto. Por ejemplo, si sabemos que un individuo  $p \in Padre$ ,

podemos estar seguros de que  $p$  tiene al menos un hijo, aunque no sabemos específicamente cuál es.

Palabras como “alguno”, “uno” y “al menos”, son algunos ejemplos de indicadores de cuantificadores existenciales en lenguaje natural.

Una propiedad restringida por una cuantificación universal describe la clase formada por individuos que están relacionados mediante esa propiedad **únicamente** con individuos miembros de alguna otra clase. Es particularmente útil cuando se quiere especificar una clase compuesta únicamente por miembros de un tipo. Sin embargo, cabe mencionar que los cuantificadores universales no garantizan existencia. Es decir, no basta con que al menos un individuo de la clase cumpla con la restricción impuesta por la propiedad, en su lugar lo que establece, es que si existe algún individuo que cumple con la restricción entonces o bien es el único en ese conjunto, o bien todos los elementos en él la cumplen. A su vez, las restricciones formadas por cuantificadores universales también describen los elementos que no se relacionan con una propiedad dada. Esto se debe a la relación de dualidad que existe con el cuantificador existencial,  $\forall x P(x) \Leftrightarrow \neg \exists x \neg P(x)$ .

Como regla práctica, cuando se quiere traducir una frase en lenguaje natural a un axioma lógico, las cuantificaciones existenciales ocurren más a menudo. Sin embargo, palabras como “sólo”, “exclusivamente” y “únicamente”, son algunos ejemplos de indicadores de cuantificadores universales.

Además de cuantificadores es posible utilizar restricciones de cardinalidad para especificar la cantidad de individuos involucrados en una relación en particular. Utilizando los constructores **min**, **max** y **exactly**, podemos formar restricciones que permitan construir clases con un número mínimo, máximo o exacto de miembros relacionados con una propiedad en particular.

Otra manera de describir una clase es por extensión. Al igual que en la teoría de conjuntos, OWL provee mecanismos para definir clases enumerando a sus miembros participantes.

Introducidos estos conceptos básicos para el modelado con OWL 2, a continuación se comenzará con el desarrollo de la ontología de objetos urbanos.

## Capítulo 2

# Una ontología de objetos urbanos

Para diseñar ontologías existen diferentes herramientas que facilitan el proceso de programación de la misma. Una de estas es **Protégé** [16], la cual es un editor open source que soporta la creación, visualización y manipulación de ontologías en varios formatos de representación. Permite crear ontologías en lenguaje OWL haciendo mucho más fácil escribir código OWL utilizando una sintaxis simplificada desde su interfaz de usuario. Incluye razonadores lógicos embebidos con el objetivo de derivar consecuencias lógicas y comprobar consistencia. Los más comúnmente usados son Fact++ [17], Pellet [18] y Hermit [19]. La versión de **Protégé** a utilizar será la 4.3.0.

En las siguientes secciones se presentará la ontología de objetos urbanos y se mostrará el modelado de algunas características de determinados objetos utilizando la sintaxis de **Protégé**.

### 2.1. Diccionario de objetos urbanos

Las ontologías son útiles, entre otras cosas, para describir, representar y compartir información sobre el conocimiento de un dominio en estudio. En este caso, el dominio en estudio se compone de objetos urbanos de la ciudad de *Strasbourg*. El conocimiento de estos objetos se encuentra contenido en un diccionario de objetos urbanos construido especialmente por expertos en la materia<sup>1</sup>. Este documento, similar a un diccionario regular, define cada objeto geográfico con descripciones textuales, gráficos de ejemplo, algunos valores representativos y relaciones con otros objetos. Los mismos comprenden regiones, parcelas, cuerpos de agua, caminos, edificios, etc. Son objetos espaciales de una cierta mínima escala, típicamente complejos y muchas veces compuestos por otros.

La figura 2.1 ilustra cómo se compone una “ficha” típica del diccionario de objetos. Cada una de estas fichas está compuesta de diferentes piezas que contienen cierta información, como ser: el nombre del objeto, el tipo de objeto elemental, las resoluciones espaciales a las que es distinguible el objeto, una definición textual, algunas imágenes de ejemplo a

---

<sup>1</sup>Nota: Pese a que los objetos puedan ser comunes a los de otras ciudades, es muy probable que las descripciones de estos objetos varíen y que por lo tanto esta ontología no sea útil para caracterizar otros entornos urbanos.

diferentes resoluciones, una tabla de relaciones con otros objetos y una tabla de atributos con valores. Toda esta información representa para la ontología el conocimiento explícito que se tiene de un objeto en particular.

A continuación se realizará una breve descripción de las piezas más relevantes de una ficha.

**Nº de Ficha: Nombre del Objeto**

**A. Identificación del objeto**

Tipo	Nombre del Objeto	Tipo de Objeto Elemental	Resoluciones en las cuales el objeto puede ser identificado
 Polygone			

**B. Descripción del objeto en el mundo real**

- **B.1** - Definición textual del objeto en el mundo real
- **B.2** - Representación gráfica del objeto en cada resolución en la que puede ser identificado

**C. Descripción del objeto en la imagen**

- **C.1** - Naturaleza del objeto
- **C.2** - Definición textual del objeto en la imagen
- **C.3** - Tabla de las principales relaciones que el objeto mantiene con otros según la resolución
- **C.4** - Tabla de atributos del objeto diferenciados por resolución

Figura 2.1: Una ficha típica del diccionario

## Resolución espacial

Constituye uno de los tipos de resolución que caracterizan a una imagen satelital. Representa al objeto más pequeño que puede distinguirse en una imagen y está determinada por el tamaño de pixel, el cual es medido en metros sobre el terreno. Otros tipos de resoluciones utilizadas en el estudio de imágenes satelitales son: la resolución espectral, que consiste en el número de canales espectrales (y su ancho de banda) que es capaz de captar un sensor; y la resolución temporal, la cual representa la frecuencia de paso del satélite por un mismo punto de la superficie terrestre.

El diccionario de objetos urbanos utiliza la resolución espacial como parámetro en la definición de objetos. Establece tres niveles de resolución espacial, los cuales implican diferentes propiedades y relaciones inclusive para un mismo objeto. Los niveles identificados por THR1, HR1 y HR2, representan resoluciones espaciales en el rango de 0,5m a 5m, de 5m a 15m y de 15m a 30m respectivamente. Esto significa que para una imagen con resolución espacial de 2m, un pixel estará representando una superficie de 2m x 2m.

## Objeto elemental

El tipo de objeto elemental hace referencia a un objeto no descrito en el diccionario, que representa a un tipo objeto genérico, común a muchos otros.

## Relaciones

Según el diccionario, en general un objeto se relaciona con otros por medio de las siguientes relaciones:

- **Adyacencia:** Refiere a todos los objetos que son vecinos de un objeto en particular.
- **Inclusión:** Es una relación mantenida entre un objeto cualquiera y su objeto contenedor.
- **Composición:** Un objeto mantiene una relación de composición con los objetos que lo conforman.
- **Distancia:** Especifica la distancia que mantiene un objeto con respecto a otro particular.

## Atributos

Además de sus relaciones, un objeto puede ser caracterizado por los siguientes atributos: *forma, superficie, longitud, amplitud, textura, densidad, elongación, alineación y discontinuidad*

Presentado formalmente el diccionario de objetos urbanos, comenzaremos con el diseño de la ontología. Como primer paso se realizará un análisis “superficial” de la información del mismo, con el objetivo de capturar el conocimiento que permitirá construir una primera versión de la jerarquía de clases.

## 2.2. Construyendo la jerarquía

La jerarquía de clases estructura las clases en una taxonomía<sup>2</sup> de tipo superclase-subclase de acuerdo a las definiciones y propiedades que les fueron impuestas de antemano. De esta forma las clases con conceptos más detallados serán subclases de clases con conceptos más generales.

Al estudiar el contenido del diccionario, lo primero que podemos notar es que todos los objetos tienen exactamente un tipo de objeto elemental. Con esta información podríamos crear grupos de objetos organizados según su tipo elemental e implícitamente estaríamos

---

<sup>2</sup>Definida como la ciencia de identificar y nombrar objetos ordenándolos en una clasificación. Habitualmente utilizado para designar a la taxonomía biológica, la cual hace referencia al ordenamiento de la diversidad de especies.

Categoria de objeto	Objeto elemental	Ficha
<i>Objetos Simples</i>	<i>Bâtiment</i>	Ficha 1 : Bâtiment de l’habitat individuel (Pavillon) Ficha 2 : Bâtiment de l’habitat collectif discontinu Ficha 3 : Bâtiment de l’activité Ficha 4 : Bâtiment isolé, de l’habitat rural Ficha 5 : Autre Bâtiment
	<i>Végétation</i>	Ficha 6 : Arbre Ficha 7 : Surface herbeuse (Pelouse) Ficha 8 : Jardin (ouvrier ou familial) Ficha 9 : Parcelle agricole Ficha 25 : Surface boisée
	<i>Tronçon ou voie</i>	Ficha 10 : Route à grande vitesse Ficha 11 : Chemin de fer Ficha 12 : Autre route Ficha 13 : Chemin, piste
	<i>Eau</i>	Ficha 14 : Cours d’eau Ficha 15 : Canal Ficha 16 : Bassin artificiel Ficha 17 : Etendue d’eau
	<i>Equipement divers</i>	Ficha 18 : Terrain de sport Ficha 19 : Piscine extérieure Ficha 20 : Cimetière Ficha 21 : Sol nu Ficha 22 : Aire de stationnement ou parking Ficha 23 : Place Ficha 24 : Ilot de circulation
<i>Objetos Compuestos</i>	<i>Surface végétale</i>	Ficha 26 : Groupe d’arbres Ficha 27 : Alignement d’arbres Ficha 28 : Parc
	<i>Surface artificialisée</i>	Ficha 29 : Zone industrielle, commerciale, tertiaire Ficha 30 : TU <sup>a</sup> continu Ficha 31 : TU discontinu de type individuel Ficha 32 : TU discontinu de type collectif Ficha 33 : TU discontinu Ficha 34 : Zone sportive ou de loisir Ficha 35 : Zone d’extraction Ficha 36 : Zone aéroportuaire

Tabla 2.1: Objetos geográficos urbanos

<sup>a</sup>Corresponde a las siglas de *Tissu Urbain* o Tejido Urbano. Este concepto, normalmente asociado con el diseño urbano, combina espacios abiertos y construidos con la actividad humana en una ciudad.

diciendo que un objeto elemental es una especie de objeto más general que representa a otros.

Mediante un análisis más profundo del diccionario, podemos apreciar que algunos objetos tienen presente la relación *Composition* en su tabla de relaciones, como es el caso del objeto *Parc* (figura 2.2a), mientras que otros carecen de esta relación, como es el caso del objeto *Pavillon* (figura 2.2b). Esto nos lleva a realizar otra clasificación, creando dos grandes conjuntos: uno denominado **objetos compuestos**, que está formado por aquellos elementos que resultan de la organización o estructuración espacial específica de los objetos que lo componen; y otro, denominado **objetos simples**, formado por aquellos objetos que no pueden ser descompuestos. Dado que el conjunto de objetos compuestos y el conjunto de objetos simples no comparten objetos elementales, es decir que no existen objetos en ambos conjuntos que tengan el mismo tipo de objeto elemental, podemos decir que ambos conjuntos son disjuntos. El resultado de este análisis se encuentra representado en la tabla 2.1.

<p><b>C.2. Définition textuelle</b></p> <p>L'objet « parc » identifiable à THR1 dans l'image se rattache à la définition de jardin public</p> <p><b>C.3. Principales relations</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="border: 2px solid red;">Composition</td> <td style="border: 2px solid red;">Arbre, pelouse, chemin, sol nu, étendue d'eau, bâtiment isolé, groupe d'arbres</td> </tr> <tr> <td>Adjacence</td> <td>autre route</td> </tr> <tr> <td>Alignement</td> <td>non</td> </tr> <tr> <td>Distance = relation de voisinage</td> <td>non</td> </tr> <tr> <td>Inclusion</td> <td>à un TU discontinu (HR1/HR2)</td> </tr> </table> <p style="text-align: center;">(a) <i>Parc</i></p>	Composition	Arbre, pelouse, chemin, sol nu, étendue d'eau, bâtiment isolé, groupe d'arbres	Adjacence	autre route	Alignement	non	Distance = relation de voisinage	non	Inclusion	à un TU discontinu (HR1/HR2)	<p><b>C.2. Définition textuelle</b></p> <p>L'objet « pavillon » ou « maison individuelle » est représenté graphiquement par un polygone dont la surface correspond à l'emprise au sol du bâtiment.</p> <p><b>C.3. Principales relations</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td>Adjacence</td> <td>Objets de type « végétation » Objets de type « autre route »</td> </tr> <tr> <td>Alignement</td> <td>oui</td> </tr> <tr> <td>Distance entre barycentre</td> <td>2 à 15m Faible (appartenance à une cité ouvrière, individuel dense) Moyenne (individuel moyennement dense) Elevée (individuel lâche)</td> </tr> <tr> <td>Distance à la route</td> <td>4 à 12m Faible / Moyen /Elevée</td> </tr> <tr> <td>Inclusions</td> <td>TU discontinu de type individuel HR1 (<a href="#">fiche 31</a>) et/ou HR2 (<a href="#">fiche 33</a>)</td> </tr> </table> <p style="text-align: center;">(b) <i>Pavillon</i></p>	Adjacence	Objets de type « végétation » Objets de type « autre route »	Alignement	oui	Distance entre barycentre	2 à 15m Faible (appartenance à une cité ouvrière, individuel dense) Moyenne (individuel moyennement dense) Elevée (individuel lâche)	Distance à la route	4 à 12m Faible / Moyen /Elevée	Inclusions	TU discontinu de type individuel HR1 ( <a href="#">fiche 31</a> ) et/ou HR2 ( <a href="#">fiche 33</a> )
Composition	Arbre, pelouse, chemin, sol nu, étendue d'eau, bâtiment isolé, groupe d'arbres																				
Adjacence	autre route																				
Alignement	non																				
Distance = relation de voisinage	non																				
Inclusion	à un TU discontinu (HR1/HR2)																				
Adjacence	Objets de type « végétation » Objets de type « autre route »																				
Alignement	oui																				
Distance entre barycentre	2 à 15m Faible (appartenance à une cité ouvrière, individuel dense) Moyenne (individuel moyennement dense) Elevée (individuel lâche)																				
Distance à la route	4 à 12m Faible / Moyen /Elevée																				
Inclusions	TU discontinu de type individuel HR1 ( <a href="#">fiche 31</a> ) et/ou HR2 ( <a href="#">fiche 33</a> )																				

Figura 2.2: Tabla de principales relaciones para objetos urbanos

Si consideramos a cada elemento de la tabla como una clase, la relación *es un* nos permite construir la taxonomía haciendo la siguiente interpretación sobre la tabla: todo elemento de una columna en particular *es un* elemento de la columna inmediatamente a la izquierda de esta. Para finalizar, se creará una clase llamada *Objet.Geo*, como la unión disjunta entre las clases de objetos compuestos y objetos simples, para expresar que si un elemento pertenece a esta categoría no puede ser compuesto y simple al mismo tiempo.

De todo este análisis, se desprende la siguiente jerarquía de objetos urbanos representada en la figura 2.3.

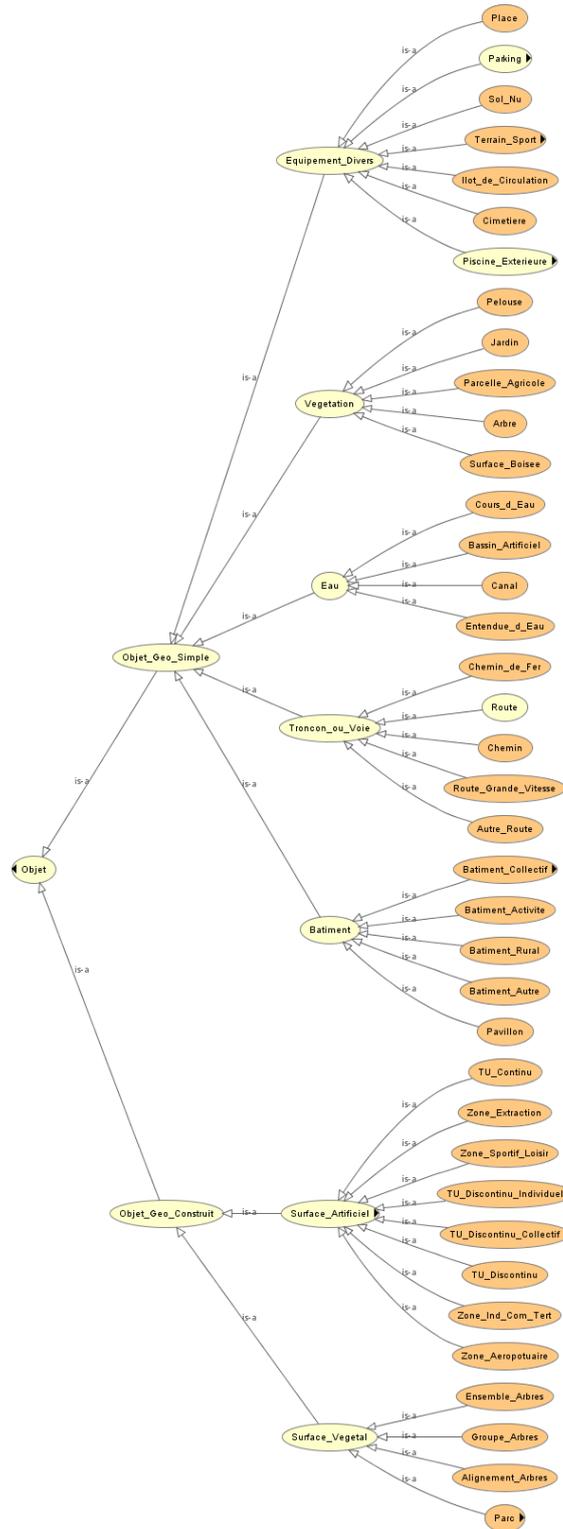


Figura 2.3: Taxonomía de objetos urbanos

### 2.3. Algunas decisiones de diseño

Con el propósito de crear un modelo consistente, se debieron tomar algunas decisiones de diseño para poder adaptar de la mejor manera la información del diccionario a las necesidades del trabajo. Debido a que los autores del diccionario no disponían de mucho tiempo, no todas las decisiones tomadas fueron consensuadas con ellos. Por lo general, aquellas que involucraron grandes cambios sí se hicieron bajo su consentimiento.

Originalmente el diccionario contenía valores numéricos para ponderar atributos. Una de estas decisiones fue remplazar estos valores por valores cualitativos con el objetivo de obtener una ontología que fuera independiente de las restricciones numéricas. Esto a su vez facilitaría el trabajo de los expertos geógrafos y urbanistas, ya que razonar se vuelve más natural al hacerlo con valores cualitativos. Vale aclarar que esta modificación fue hecha en el diccionario por sus autores con valores puestos a conveniencia y justificados por los mismos.

De esta forma, atributos como por ejemplo superficie, tendrían valores como “Grand”, “Moyen” o “Petit”, en lugar de un valor numérico. Como casi todos los atributos, a excepción de la textura y la forma representan magnitudes, estos valores propuestos se adaptan perfectamente a pesar de la diferencias de escala y de unidades entre estos.

Decir que un jardín tiene un valor de superficie “Grand” no significa que tenga el mismo tamaño que un parque con valor de superficie “Grand”. En este caso, un mismo valor cualitativo tiene diferente peso según el contexto. Encontrar una relación entre los valores numéricos y los cualitativos para cada objeto, escapa de los límites de este trabajo.

Los valores cualitativos representan una forma abstracta de reproducir valores de atributos, sin embargo durante la conversión cuantitativa-cualitativa los autores han utilizado más de un valor cualitativo para denotar semánticamente lo mismo, por lo que se decidió mantener cierta homogeneidad en la ontología y remplazar esos valores repetidos por un sólo valor que los represente. Por ejemplo, “Petit” y “Faible” simbolizan la misma idea (“Pequeño”), por lo tanto se decidió que era conveniente utilizar un valor cualitativo “Petit” cuando hablamos sobre algo que tiene una magnitud pequeña. El mismo razonamiento se aplicó para los valores “Grand” y “Elevé” (Grande).

Para utilizar valores cualitativos, las *datatype properties* no pueden ser usadas debido a que estos valores no pertenecen a ningún tipo de datos conocido. Por lo tanto, se deben utilizar *object properties*, lo que implica la creación de una clase especial que represente un nuevo tipo dato que sólo contenga estos tres posibles valores. Lo que se decidió aquí fue crear una clase *Quantite* definida como la unión disjunta de las subclases: *Quantite\_Grand*, *Quantite\_Moyen* y *Quantite\_Petit*, que a su vez fueron definidas como disjuntas entre ellas. Esto significa que en cualquier momento, un objeto puede relacionarse con un miembro de una única subclase, ya sea de *Quantite\_Grand* o de *Quantite\_Moyen* o de *Quantite\_Petit*. A su vez, esto implica que si sabemos que un objeto está relacionado con un miembro de la clase *Quantite* y además tenemos conocimiento de que ese miembro no pertenece a *Quantite\_Grand* ni a *Quantite\_Moyen*, entonces con seguridad pertenece a *Quantite\_Petit*. Más adelante se entrará en detalle en las relaciones que se utilizarán para representar a los atributos de los objetos y estos valores cualitativos.

Para los atributos de textura, resolución y forma también fue necesario la creación de clases que representaran sus valores. Para esto se definieron 3 clases *Resolution*, *Texture* y *Forme* siguiendo la misma metodología aplicada para la definición de la clase *Quantite*.

Otra de las decisiones tomadas fue no modelizar atributos en aquellos objetos en los que no hubiera algún valor de referencia. Esto se decidió basado en la idea de que si un atributo no tiene un valor de referencia fue, o bien porque este atributo no importa para el objeto en cuestión y por lo tanto cualquier valor es aceptable, o bien porque no se tiene conocimiento de sus posibles valores. Sea cual sea el caso, si el atributo no se especifica en un objeto, estamos diciendo indirectamente que el atributo podría tomar cualquier valor para ese objeto. Esto se fundamenta por la hipótesis del mundo abierto, ya que al no especificar la falsedad del atributo, cualquier valor que lo haga verdadero es posible.

Existen algunos atributos y relaciones que sólo tienen sentido en los objetos compuestos a pesar de que muchas veces se encuentran en objetos simples. El atributo “Alignement” que hace referencia a si los objetos de un mismo tipo se encuentran organizados de manera alineada, no tiene sentido por ejemplo en el objeto *Pavillon* (un tipo de casa), ya que para que una casa pueda estar o no alineada debe hacerlo en referencia a otro objeto. Por lo tanto, se decidió que este atributo, al igual que “Distance entre barycentre” y “Discontinú”, sea utilizado en los objetos compuestos para describir las características espaciales de sus objetos constituyentes.

Entre la descripción textual y la tablas muchas veces se encuentran pequeñas contradicciones y otras veces complementan la información de las tablas. Se decidió que la información textual no sea tomada en cuenta en aquellas situaciones en las que aporte más dudas que certezas.

Dentro de la definición textual es común encontrar palabras que hacen referencia a los atributos y relaciones de las tablas. Entre estas se encuentran, “associé à” (asociado a) que puede ser interpretado como “adyacente a”, “organisés en” (organizado en) que puede interpretarse como “incluido en” y “proximité” (proximidad) que se puede entender como una relación de distancia pero acotada. Más adelante se tratará el caso de la relación de distancia, puesto que es un caso particular imposible de representar de manera directa con propiedades en OWL, debido a limitaciones del lenguaje para crear relaciones binarias ponderadas.

## 2.4. Modelado de relaciones y atributos

A continuación se analizarán cada una de las relaciones y atributos que se identificaron en secciones anteriores, con el objetivo de modelar cada una de estas en OWL. Para esto será necesario evaluar que tipo de propiedad mejor se adapta para cada caso (*object property* o *datatype property*) y cuales son sus características (dominio, rango, simetría, etc).

### La relación de adyacencia

La relación de adyacencia será modelada con la *object property* **isAdjacentTo**, debido a que, como anteriormente se dijo, debemos relacionar objetos con objetos. Analizando las características de esta propiedad podemos determinar:

- *Dominio y Rango*: La relación de adyacencia relaciona objetos geográficos con objetos geográficos sin tomar en cuenta si son simples o compuestos. Por lo tanto, el dominio y el rango de esta relación está conformado por la unión de objetos compuestos y simples. Puesto de otra forma, podemos decir la relación tiene rango y dominio en *Objet\_Geo*.
- *Simetría*: Si suponemos que existen un par de objetos *A* y *B*, tales que *A* es adyacente a *B*, podemos estar seguros de que *B* también es adyacente a *A*. Por lo tanto podemos asegurar que la relación de adyacencia es simétrica.
- *Funcionalidad*: No es una relación de tipo funcional, ya que un objeto puede ser adyacente a varios otros al mismo tiempo.
- *Transitividad*: No es una relación transitiva. Por ejemplo, si consideramos tres objetos *A*, *B* y *C*, de los cuales se conoce que *A* y *B* son adyacentes entre sí y que *B* y *C* también lo son, no es posible afirmar con esta información que *A* y *C* también lo sean, ya que la adyacencia entre *A* y *C* es una relación directa que no depende de otros objetos.
- *Reflexividad*: No es reflexiva puesto que ningún objeto es adyacente a sí mismo.
- *Inversa*: No tiene relación inversa asociada.

### La relación de composición

Debido a que un objeto se compone de otros, la relación de composición es una relación entre objetos y por lo tanto deberá ser modelada con una *object property*, la cual llamaremos **isComposedOf**.

- *Dominio y Rango*: Por el análisis hecho sobre la jerarquía de objetos sabemos que la relación de composición sólo tiene sentido para la clase de objetos compuestos. Y que los objetos que componen a otro pueden ser simples o compuestos. Entonces, podemos afirmar que la relación de composición relaciona objetos de la clase *Objet\_Construit* (objetos compuestos o contruidos) con miembros de la clase *Objet\_Geo*. Por lo tanto, el dominio de la relación es cualquier objeto de la clase *Objet\_Construit* y el rango es cualquier objeto que sea compuesto o simple, es decir, cualquier objeto de la clase *Objet\_Geo*.
- *Simetría*: Esta relación no puede ser simétrica debido a que un objeto *A* no puede estar compuesto por otro objeto *B* que a su vez este compuesto por *A*. Esta situación sólo puede darse si los objetos son equivalentes.

- *Funcionalidad*: Debido a que un objeto puede estar compuesto por más de uno, la composición no puede ser funcional.
- *Transitividad*: Si suponemos que existen tres objetos  $A$ ,  $B$  y  $C$ , tales que  $A$  está compuesto por  $B$  y  $B$  está compuesto por  $C$ , podríamos decir que  $A$  también está compuesto por  $C$ , pese a que  $B$  predomina sobre  $C$ .
- *Reflexividad*: No tiene sentido decir que un objeto está compuesto por sí mismo.
- *Inversa*: Analizando esta relación, podríamos decir que los objetos que componen a otro además están incluidos en este. Se podría considerar a relación de inclusión la inversa de la composición y viceversa.

### La relación de inclusión

La relación de inclusión, al igual que la de composición, relaciona objetos con objetos y por lo tanto debe ser modelada mediante la *object property* **isIncludedIn**.

- *Dominio y Rango*: Como se mencionó durante el análisis de la relación de composición, la inclusión y la composición están relacionadas de manera inversa. Por lo tanto, podemos inferir que el dominio de la relación serán los objetos de la clase de *Objet\_Geo* y el rango estará determinado por los objetos miembros de la clase *Objet\_Construit*.
- *Simetría*: No puede ser simétrica ya que, por ejemplo, un objeto  $A$  compuesto por varios objetos, entre ellos un objeto simple  $B$ , este último por definición no puede incluir o componerse de ningún objeto, en particular no puede incluir a  $A$ .
- *Funcionalidad*: La funcionalidad de esta relación a simple vista parece ser posible. Suena lógico decir que un objeto no puede estar incluido en más de un objeto al mismo tiempo. Sin embargo, casos como el siguiente, donde un objeto  $A$  está incluido en otro  $B$  que a su vez está incluido en otro objeto  $C$  implicaría que  $A$  también esté incluido en  $C$ , lo cual sería imposible para una relación funcional.
- *Transitividad*: Del análisis anterior se desprende que esta relación es transitiva.
- *Reflexividad*: No tiene sentido decir que un objeto está incluido en sí mismo.
- *Inversa*: Tal y como se dedujo durante el estudio de la relación de composición, la inversa de la inclusión es la relación de composición.

### La relación de distancia

El caso de la relación de distancia es diferente al resto de las relaciones, ya que esta vez se tiene que relacionar objetos con objetos pero a su vez se necesita especificar con qué magnitud ocurre esa relación. Previamente se mencionó que las *Properties* en OWL son vistas como relaciones binarias. En este caso, estaríamos relacionado dos objetos con un valor de distancia, en lo que vendría a ser una relación ternaria. Esto significa que las *Properties* no son útiles para este caso y se necesita buscar otra forma de representar esto. Una posible solución consiste en reificar la relación, lo que significa tratar a la relación como a una entidad. El propósito de reificar una relación es hacerla explícita cuando es necesario añadir información adicional. Una forma general de reificación es crear una nueva clase representando la relación y  $n$  nuevas *object properties* para completar una relación  $n$ -aria. La manera de recrear la relación de distancia con este método es el que se muestra en la figura 2.4.

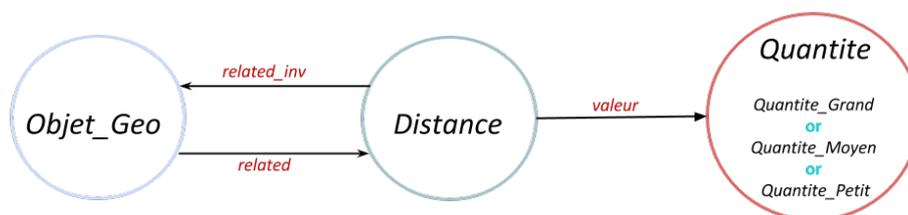


Figura 2.4: Reificación de la relación de distancia

Para esta relación ternaria se crearon tres nuevas *object properties* y una clase *Distance* que simboliza la relación. La propiedad **related**, relaciona a un miembro  $A$  de la clase *Objet\_Geo*, puesto que la distancia puede aplicarse a cualquier objeto tanto compuesto como simple, con un miembro  $D$  de la clase *Distance*. Esta última clase sirve a modo de puente entre los objetos relacionados y un valor  $V$  cualitativo de distancia, los cuales se relacionan mediante la propiedad funcional **valeur**. Por último, para cerrar el ciclo, la propiedad **related\_inv**, la cual es la inversa de **related**, relaciona al individuo  $D$  de la clase *Distance*, que estaba relacionado con  $V$  y con  $A$ , con un individuo  $B$  de la clase *Objet\_Geo*. De esta forma se estaría emulando una relación ponderada de valor  $V$  entre los individuos  $A$  y  $B$ .

### Los atributos

Para modelar los atributos de un objeto originalmente usaríamos *datatype properties* para relacionar al mismo con sus posibles valores numéricos. Sin embargo, al eliminar los valores cuantitativos y reemplazarlos por cualitativos, esto ya no es posible y por lo tanto, deberemos utilizar *object properties*. Los atributos identificados en la sección 2.1 serán detallados a continuación.

- *Longueur*: ( o Longitud) Se lo considera la distancia entre los extremos más lejanos de un objeto. Relaciona objetos de la clase *Objet\_Geo* con los miembros de la clase *Quantite*. Será identificada mediante la *object property* **hasLongueur**.
- *Largeur* : (o Amplitud) Representa la longitud más larga perpendicular al *Longueur*. Relaciona miembros de la clase *Objet\_Geo* con los miembros de la clase *Quantite*. Será identificada mediante la *object property* **hasLargeur**.
- *Elongation*: Representa al cociente entre *Largeur* y *Longueur*, y relaciona a cualquier miembro de la clase *Objet\_Geo* con los miembros de la clase *Quantite*. Será identificada mediante la *object property* **hasElongation**.
- *Densité*: (o Densidad) Este atributo se utiliza en algunos objetos compuestos para representar el grado de concentración de los objetos que lo componen. Por ejemplo, los objetos de tipo *TU* (Tejido Urbano), los cuales se componen principalmente de casas y de otros tipos de edificaciones, pueden considerarse más o menos densos de acuerdo al nivel de concentración de estos elementos en relación a su superficie. Dado que este atributo tiene sentido en algunos objetos compuestos, el mismo relaciona miembros de la clase *Objet\_Construit* con valores cualitativos de la clase *Quantite*, la cual se utiliza para representar el nivel de densidad de los elementos que componen al objeto compuesto. Será identificada mediante la *object property* **hasDensity**.
- *Surface*: Representa el tamaño de la superficie del objeto, el cual puede tomar los valores cualitativos de la clase *Quantite*. Será identificada mediante la *object property* **hasSurface**.
- *Forme*: Indica la forma del objeto. Relaciona objetos geográficos con miembros de la clase *Forme*, los cuales a su vez pueden ser miembros de una de las siguientes subclases: *Forme\_Carre*, *Forme\_Rectangulaire*, *Forme\_Circulaire*, *Forme\_Lineaire*, *Forme\_Triangulaire* o *Forme\_Autre* en caso de que ninguna de las anteriores subclases se ajuste a la forma del objeto. Será identificada mediante la *object property* **hasForme**.
- *Texture*: En una imagen digital las variaciones espaciales de tono o color entre los píxeles pueden entenderse como un cierto grado de “rugosidad” o de textura, la cual puede ser calificada como homogénea, cuando dicha variación es mínima, o heterogénea en el caso contrario. Este atributo, por lo tanto, relaciona objetos urbanos con miembros de la clase *Texture*, que a su vez pueden ser alguna de las subclases *Texture\_Heterogene* (Heterogénea) o *Texture\_Homogene* (Homogénea). Será identificada mediante la *object property* **hasTexture**.
- *Resolution*: Este atributo relaciona a un objeto urbano con los posibles intervalos de resolución espacial en los cuales puede ser identificado. Esto significa que los objetos son reconocibles y clasificados según las características intrínsecas que pueden ser observadas a esas resoluciones. Recordemos que estos intervalos representan los rangos de tamaño para el detalle más pequeño que puede ser detectado y están definidos de la siguiente forma:

- THR1: [0.5m to 5m]
- HR1: [5m to 15m]
- HR2: [15m to 30m]

Será identificada mediante la *object property* **hasResolution**

Todos los atributos serán representados mediante **Object Properties** funcionales, dado que un objeto no puede tener más de un valor cualitativo a la vez.

Como caso especial, tenemos el atributo “Alignement”. Este representa cuando un objeto compuesto contiene a sus objetos alineados de alguna forma. En este caso es posible utilizar los beneficios de OWL para utilizar tipos de datos y representar esta relación como una *datatype property* llamada **isAlignee** con dominio en la clase *Objet\_Construit* y rango de tipo booleano. Como las *datatype properties* no pueden ser transitivas, ni simétricas, ni puede tener inversa, sólo nos queda analizar la funcionalidad. Dado que un objeto compuesto sólo puede estar en uno de dos estados posibles de alineación (verdadero o falso), esta propiedad es puramente funcional.

Al igual que con “Alignement”, el atributo “Discontinú” o “Discontinue”, que hace referencia a si un objeto tiene o no alguna discontinuidad espacial en su superficie, será modelizado con una *datatype property* de rango booleano y dominio *Objet\_Geo*.

Las siguientes figuras representan ejemplos de objetos geográficos ilustrando algunos de los atributos mencionados.

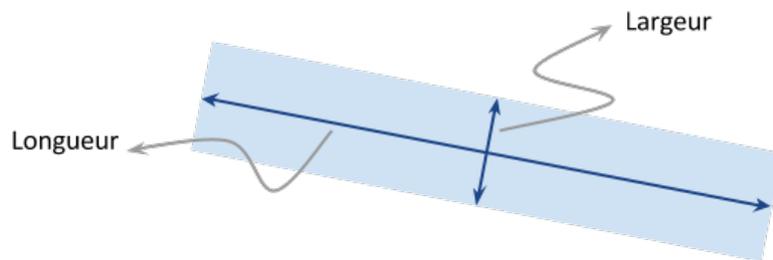


Figura 2.5: Figura rectangular

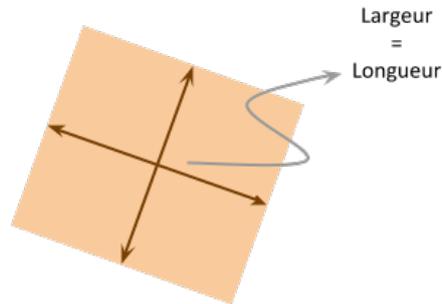


Figura 2.6: Figura cuadrada

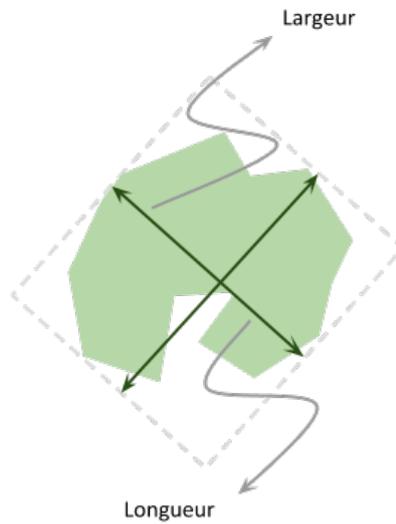


Figura 2.7: Figura sin forma reconocible

## 2.5. Modelado de las clases

Como se mencionó en el apartado “Clases e Instancias” de la sección 1.2.1, para definir clases es necesario crear restricciones, las cuales deberán estar escritas en términos de las relaciones y atributos definidas anteriormente, combinándolas convenientemente por medio de constructores.

Para comenzar con el análisis, tomaremos la ficha *Pavillon* (un tipo de casa), que se muestra en la figura 2.8, como ejemplo para hacer más sencillo el análisis. Para el resto de los objetos identificados en la tabla 2.1 se utilizará la misma metodología.

### Fiche 1 : Bâtiment de l'habitat individuel (Pavillon)

#### A. Identification de l'objet

Type	Nom de l'objet :	Type d'objet élémentaire	Résolution
 Polygone	<i>Pavillon</i>	<i>Bâtiment</i>	THR1

#### B. Description de l'objet dans le monde réel

##### B.1 Définition textuelle

L'objet « pavillon » ou « maison individuelle » appartient à la catégorie d'objets élémentaires « bâtiment ». Il désigne une construction durable destinée à abriter l'activité humaine reliée à l'habitat.

La portée de cette définition est restreinte par les critères suivants. En général, un pavillon :

- est situé dans un îlot physique (domaine privé) ;
- a une emprise au sol d'au moins 12 m<sup>2</sup> ;

Ces critères visent à exclure notamment les abris, les aubettes, les remises à outil, cabanes de jardin etc. de la définition de l'objet « pavillon ».

Le pavillon ou maison individuelle est le plus souvent organisée en lotissement ou cité (cf. **objet construit**). Il peut également être disposé en bandes ou mitoyen.

##### B.2 Illustration graphique : THR

	<i>THR1 – QB MS (2.8 m)</i>		
<b>- Pavillon</b>			

##### C.3. Principales relations

Adjacence	Objets de type « végétation » Objets de type « autre route »
Alignement	oui
Distance entre barycentre	2 à 15m Faible (= appartenance à une cité ouvrière, individuel dense) Moyenne (individuel moyennement dense) Elevée (individuel lâche)
Distance à la route	Faible / Moyen /Elevée 4 à 12m
Inclusions	TU discontinu de type individuel HR1 ( <b>fiche 31</b> ) et/ou HR2 ( <b>fiche 33</b> )

##### C.4. Attributs

Longueur ou diamètre (m)	Faible - 10 à 15
Largeur (m)	Faible - 10 à 15
Elongation (m)	Faible - 10 à 15
Surface (m <sup>2</sup> )	Petite - 100 à 225
Indice de forme	Carré à rectangulaire - I <sub>M</sub> = 0.51 à 0.63
Texture	Homogène
Orientation de l'élongation (°)	/
Hauteur (en nb d'étages)	1 à 3

Figura 2.8: Descripción del objeto *Pavillon*

La primera información de un *Pavillon* que podemos obtener a simple vista es:

- Es un tipo de *Bâtiment*. Esto ya se dedujo en la sección 2.2.
- Es sólo reconocible bajo resolución THR1.
- Es adyacente a objetos de *Végétation* y *Autre Route*.
- Está incluido en objetos de tipo *TU Discontinu* and *TU Discontinu de type individuel*.
- Tiene una longitud pequeña (Longueur “Petit”).
- Tiene una amplitud pequeña (Largeur “Petit”).
- Tiene una elongación pequeña (Elongation “Petit”).
- Tiene una superficie pequeña (Surface “Petit”).
- Tiene una forma que varía de Rectangulaire a Cuadrada (Forme “Rectangulaire” a “Carre”).
- Tiene textura Homogénea (Texture “Homogene”).

Notar que aquí no estamos usando la relación de alineación porque, como se ha discutido en la sección 2.3, la alineación es una propiedad de los objetos compuestos y *Pavillon* no es uno de ellos.

Al mirar la definición textual la única información relevante que podemos extraer es que un *Pavillon* tiene una superficie de al menos  $12m^2$ , y que está incluido en algún *îlot physique*. Esta información no es útil para los propósitos de la ontología, primero porque los valores numéricos no se utilizan más, y segundo porque esas propiedades ya están especificadas en la tabla.

Ahora necesitamos evaluar cada una de las afirmaciones que hemos capturado del objeto con el propósito de convertir esta información en restricciones que puedan usarse para definir la clase *Pavillon*.

Analizando la adyacencia del objeto *Pavillon* podemos ver que sus objetos adyacentes son *Végétation* y *Autre Route*. Sin embargo esto no significa que todos los objetos *Pavillon* en nuestro dominio serán adyacentes a esos dos objetos al mismo tiempo. Tampoco está significando que esos son los únicos objetos que son adyacentes a un *Pavillon*, porque es imposible asegurar que así lo sea en todos los casos. Lo que esta sentencia está diciendo es que, en general, un *Pavillon* se caracteriza por ser adyacente a por lo menos uno de estos objetos. Esto se puede escribir en OWL utilizando una restricción sobre la relación **isAdjacent** con un cuantificador existencial de la siguiente forma:

**isAdjacentTo** some (*Vegetation*  $\vee$  *Route\_Autre*)

En este caso la relación de inclusión establece que un *Pavillon* está incluido en un objeto de tipo *TU Discontinu* o en uno de tipo *TU Discontinu de type individuel*, por ende, para garantizar la inclusión en al menos uno de estos objetos, ya que la transitividad de la relación hace que sea posible la inclusión en otros objetos, necesitamos utilizar un cuantificador existencial:

**isIncludedIn** some (*TU\_Discontinu*  $\vee$  *TU\_Discontinu\_Individuel*)

La relación de distancia no representa una restricción en este caso ya que como un *Pavillon* puede estar a cualquier distancia de la ruta, todos los valores de distancia son posibles.

Con esto se finalizaría el modelado de las relaciones espaciales del objeto. Para los atributos, el análisis es similar, ya que también se utilizarán *object properties* para su representación. Tomemos como ejemplo el caso del atributo “Longueur” , el cual es representado por la *object property* **hasLongueur**. Según el diccionario, un *Pavillon* tiene un “Longueur Petit” (o longitud pequeña), por lo tanto, en nuestra representación esto significa que la propiedad relaciona objetos de tipo *Pavillon* con un valor pequeño de cantidad. Generalizando un poco, como *Longueur* es un atributo presente en casi todos los objetos urbanos, tanto simples como compuestos, podemos decir que **hasLongueur** relaciona miembros de la clase *Obj\_Geo* con miembros de la clase *Quantite\_Petit*. Usualmente esta restricción se hubiera escrito con un cuantificador universal, dado que parece ser que la restricción sólo admite la relación entre objetos urbanos y miembros de la clase *Quantite\_Petit*. Sin embargo, dado que se debe garantizar la existencia de la relación, se deberá utilizar un cuantificador existencial. El único problema sería lograr que la restricción sólo permitiera la relación entre objetos urbanos y miembros de la clase *Quantite\_Petit*. Para esto, normalmente se escribiría lo que se denomina un axioma de clausura, el cual consiste en crear una intersección entre la restricción cuantificada existencialmente y la misma restricción pero escrita con un cuantificador universal. De esta manera, si se satisfacen ambas restricciones, la existencia y la unicidad están garantizadas. Por fortuna esto no es necesario. Debido a que los atributos se definieron con relaciones funcionales, cuando un objeto de tipo *Pavillon* se relacione con al menos un valor de cantidad pequeño, lo cual se garantiza por la cuantificación existencial, la funcionalidad asegura que no podrá haber ningún otro valor relacionado con ese objeto en ese momento, y por lo tanto será el único.

Habiendo dejado esto en claro, podemos escribir en OWL esta restricción de la siguiente forma:

**hasLongueur** some *Quantite\_Petit*

Para el caso del atributo “Forme” , la descripción del objeto *Pavillon* expresa que los objetos de este tipo tienen una forma que puede ir de rectangular a cuadrada. Continuando el análisis realizado con el atributo “Longueur”, esto podemos escribirlo en OWL de la siguiente forma:

**hasForme** some (*Forme\_Rectangulaire*  $\vee$  *Forme\_Carre*)

Para el resto de los atributos presentes en la definición de *Pavillon* el análisis es análogo y por ende también lo es su representación en OWL.

Una vez traducidas las restricciones textuales en forma lógica, debemos analizar si esas restricciones deben definir a la clase *Pavillon* como **Definida** o **Primitiva**.

Si se definiera a *Pavillon* como a una clase **Primitiva**, es decir escribiendo las restricciones como condiciones necesarias, esto sería equivalente a decir que  $Pavillon \Rightarrow Restricciones$ , lo cual significa que si un individuo es miembro de la clase *Pavillon* entonces satisface las restricciones, lo cual es cierto. Sin embargo, con esto solamente no alcanza, ya que definido de esta forma no podemos asegurar que si un individuo cumple con las restricciones entonces es miembro de *Pavillon*. Otras de las desventajas de definir una clase como primitiva es que los razonadores no pueden inferir automáticamente sus miembros. El objetivo es capturar toda la información que se tenga sobre un objeto geográfico urbano, ya sea proveniente de la salida de un software especializado de reconocimiento de imágenes o del conocimiento de un experto tras analizar una imagen satelital, e ingresar esa información en la ontología para conocer a cual de todas las clases definidas por el diccionario pertenece. Para lograr esto, necesitamos definir las clases como **Definidas**, esto significa definir las clases como si fueran clases equivalentes, o puesto de otra forma,  $Pavillon \Leftrightarrow Restricciones$ .

Sin embargo tenemos varias formas de hacer esto. Si tratamos a cada restricción como una condición como necesaria y suficiente implícitamente estaríamos considerando una disjunción de restricciones que en conjunto forman una condición necesaria y suficiente, es decir algo de la siguiente forma:

$$Pavillon \Leftrightarrow \left\{ \begin{array}{l} \text{Restriccion 1} \\ \vee \\ \text{Restriccion 2} \\ \vee \\ \text{Restriccion 3} \\ \vee \\ \vdots \\ \vee \\ \text{Restriccion N} \end{array} \right.$$

Escrito de esta forma, *Pavillon* sería equivalente a cada una de las clases anónimas definidas por cada restricción. Esto implica que un miembro de una de estas clases anónimas será miembro de *Pavillon*.

Para hacer aún más evidente este problema, consideremos como ejemplo la restricción de adyacencia de *Pavillon*, la cual define la clase de objetos que pueden ser adyacente a por lo menos un objeto de tipo *Végétation* o *Autre Route*. En el diccionario podemos encontrar objetos que son adyacentes a *Autre Route* entre otros objetos, como por ejemplo *Chemin* (Camino). Esta última restricción conforma una clase anónima más general que incluye a la anterior (figura 2.9).

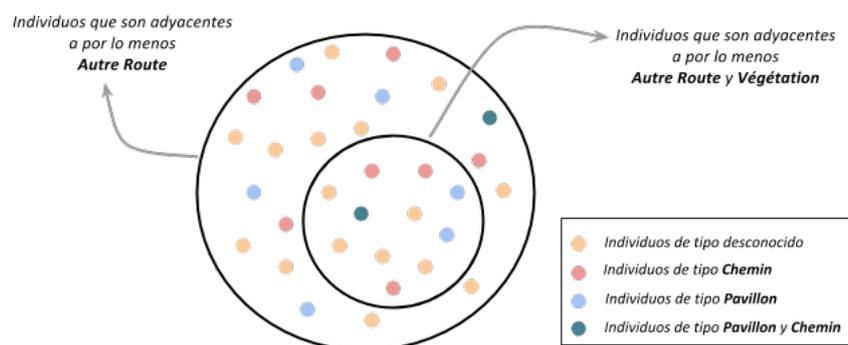


Figura 2.9: Equivalencia entre las clases *Pavillon* y *Chemin*

Si suponemos que *Chemin* se definió de la misma forma que *Pavillon*, podemos asegurar que la clase definida por los objetos adyacentes a por lo menos *Autre Route* es también equivalente a la clase *Chemin*. Esto no impide la existencia de algún individuo adyacente a *Autre Route* que simultáneamente sea clasificado como *Pavillon* y como *Chemin* (individuos de color verde en la figura 2.9), ya que las restricciones de adyacencia de ambas clases se satisfacen, lo que implica que *Pavillon* y *Chemin* sean equivalentes. Si estos objetos se definieran como disjuntos<sup>3</sup>, lo cual parece lo más lógico, ocurriría una inconsistencia en la ontología.

Pero supongamos por el momento que la afirmación de que ambos objetos sean disjuntos no existe. Tomando la restricción de elongación de cada objeto, observamos que un *Pavillon* tiene una elongación pequeña, mientras que un *Chemin* posee una elongación grande. Las clases anónimas definidas por estas restricciones son disjuntas, ya que las subclases cualitativas *Quantite\_Petit* y *Quantite\_Grand*, utilizadas para representar magnitudes pequeñas y grandes respectivamente, fueron definidas como disjuntas anteriormente. Por lo tanto, como cada uno de estos objetos es equivalente a las clase anónima definida por su respectiva restricción, estaríamos representando una disjunción entre las clases *Pavillon* y *Chemin* (figura 2.10).

Si tomamos al mismo individuo adyacente a *Autre Route* del ejemplo anterior (individuos de color verde de la figura 2.9) y agregamos a su definición que posee una elongación pequeña, sería ahora clasificado sólo como *Pavillon*. Sin embargo, como el individuo había sido clasificado anteriormente como miembro de *Pavillon* y *Chemin*, se generará una inconsistencia en el modelo debido a que no puede haber un individuo miembro de *Chemin* que tenga elongación pequeña, e inversamente, no puede existir un individuo de tipo *Pavillon* con elongación grande.

<sup>3</sup>La disjunción de objetos será un tema a tratar en la próxima sección.

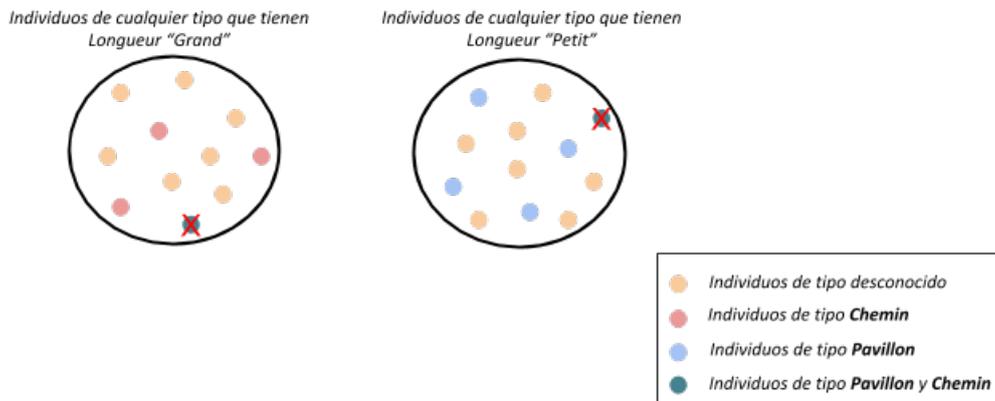


Figura 2.10: Ejemplo de Inconsistencia

El considerar a las restricciones de manera individual trae aparejado este tipo de inconsistencias en la ontología. Una manera de solucionar esto es crear una única restricción que represente la conjunción de todas las restricciones de un objeto. Lo cual tiene sentido, debido a que un objeto se define por todas sus características en conjunto.

Utilizando esta proposición, podemos describir la clase *Pavillon* de la siguiente forma:

$$\text{Pavillon} \Leftrightarrow \left\{ \begin{array}{l}
 \text{hasResolution some } THR1 \\
 \wedge \\
 \text{isAdjacentTo some } (Route\_Autre \vee Vegetation) \\
 \wedge \\
 \text{isIncludedIn some } (TU\_Discontinu \vee TU\_Discontinu\_Individuel) \\
 \wedge \\
 \text{hasSurface some } Quantite\_Petit \\
 \wedge \\
 \text{hasLongueur some } Quantite\_Petit \\
 \wedge \\
 \text{hasLargeur some } Quantite\_Petit \\
 \wedge \\
 \text{hasElongation some } Quantite\_Petit \\
 \wedge \\
 \text{hasForme some } (Forme\_Carre \vee Forme\_Rectangulaire)
 \end{array} \right.$$

En **Protégé** esto se puede representar de la siguiente manera:



Figura 2.11: Clase *Pavillon* en Protégé

Esta será la metodología de trabajo para evaluar y describir las clases de objetos urbanos. Hay algunos casos especiales que tendrán lugar para el análisis en las siguientes secciones, pero en líneas generales, la representación de cada objeto se hará de esta forma.

Con esta representación, se reducen las inconsistencias en la ontología, ya que ahora, uno de los motivos por los que pueden ocurrir es si por ejemplo información incorrecta es ingresada de manera explícita. Esto puede ocurrir, al especificar dos valores cualitativos diferentes para un mismo atributo, lo cual puede solucionarse si se utilizara alguna interfaz de usuario que impida ingresar esa información más de una vez (más adelante se tratará esto).

La disjunción de objetos urbanos trae aparejada otra potencial fuente de inconsistencias. En la próxima sección veremos que todavía es posible, aunque poco probable, que ante cierta combinación de atributos, un objeto sea miembro de más de una clase, lo cual haría inconsistente al modelo si todos los objetos fueran considerados como disjuntos. A simple vista parecería que la disjunción de objetos es la peor opción, sin embargo no debemos olvidar el compromiso asumido de modelar el dominio de la manera más fiel posible, de lo contrario estaríamos permitiendo situaciones que no serían representables en el mundo real, o tendríamos un modelo deficiente que sería insuficiente para representar todo el dominio.

En la próxima sección se tratará, entre otros temas relacionados con el diseño del modelo, el dilema de la disjunción de objetos urbanos.

### 2.5.1. Decisiones de diseño sobre la ontología

#### Disjunción de objetos geográficos

Parecería obvio decir que los objetos geográficos definidos en el diccionario son disjuntos entre sí. Sin embargo veremos que no es tan sencillo como parece.

Uno de principales problemas del diccionario de objetos urbanos es la falta de información para describir completamente a algunos de estos objetos. La falta de información,

junto a la débil<sup>4</sup> restricción utilizada para caracterizar la adyacencia de objetos, hacen muy difícil poder expresar que dos objetos son diferentes cuando sus datos indican que semánticamente son equivalentes, o que una clase no es una generalización de otra porque no tenga suficiente información para establecer lo contrario.

Además de tener algunos objetos descriptos deficientemente, en el otro extremo tenemos objetos que describen sus atributos y relaciones con un rango amplio de valores, lo que muchas veces lleva a la superposición de clases cuando estos atributos y relaciones se combinan convenientemente con determinados valores. Si un individuo satisface las restricciones de dos o más objetos urbanos, decimos entonces que se produjo una “multi-clasificación” del individuo como miembro de esos objetos, producto de la combinación adecuada de sus características. La falta de información en la definición de las clases de los objetos urbanos también es una causa de la “multi-clasificación”. Por ejemplo, un individuo que es miembro de una de estas clases definidas de manera incompleta seguirá siendo miembro de esta clase aún si nueva información se agregara al individuo. Esto es porque OWL asume por la hipótesis del mundo abierto que, ante la falta de información concreta que indique lo contrario, esta nueva información puede ser válida para la clase y entonces clasifica al individuo como tal. Por lo tanto, es posible que esta nueva información, junto con la que ya tenía el individuo, sea suficiente para satisfacer las restricciones de alguna de las clases de objetos urbanos, lo que llevaría a la “multi-clasificación” del individuo.

Para ilustrar el problema de la “multi-clasificación”, tomemos el caso de los objetos *Canal* y *Cours d’Eau*, con tipo elemental *Eau*. Con el fin de demostrar este punto de manera sencilla, solamente consideraremos las restricciones correspondientes a la resolución HR2.

$$\text{Canal} \Leftrightarrow \left\{ \begin{array}{l}
 \text{hasResolution some } HR2 \\
 \wedge \\
 \text{isAdjacentTo some } (Surface\_Boisee \vee TU\_Discontunu\_Individuel \\
 \vee TU\_Discontunu\_Collectif) \\
 \wedge \\
 \text{hasLargeur some } Quantite\_Grand \\
 \wedge \\
 \text{hasTexture some } Texture\_Homogene \\
 \wedge \\
 \text{hasElongation some } Quantite\_Grand \\
 \wedge \\
 \text{hasForme some } Forme\_Lineaire
 \end{array} \right.$$

---

<sup>4</sup>Con débil nos referimos a que no estamos limitando de manera estricta la adyacencia a aquellos objetos que aparecen en la definición de un objeto urbano, sino que permitimos la existencia de otros si al menos alguno de los objetos especificados está presente.

$$\text{Cours d'Eau} \Leftrightarrow \left\{ \begin{array}{l}
 \text{hasResolution some } HR2 \\
 \wedge \\
 \text{isAdjacentTo some } (Surface\_Boisee \vee Pelouse \\
 \vee Alignement\_Arbres \\
 \vee Zone\_Ind\_Com\_Tert \\
 \vee TU\_Discontinu\_Collectif \\
 \vee TU\_Discontinu\_Individuel) \\
 \wedge \\
 \text{hasLargeur some } (Quantite\_Grand \vee Quantite\_Moyen) \\
 \wedge \\
 \text{hasTexture some } Texture\_Homogene \\
 \wedge \\
 \text{hasElongation some } Quantite\_Grand \\
 \wedge \\
 \text{hasForme some } Forme\_Lineaire
 \end{array} \right.$$

Como se puede apreciar, ambas clases son identificables bajo resolución HR2, se caracterizan por tener la misma textura, por poseer una gran amplitud, por tener una forma lineal y por compartir algunos objetos adyacentes.

Tomemos como ejemplo al individuo “TEST\_1”, definido con las mismas características comunes de estos objetos, pero adyacente a objetos de tipo *Pelouse*. Luego de la ejecución de un razonador lógico, en este caso **Hermit** [19], el individuo se clasifica como miembro de *Course d'Eau*. En la figura 2.12 podemos observar, en el panel derecho las afirmaciones hechas sobre este individuo, y en el panel izquierdo el tipo de objeto inferido por el razonador. Los individuos *iGrand*, *iLineaire*, *iHomogene*, *iHR2* y *pelouse*, de tipos *Quantite\_Grand*, *Forme\_Lineaire*, *Texture\_Homogene*, *Resolution\_HR2* y *Pelouse* respectivamente, fueron creados para este ejemplo en particular.

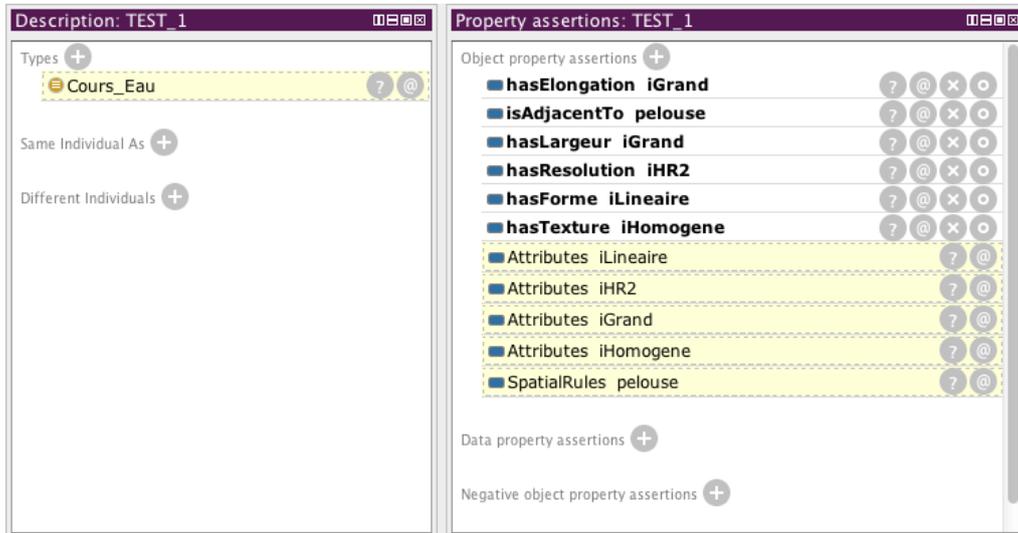


Figura 2.12: Clasificación del individuo “TEST\_1” como miembro de *Cours d'Eau*

Si ahora suponemos que se detecta otro objeto adyacente, como ser un objeto de tipo *Surface Boisee*, el individuo sería identificado como miembro de *Curs d'Eau* y *Canal*, por estar este último objeto definido como adyacente a por lo menos un objeto de tipo *Surface Boisee*. La figura 2.13 ilustra esta situación.

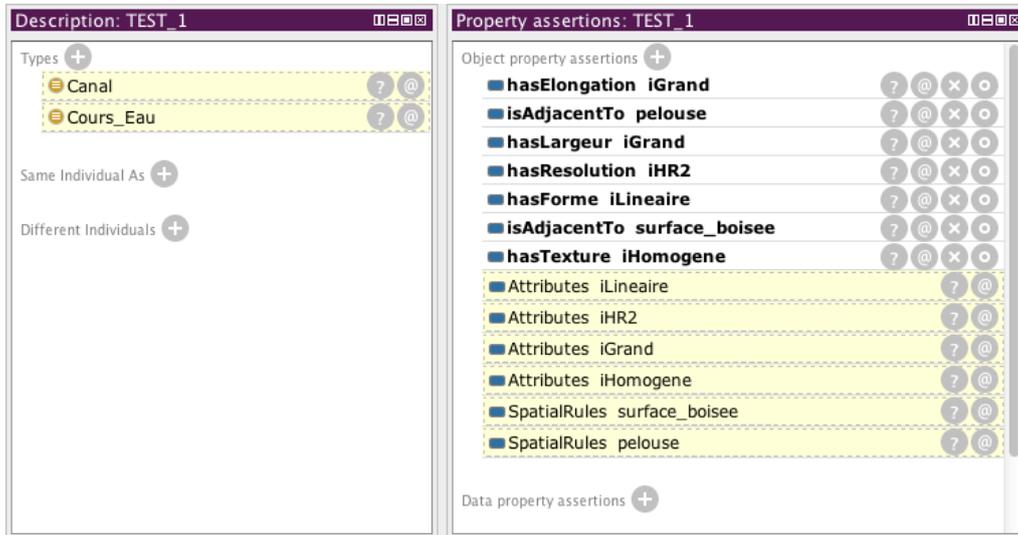


Figura 2.13: Clasificación del individuo “TEST\_1” como miembro de *Curs d'Eau* y de *Canal*

El caso más grave ocurre cuando en la “multi-clasificación” participan objetos con diferentes tipos elementales. Por ejemplo, tomemos el caso de los objetos *Jardin* y *Cimetiere*:

$$Cimetiere \Leftrightarrow \left\{ \begin{array}{l} \text{hasResolution some } THR1 \\ \quad \wedge \\ \text{isAdjacentTo some } (Chemin\_de\_Fer \vee Route) \\ \quad \wedge \\ \text{hasSurface some } Quantite\_Grand \\ \quad \wedge \\ \text{hasTexture some } Texture\_Heterogene \end{array} \right.$$

$$Jardin \Leftrightarrow \left\{ \begin{array}{l} \text{hasResolution some } THR1 \\ \quad \wedge \\ \text{isAdjacentTo some } (Chemin\_de\_Fer \vee Route\_Grande\_Vitesse) \\ \quad \wedge \\ \text{hasSurface some } Quantite\_Grand \\ \quad \wedge \\ \text{hasTexture some } Texture\_Heterogene \end{array} \right.$$

Estas clases además de diferir en su tipo elemental (*Végétation* y *Equipement Divers* respectivamente), difieren en algunos de sus objetos contiguos. *Cimetiere* además de ser adyacente a objetos de tipo *Chemin de Fer*, al igual que *Jardin*, también se encuentra próximo a objetos de tipo *Route*, el cual es un tipo más general definido especialmente para englobar a objetos similares, como por ejemplo *Route a Grande Vitesse*.

Si definimos un individuo “TEST\_2” con la misma resolución de estos objetos, el mismo tamaño de superficie, la misma textura, y próximo a un objeto de tipo *Chemin de Fer*. Luego de la ejecución del razonador, este individuo se clasifica satisfactoriamente como miembro de *Jardin*.

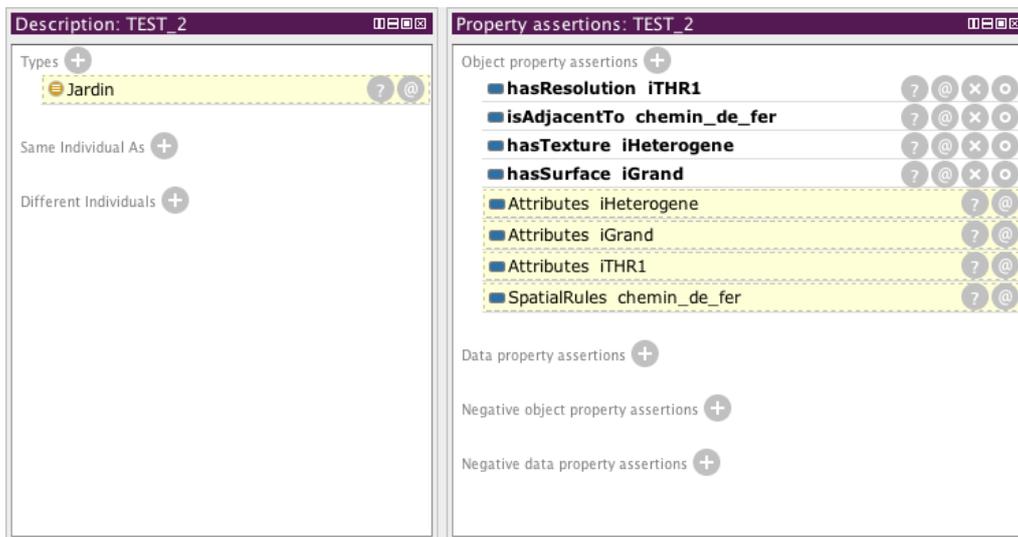


Figura 2.14: Clasificación del individuo “TEST\_2” como miembro de *Jardin*

Sin embargo, dado que *Cimetiere* es adyacente a *Route* el cual fue definido como un objeto más general que *Route a Grande Vitesse*, y teniendo en cuenta que *Jardin* y *Cimetiere* tienen casi las mismas características, el razonador clasifica a *Jardin* como subclase de *Cimetiere* (figura 2.15), lo cual implica que “TEST\_2” también es miembro de *Cimetiere* (figura 2.16), produciéndose otra “multi-clasificación”.

Estos resultados no eran visibles en la figura 2.14 debido a que **Protégé** sólo muestra la inferencia directa de tipos en la vista de descripción de individuos.



Figura 2.15: Clasificación de *Jardin* como subclase de *Cimetiere*

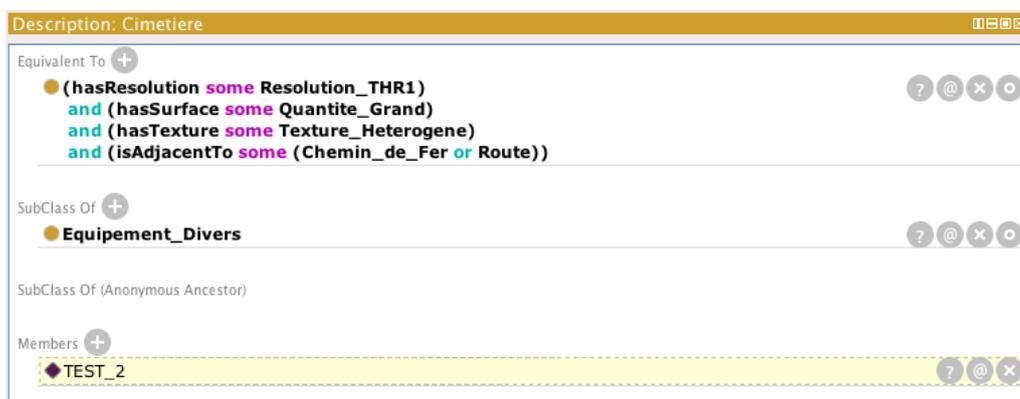


Figura 2.16: Clasificación del individuo “TEST\_2” como miembro de *Cimetiere*

En el primer caso (figura 2.13), la “multi-clasificación” no sería tan grave, puesto que se trata de objetos del mismo tipo elemental, con lo cual un error de este tipo podría considerarse menor, o inclusive podría ser utilizado a favor, dependiendo del propósito final de la ontología.

En el último caso (figura 2.14) la situación es un poco más grave, ya que sería un error conceptual permitir que un individuo sea clasificado bajo diferentes tipos elementales. Para no permitir este tipo de situaciones debería definirse al menos la disjunción de objetos elementales. Sin embargo, esto no impide la clasificación de objetos con más de un tipo elemental, sólo establece que si esto ocurriese, se estaría violando una regla del modelo, volviéndolo inconsistente.

Para impedir este tipo de inconsistencias, necesitamos de algún mecanismo que impida que un individuo sea clasificado bajo diferentes clases elementales. Una posible solución sería ingresar información competente al tipo elemental del individuo a clasificar con el objetivo de forzar la clasificación bajo una misma clase elemental. Pero, dado que en el diccionario no existe información específica que permita identificar y diferenciar a las clases de objetos elementales, no es posible la clasificación de un individuo como miembro de alguna de estas clases a menos que se lo indique de manera explícita. Teniendo esto en

cuenta, y sabiendo que estamos limitados por el diccionario, la propuesta es modificar la definición de las clases de los objetos urbanos para que incluyan al tipo elemental como una restricción más. Siguiendo con el ejemplo de *Bâtiment*, la nueva definición quedaría de esta forma:

$$\text{Pavillon} \Leftrightarrow \left\{ \begin{array}{l}
 \textit{Bâtiment} \\
 \wedge \\
 \text{hasResolution some } \textit{THR1} \\
 \wedge \\
 \text{isAdjacentTo some } (\textit{Route\_Autre} \vee \textit{Vegetation}) \\
 \wedge \\
 \text{isIncludedIn some } (\textit{TU\_Discontinu} \vee \textit{TU\_Discontinu\_Individuel}) \\
 \wedge \\
 \text{hasSurface some } \textit{Quantite\_Petit} \\
 \wedge \\
 \text{hasLongueur some } \textit{Quantite\_Petit} \\
 \wedge \\
 \text{hasLargeur some } \textit{Quantite\_Petit} \\
 \wedge \\
 \text{hasElongation some } \textit{Quantite\_Petit} \\
 \wedge \\
 \text{hasForme some } (\textit{Forme\_Carre} \vee \textit{Forme\_Rectangulaire})
 \end{array} \right.$$

Esta nueva restricción refuerza la idea de que sólo un objeto de tipo *Bâtiment* puede ser clasificado como *Pavillon* si y sólo si cumple con todas estas restricciones al mismo tiempo. Sin embargo, la desventaja principal de esta alternativa es que se necesita conocer el tipo elemental de antemano para poder clasificar al objeto correctamente. Luego con la información adicional sólo se podrá clasificar al objeto en alguna de las subclases de *Bâtiment*, o en el peor de los casos, no podrá clasificarse y continuará siendo un *Bâtiment*.

### Objetos identificables en múltiples resoluciones

Diferentes características son apreciadas a diferentes resoluciones, es por esto que la resolución es un factor clave cuando de clasificar objetos se trata. Por ejemplo, al observar la descripción del objeto *Bassin Artificiel* figura 2.17, se puede apreciar que el mismo es identificable bajo las resoluciones THR1, HR1 y HR2.

**Fiche 16 : Bassin artificiel**

**A. Identification de l'objet**

Type  Polygone	Nom de l'objet : <b>Bassin artificiel</b>	Type d'objet élémentaire <b>Eau</b>	Résolution <b>THR1 – HR1 – HR2</b>
---	--	--	---------------------------------------

**B. Description de l'objet dans le monde réel**

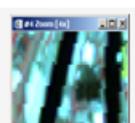
**B.1 Définition textuelle**

L'objet « bassin artificiel » appartient à la catégorie d'objets élémentaires « eau ».  
 Il désigne la partie d'un canal (incluse dans un port) réservée à l'ancrage des bateaux, limitée par des quais ou des jetées. Il peut être à flot et éventuellement fermé par une écluse. (p 338)  
 Il regroupe les bassins portuaires, bassins de rétention, réserves artificielles, retenues hydro-électriques...

En général les « bassins artificiels » :

- appartiennent à un ensemble fonctionnel de type « zone portuaire » correspondant à une zone industrielle (fiche 29)
- sont bordés de bâtiments liés à l'activité ou de zones industrielles
- si plus d'un alors parallélisme

**B.2 Illustration graphique :**

	THR1	HR1 (15.m)	HR2 (30.m)
- <i>bassin artificiel</i>			

**C.3. Principales relations**

	THR1	HR1	HR2
Adjacence	Objets de type bâtiment de l'activité	Forêt, zone industrielle, TU ind, TU collectif	Forêt, zone industrielle, TU discontinu

**C.4. Attributs**

	THR1	HR1	HR2
Longueur ou diamètre (m)	/	/	/
Largeur (m)	Moyenne à élevée	Moyenne à élevée	Moyenne à élevée
Elongation (m)	forte	forte	forte
Surface (m <sup>2</sup> )	/	/	/
Indice de forme	Rectangle	Rectangle	Rectangle
Texture (variance)	hétérogène	homogène	homogène

Figura 2.17: Ficha para el objeto *Bassin Artificiel*

Observando la tabla de sus atributos y relaciones, podemos distinguir que sus valores y objetos relacionados varían de acuerdo a la resolución a la que se está observando el objeto. Teniendo esto en cuenta, si consideramos que la descripción de un objeto se compone de pequeños bloques de información, como por ejemplo, el bloque formado por las restricciones que están caracterizadas bajo la resolución THR1, podríamos describir a los objetos como una combinación adecuada de estos bloques. Nuestra propuesta, ilustrada en la figura 2.18, se detalla a continuación:

1. Por cada resolución a la que el objeto es identificable, se creará un bloque con las restricciones que ocurren en esa resolución.
2. Extraer todas las restricciones comunes a los bloques de resolución y crear un nuevo bloque  $C$  con estas.
3. Crear un bloque  $R$  formado por la disjunción de los bloques de resolución. De esta forma, cuando se satisfacen las restricciones de al menos uno de los bloques de resolución, se satisface el bloque  $R$ .
4. Poner en conjunción el bloque  $R$  con el bloque  $C$ , para que así se cumplan todas las propiedades y relaciones dependientes de una determinada resolución.

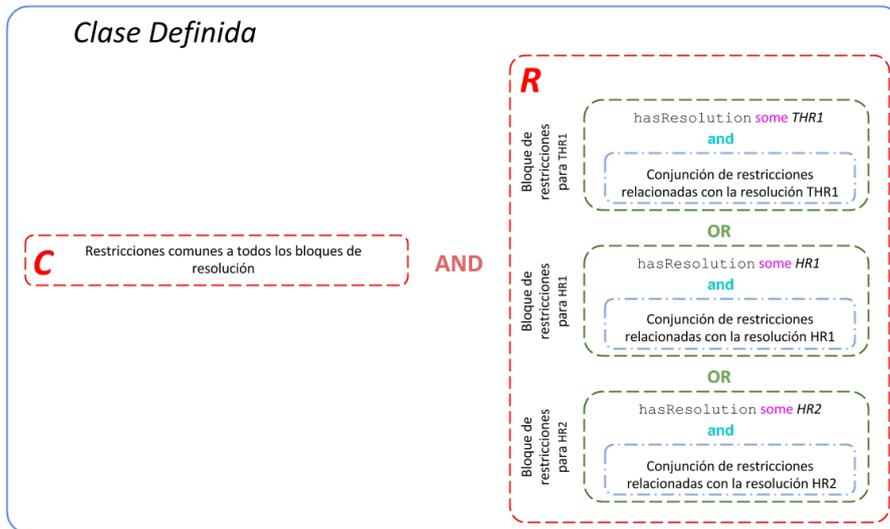


Figura 2.18: Propuesta para objetos identificables en múltiples resoluciones

Con esta nueva definición, para que un individuo sea miembro de una clase determinada se deben satisfacer las restricciones del bloque común  $C$  y las restricciones de al menos de uno de los bloques de resolución (bloques verdes) del bloque  $R$ . Dado que un individuo sólo puede tener un único valor de resolución en un momento dado, el bloque

que se satisface será el único. Sin embargo, si la resolución no se especifica en el objeto, el individuo no podrá ser inferido como miembro de alguna clase, ya que la resolución es una de las condiciones necesarias para establecer la membresía. Por otro lado, puede ocurrir que las restricciones en un objeto para una resolución sean similares a las restricciones impuestas en otro objeto para otra resolución. Por ejemplo, podría ocurrir que existiera un objeto identificable a resolución HR2 que tuviera las mismas restricciones que *Bassin Artificiel* tiene en THR1. Esto no significa que estos dos objetos con las mismas restricciones pero observables a diferentes resoluciones sean equivalentes. De hecho esto expresa que lo contrario, ya que para que sean semánticamente equivalentes deben cumplirse todas las restricciones impuestas para una misma resolución. Con estas modificaciones podemos describir a la clase *Bassin Artificiel* de la siguiente forma:

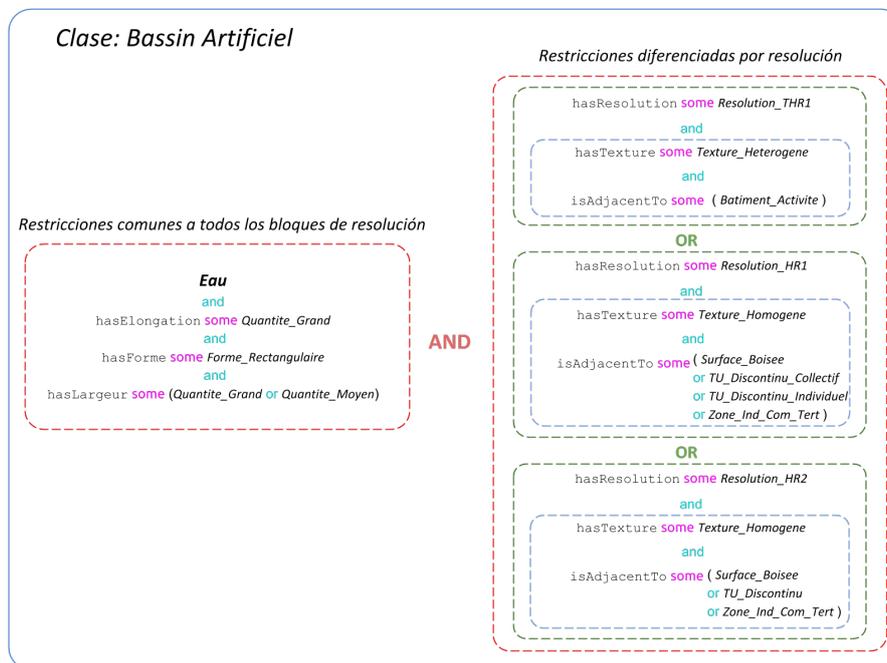


Figura 2.19: Clase *Bassin* definida en bloques

### Definiendo super clases para objetos con restricciones en común

Muchas veces se tratan de identificar objetos cuya información es insuficiente para que puedan ser clasificados en alguna de las clases de objetos urbanos definidos por el diccionario. Si pudiéramos reducir las restricciones de las clases de alguna forma sin perder información, podríamos abarcar estos casos y aumentar así el grado de éxito de las clasificaciones. Aplicando el mismo razonamiento que se utilizó para extraer las restricciones comunes de un objeto observable a diferentes niveles de resolución, podemos tomar obje-

tos de un mismo tipo que tengan restricciones en común, definir una nueva clase con esas restricciones y convertirla en superclase de esos objetos. Con esta acción lo que buscamos es poder clasificar a un objeto cuya información sea suficiente para satisfacer las mínimas restricciones impuestas por estas nuevas clases, pero insuficiente para poder ser clasificado bajo alguna de sus subclasses. Sin embargo hacer esto no siempre es posible, ya que se podrían crear clases que resulten demasiado generales, las cuales distorsionarían el comportamiento de la ontología, obteniendo resultados no deseados. Por lo tanto, como regla general, sólo se extraerán las restricciones en común de los objetos en aquellas situaciones que tenga sentido crear una nueva clase, o modificar la jerarquía original.

Para mostrar esta proposición, tomemos como ejemplo a los objetos *Groupe d'Arbres* (agrupamiento de árboles) y *Alignement d'Arbres* (alineamiento de árboles), los cuales son muy similares en su definición. Estos objetos son identificables bajo resolución THR1, tienen los mismos objetos adyacentes, están incluidos en objetos de tipo *Parc* y tienen una textura heterogénea. Aplicando lo expuesto anteriormente, podemos extraer esta información en común y escribir una nueva clase especialmente creada, la cual será superclase llamada *Ensemble d'Arbres* (conjunto de árboles) de las clases *Groupe d'Arbres* y *Alignement d'Arbres*. Por último, se agregará una restricción del mismo tipo que se utilizó anteriormente para restringir la clasificación a objetos de un mismo tipo elemental. Estas modificaciones de las clases de *Groupe d'Arbres* y *Alignement d'Arbres* pueden apreciarse en la figura 2.20.

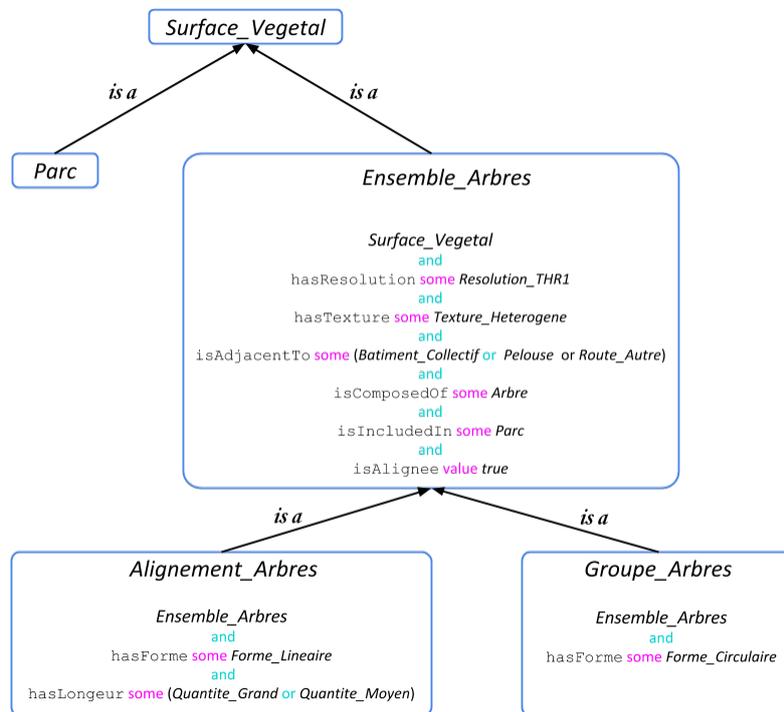


Figura 2.20: Modificación de la taxonomía de *Surface Vegetal*

De esta manera, si un individuo tuviera información suficiente para satisfacer las restricciones impuestas por *Ensemble d'Arbres*, pero a su vez esta información no alcanzara para satisfacer las restricciones de alguna de sus subclases, sería clasificado como miembro de *Ensemble d'Arbres*. Antes de esta modificación, la clasificación hubiera tenido lugar en un nivel mucho más alto del árbol de jerarquía de clases o incluso en la raíz del mismo, por no tener una manera de utilizar la escasa pero valiosa información del objeto. Con esta nueva metodología se intenta aprovechar toda la información presente para intentar clasificar en el nivel más profundo posible de la jerarquía.

Algo similar podemos hacer con los objetos que poseen variantes de si mismos. Por ejemplo, el objeto *Parc*, ilustrado en la figura 2.21, tiene diferentes variantes de acuerdo a su superficie, la cual puede ser pequeña, mediana o grande.

### Fiche 28 : Parc

#### A. Identification de l'objet

Type	Nom de l'objet :	Type d'objet élémentaire	Résolution
 Polygone	<i>Parc</i>	<i>Surface végétale</i>	THR1

#### B. Description de l'objet dans le monde réel

##### B.1 Définition textuelle

L'objet « *parc* » est un objet construit constitué de plusieurs objets élémentaires (arbre, groupement d'arbre, chemin, plan d'eau, etc).

Il désigne un espace vert généralement enclos et de grandes dimensions, largement planté d'arbres.

Dans le parc dit « urbain » ou « péri-urbain », on peut distinguer 3 niveaux d'importance selon leur dimension :

- *le parc d'agglomération* : de l'ordre de la centaine d'hectares ou plus  
Il est en général un grand espace à caractère végétal très marqué, où les installations spécifiques mobilisent moins de la moitié du terrain.
- *le jardin public* (à l'échelon d'une partie de grande ville ou d'une petite ville) : de quelques hectares à la dizaine d'hectares ou plus  
Il s'organise très souvent autour d'un plan d'eau central ou d'un massif boisé préexistant. Composés en majorité d'espaces créés (pelouses, boisements, aires de jeux), il offre un lieu de repos, de détente et de pratique d'activités récréatives. On peut également y trouver diverses installations de loisirs telle que théâtre, jardin botanique.
- *le square* (à l'échelon du quartier ou d'un groupe d'immeubles d'habitation) : de un à quelques hectares ou moins  
Il correspond à une place comprenant un jardin public central.

Il est exclu de cette définition le parc naturel qui est une vaste étendue de territoire à l'intérieur de laquelle la faune, la flore et le milieu naturel en général sont protégés de l'action destructrice de l'homme.

Figura 2.21: Descripción del objeto *Parc*

Aplicando la metodología para extraer las restricciones en común, pero a la inversa, se agregan tres nuevas subclases descendientes de *Parc*, llamadas *Parc\_Carre*, *Parc\_Public* y

*Parc\_Agglom*, las cuales se definen utilizando restricciones sobre la superficie dependiendo del tamaño.

Las restricciones comunes a todas las variedades de parques quedan comprendidas en *Parc* y sus variantes más específicas en las subclases. Por ende, si a un objeto de tipo *Parc* se le especifica el tamaño de superficie, se puede conocer con exactitud de que subtipo de parque se trata, situación que antes no era posible ya que esta distinción no existía. El resultado de esta modificación para la clase y la definición de las nuevas subclases puede verse en la figura 2.22.

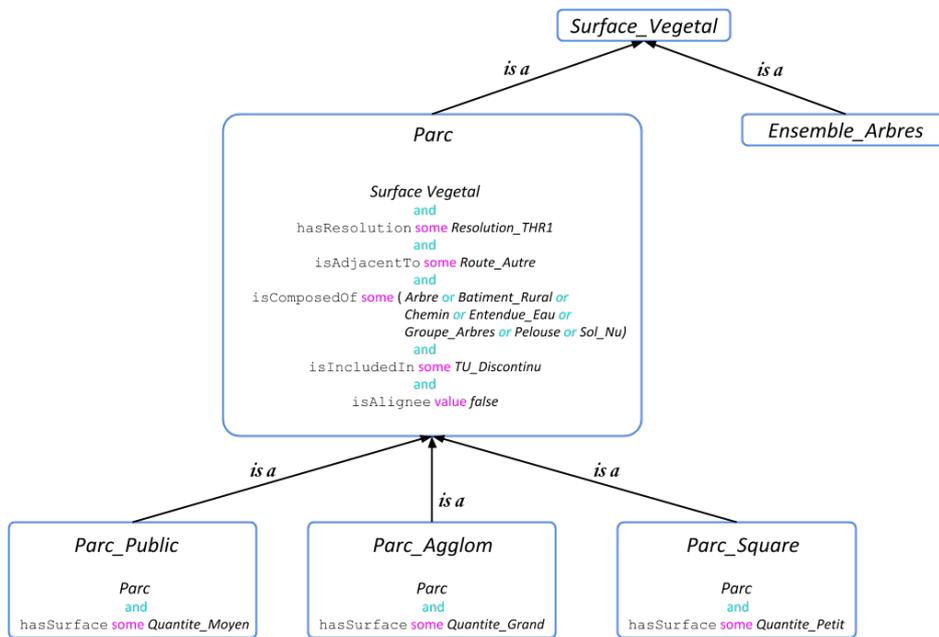


Figura 2.22: Separación de la clase *Parc* en tres subclases

Con estas modificaciones sobre la ontología la jerarquía de objetos urbanos toma su forma final, la cual es representada en dos partes para su mejor visualización. En la figura 2.23 se puede apreciar la taxonomía final que respecta a la rama del árbol perteneciente a los objetos simples. La rama correspondiente a los objetos compuestos puede verse en la figura 2.24. Notar como los cambios en objetos como *Parc*, y la creación de otros objetos, como *Ensamble d'Arbres*, han impactado sobre la jerarquía original.

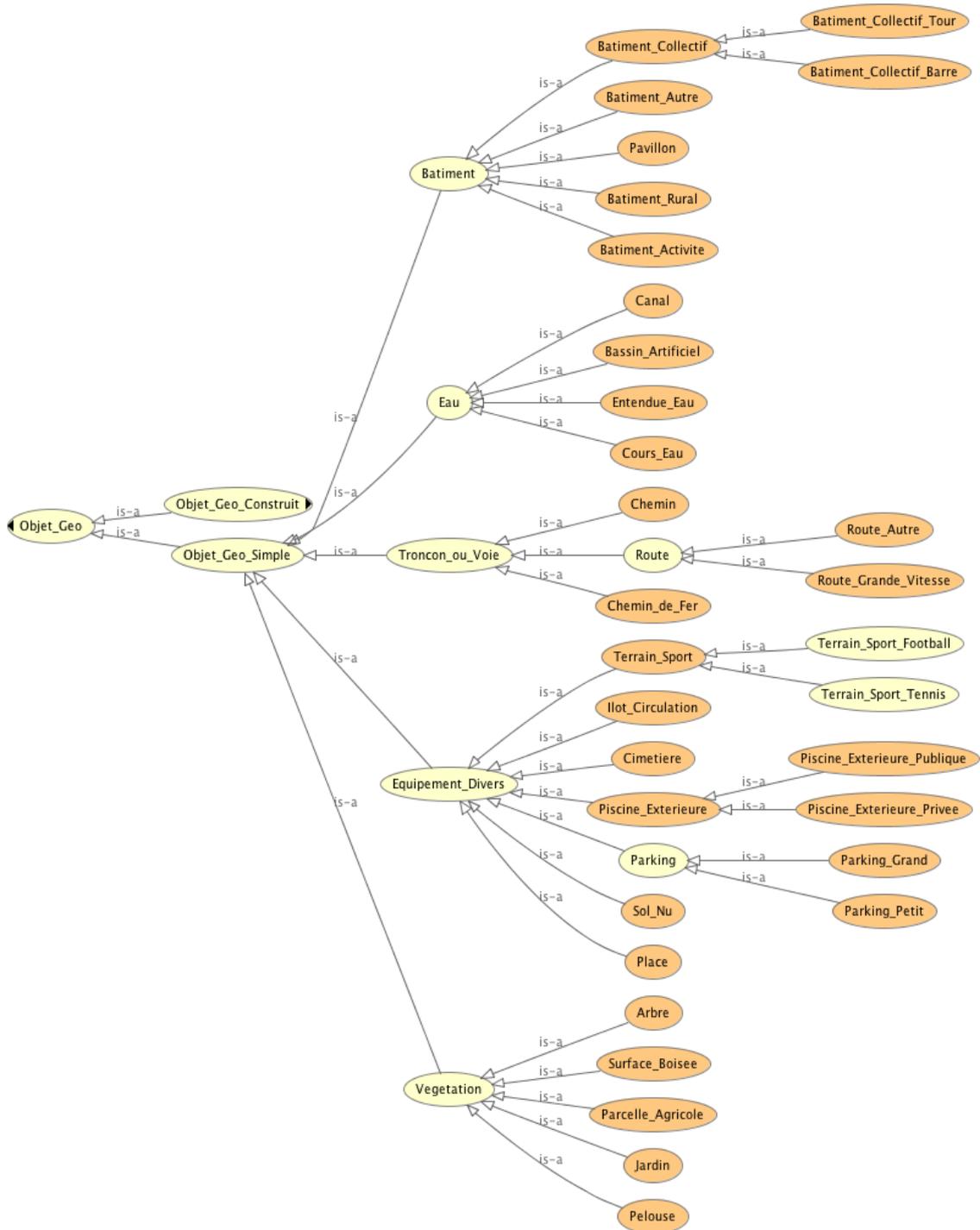


Figura 2.23: Taxonomía final para objetos simples

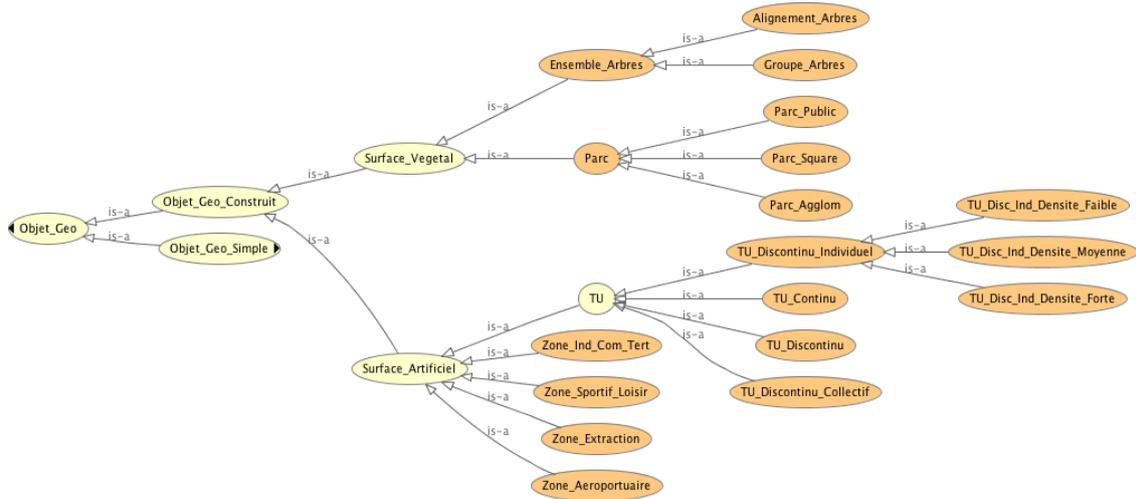


Figura 2.24: Taxonomía final para objetos compuestos

Esta ontología, aunque funcional en la teoría, no es práctica para los objetivos que fueron planteados en la introducción de este documento. Para que expertos urbanistas y geógrafos puedan darle una verdadera utilidad, es necesario primero, utilizar las mismas teorías que estos aplican para el estudio de objetos espaciales, y en segundo lugar, ocultar la ontología con alguna interfaz amigable que permita interactuar con esta de manera transparente, sencilla y segura. En los próximos capítulos trataremos estos temas.

## Capítulo 3

# Relaciones espaciales aplicadas a la ontología

En nuestra ontología, las clases que representan a los objetos urbanos derivan de la conceptualización del diccionario definido por los expertos geógrafos. Sin embargo, en ningún momento estamos teniendo en cuenta la información actual de la imagen, como lo haría un software de reconocimiento de imágenes. Las *relaciones espaciales* entre objetos es un dato que puede extraerse directamente de la imagen realizando un análisis topográfico. Para poder utilizar esta información en la ontología, necesitamos conceptualizar las relaciones espaciales.

Cuando hablamos de relaciones espaciales nos referimos a aquellas relaciones binarias utilizadas para caracterizar como un objeto se localiza en el espacio respecto a algún otro objeto de referencia. Las relaciones espaciales topológicas, pueden caracterizar, por ejemplo, la superposición de objetos y la disjunción de objetos, entre otras situaciones.

### 3.1. El modelo RCC

El razonamiento espacial cualitativo trata sobre los aspectos cualitativos para la representación y el razonamiento sobre entidades espaciales. Su principal objetivo es proporcionar un cálculo que permita a las máquinas representar y razonar con entidades espaciales, sin tener que recurrir a las técnicas cuantitativas tradicionales.

El modelo denominado *Region Connection Calculus* (**RCC**), introducido por [23], tuvo la intención de proporcionar un *framework* lógico para incorporar razonamiento espacial a sistemas basados en inteligencia artificial. El modelo RCC define un conjunto de relaciones binarias básicas, suficientes para describir las regiones presentes en un espacio topológico mediante las relaciones entre estas. Un subconjunto de RCC, ampliamente estudiado y utilizado, conocido como **RCC8**, se compone de ocho de estas relaciones topológicas elementales. Dichas relaciones son binarias y se aplican a un par de regiones  $X$  e  $Y$  en un espacio  $n$ -dimensional. Las mismas son exhaustivas y mutuamente exclusivas, lo que significa que cualquier configuración de dos regiones espaciales puede ser descripta por

este conjunto, y si una de estas relaciones se satisface, entonces las demás son falsas. Las ocho relaciones que conforman el subconjunto RCC8 son:

- $EQ(x, y)$  :  $x$  es equivalente a  $y$ .
- $TPP(x, y)$  :  $x$  es parte tangencial propia de  $y$ .
- $TPP^{-1}(x, y)$  :  $y$  es parte tangencial propia de  $x$ .
- $NTPP(x, y)$  :  $x$  no es parte tangencial propia de  $y$ .
- $NTPP^{-1}(x, y)$  :  $y$  no es parte tangencial propia de  $x$ .
- $PO(x, y)$  :  $x$  se superpone parcialmente con  $y$ .
- $EC(x, y)$  :  $x$  está externamente conectado con  $y$ .
- $DC(x, y)$  :  $x$  está desconectado de  $y$ .

Estas relaciones se utilizan para describir las siguientes situaciones entre dos regiones:

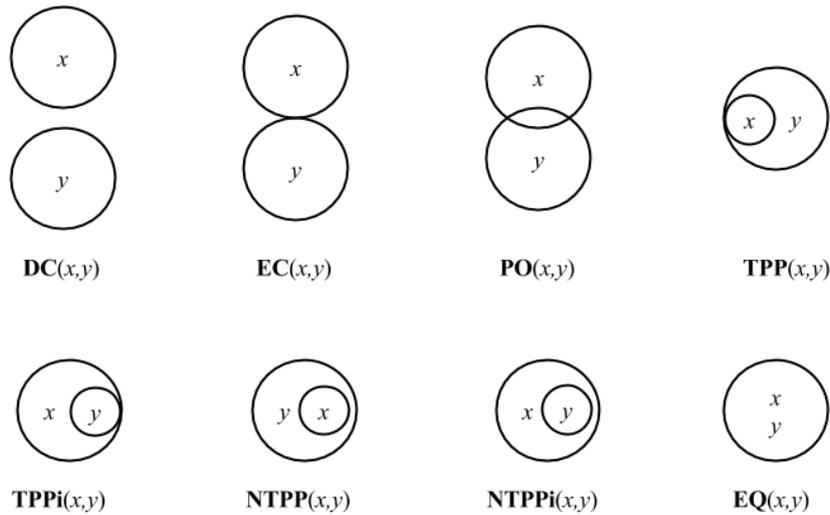


Figura 3.1: Representación gráfica de las relaciones RCC8

Como complemento, las siguientes operaciones matemáticas resultan útiles para realizar inferencias sobre relaciones topológicas:

- La relación *inversa* de una relación  $r$  es una relación  $r^{-1}$  tal que  $\forall x, \forall y, r(x, y) \Leftrightarrow r^{-1}(y, x)$ .
- Las relaciones  $r_1$  y  $r_2$  son *disjuntas* si  $\forall x, \forall y, r_1(x, y) \Rightarrow \neg r_2(x, y)$ .

- El *complemento* de una relation  $r$  es la relación  $rc$  tal que :  $r$  y  $rc$  son disjuntas y  $\forall x, \forall y, r(x, y) \vee rc(x, y)$  es verdadero.
- Dadas tres regiones espaciales  $x, y, z$ , y un par de relaciones  $r1$  y  $r2$ , la *composición* de  $r1(x, y)$  y  $r2(y, z)$  es la disjunción  $r(x, z)$  de todas las posibles relaciones entre  $x$  y  $z$ .

La relación de composición es de particular interés dado que permite inferir las posibles relaciones entre dos regiones  $x$  y  $z$  a partir de las relaciones que mantienen las regiones  $x$  e  $y$  por un lado, y las regiones  $y$  y  $z$  por el otro.

Las reglas de composición para las relaciones espaciales RCC8 se representan en la tabla de composición que puede consultarse en [24].

## 3.2. Implementación de relaciones RCC8 en OWL

Varios intentos de combinar razonamiento espacial y lógica de descripción (DL) han sido propuestos. Algunos definen a las relaciones espaciales como roles DL, sin embargo, se requiere de cierto nivel de expresividad para representar las inferencias del modelo RCC8 que excede a la capacidad expresiva de OWL sobre los roles (o properties), llevando en ocasiones a indecidibilidad. Los autores de [22] proponen un enfoque diferente del resto. Su proposición consiste en reificar las relaciones RCC8 y un conjunto de primitivas denominado **CM8**, como conceptos OWL, permitiendo de esta forma implementar conjunción, disjunción y negación de roles, lo cual antes no era posible con los enfoques convencionales. Debido a que esta representación impide caracterizar a las relaciones como simétricas y transitivas, lo cual es necesario para tratar con las inferencias del modelo RCC8, los autores proponen suplantarlo mediante el uso de reglas **SWRL**.

### 3.2.1. Reglas SWRL

El *Semantic Web Rule Language* ([25, 26]), **SWRL**, es un lenguaje expresivo basado en OWL. Permite a los usuarios escribir reglas en forma de cláusulas de *Horn*, las cuales pueden expresarse en términos de conceptos OWL para mejorar las capacidades de inferencia del lenguaje.

Las reglas SWRL se escriben como una implicación entre un antecedente y un consecuente. El significado que las reglas quieren expresar es el siguiente: cada vez que la condición especificada en el antecedente se satisface, la condición especificada en el consecuente también se cumple. Ambos, antecedente y consecuente, consisten en la conjunción de cero o más átomos, los cuales pueden tener la forma de  $C(x)$ ,  $P(x, y)$ ,  $sameAs(x, y)$  o  $differentFrom(x, y)$ , donde  $C$  es una clase OWL,  $P$  es una OWL property, y  $x$  e  $y$  pueden ser variables, individuos OWL o valores de datos OWL.

### 3.2.2. Primitivas CM8

Para extraer automáticamente la información topológica referida a las relaciones RCC8 de una imagen se necesitan métodos computables que puedan comprobar dichas relaciones entre las regiones presentes. Las primitivas CM8, desarrolladas por [27] pueden fácilmente ser calculadas mediante rutinas de procesamiento de imágenes, y permiten vincular la información topológica presente en una imagen, con modelos formales de relaciones topológicas, como por ejemplo RCC8.

Estas primitivas se basan en cuatro operaciones elementales que involucran a los pixeles de una imagen, más precisamente, al conjunto formado por los pixeles de la frontera de una región y al conjunto formado por los pixeles del interior de una región. Estas operaciones se componen de la siguiente manera: la intersección de los conjuntos interiores,  $x^\circ \cap y^\circ$ ; la intersección de los conjuntos de fronteras,  $\partial x \cap \partial y$ ; las dos diferencias de los conjuntos de interiores,  $x^\circ - y^\circ$ ,  $y^\circ - x^\circ$ .

En base a estas operaciones elementales se definieron las primitivas CM8 :

- $x^\circ - y^\circ = \emptyset$ ,  $x$  es parte de  $y$ , denotado como  $P(x, y)$ .
- $x^\circ - y^\circ \neq \emptyset$ ,  $x$  no es parte de  $y$ , denotado como  $NP(x, y)$ .
- $y^\circ - x^\circ = \emptyset$ ,  $x$  contiene a  $y$ , denotado como  $P^{-1}(x, y)$ .
- $y^\circ - x^\circ \neq \emptyset$ ,  $x$  no contiene a  $y$ , denotado como  $NP^{-1}(x, y)$ .
- $x^\circ \cap y^\circ = \emptyset$ ,  $x$  es discreto respecto a  $y$ , denotado como  $DR(x, y)$ .
- $x^\circ \cap y^\circ \neq \emptyset$ ,  $x$  se superpone a  $y$ , denotado como  $O(x, y)$ .
- $\partial x \cap \partial y = \emptyset$ ,  $x$  no comparte frontera con  $y$ , denotado como  $NA(x, y)$ .
- $\partial x \cap \partial y \neq \emptyset$ ,  $x$  comparte frontera con  $y$ , denotado como  $A(x, y)$ .

Estas primitivas pueden expresarse en términos de relaciones RCC8, y viceversa, según lo muestra la siguiente tabla:

	$P$	$NP$	$P^{-1}$	$NP^{-1}$	$DR$	$O$	$NA$	$A$
$EQ$	1	0	1	0	0	1	0	1
$NTPP$	1	0	0	1	0	1	1	0
$TPP$	1	0	0	1	0	1	0	1
$NTPP^{-1}$	0	1	1	0	0	1	1	0
$TPP^{-1}$	0	1	1	0	0	1	0	1
$PO$	0	1	0	1	0	1	0	1
$EC$	0	1	0	1	1	0	0	1
$DC$	0	1	0	1	1	0	1	0

Tabla 3.1: Equivalencias entre primitivas CM8 y relaciones RCC8.

Esta tabla puede interpretarse de la siguiente forma:

- Tomando una fila cualquiera, podemos obtener la relación RCC8 asociada, como la conjunción de aquellas primitivas CM8 que tengan un 1 como valor de celda en esa fila. Por ejemplo,  $EQ \Leftrightarrow P \wedge P^{-1} \wedge O \wedge A$ , significa que dos regiones  $x$  e  $y$  son equivalentes, si y sólo si,  $x$  es parte de  $y$ ,  $y$  es parte de  $x$ ,  $x$  se superpone a  $y$ , y la intersección entre los conjuntos de fronteras de  $x$  e  $y$  es distinto de vacío.
- Tomando una columna cualquiera, podemos obtener la primitiva CM8 asociada, como la disjunción de las relaciones RCC8 que tengan un 1 como valor de celda en esa columna. Por ejemplo,  $P \Leftrightarrow EQ \vee NTPP \vee TPP$ , significa que  $x$  es parte de  $y$ , si y sólo si,  $x$  es equivalente a  $y$ , o si  $x$  es parte tangencial propia de  $y$ , o si  $x$  no es parte tangencial propia de  $y$ .

### 3.2.3. Reificación de las relaciones RCC8

Para reificar las relaciones espaciales, los autores propusieron lo siguiente:

1. Representar cada una de las relaciones espaciales y primitivas como clases OWL.
2. Definir dos *object properties* funcionales, **from** y **to**, con sus respectivas inversas, **from**<sup>-1</sup> y **to**<sup>-1</sup>, para que las relaciones puedan efectivamente asociar dos objetos geográficos (figura 3.2). Esto se basa en que las relaciones RCC8 no son generalmente simétricas, y en que además, una relación espacial tiene lugar desde un primer objeto geográfico hacia un segundo objeto geográfico.

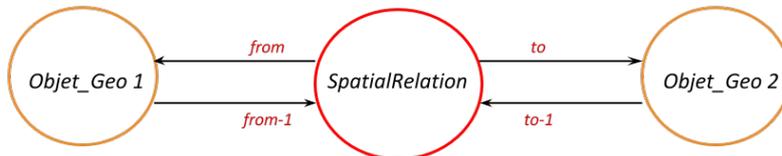


Figura 3.2: Reificación de relaciones espaciales

La clase *SpatialRelation* de la figura 3.2, representa un concepto superior que incluye a cualquier tipo de relación espacial, en particular, a las relaciones RCC8. La clase *RCC8* se definió como la unión disjunta de las clases que representan a estas relaciones, y a su vez, como equivalente a :

$$SpatialRelation \wedge (\mathbf{from\ some\ } Obj\_Geo) \wedge (\mathbf{to\ some\ } Obj\_Geo)$$

Las clases que representan las primitivas CM8 se definieron como una disjunción de relaciones RCC8. Por ejemplo, las clases que representan inclusión ( $P$ ) y superposición ( $O$ ), de acuerdo a la tabla 3.1, se definen de la siguiente manera:

$$P \equiv EQ \vee NTPP \vee TPP$$

$$O \equiv EQ \vee NTPP \vee TPP \vee NTPP^{-1} \vee TPP^{-1} \vee PO$$

### 3.2.4. Desventajas de la reificación de relaciones

La reificación de relaciones espaciales introduce una notación más compleja comparada a la versión no reificada<sup>1</sup>, la cual además de ser compleja de escribir, también lo es para leer e interpretar. Sin embargo, el proceso de convertir relaciones no reificadas en reificadas puede ser fácilmente automatizable de acuerdo a las reglas planteadas en [22].

Otra de las desventajas es la necesidad de crear un individuo adicional para representar cada relación entre pares de objetos. Teniendo esto en cuenta, y con el objetivo de hacer esta tarea más sencilla, en [22] se ha planteado un algoritmo que permite automatizar esta tarea.

La reificación de relaciones busca ganar expresividad para representar ciertas características que los roles OWL no pueden, como por ejemplo, la negación de relaciones. Sin embargo, lo que se gana por un lado se pierde por el otro, ya que luego de reificar no es posible representar inversa, simetría y transitividad. Para solucionar esto, se plantea el uso de reglas SWRL.

### 3.2.5. Utilizando reglas SWRL para extender el modelo

Para expresar la inversa de una relación RCC8 en lógica de primer orden, normalmente escribiríamos:

$$\forall x, \forall y, TPP(x, y) \Rightarrow TPP^{-1}(y, x)$$

Luego de la reificación, las relaciones espaciales se convierten en objetos del universo y la regla anterior se expande de la siguiente forma:

$$\forall r1, \forall r2, \forall x, \forall y, TPP(r1) \wedge from(r1, x) \wedge to(r1, y) \wedge from(r2, y) \wedge to(r2, x) \Rightarrow TPP^{-1}(r2)$$

Reemplazando  $TPP^{-1}$  por  $TPPi$ , el cual es el nombre que se le ha dado a esta clase, podemos escribir esta regla con la sintaxis de SWRL de la siguiente manera:

$$TPP(?r1), from(?r1, ?x), to(?r1, ?y), from(?r2, ?y), to(?r2, ?x) \Rightarrow TPPi(?r2)$$

El mismo principio se utiliza para representar la transitividad de relaciones. Dado que la composición puede verse como una especie de regla de transitividad entre diferentes relaciones, es posible utilizar el mismo método para su representación. Por ejemplo, el resultado de combinar  $PO$  con  $TPP$  puede ser  $TPP \vee NTPP \vee PO$ <sup>2</sup>. Esto significa que el resultado de esta composición puede ser cualquier combinación de estas clases.

<sup>1</sup>En este documento no se hará hincapié en esta notación, puesto que como se verá más adelante, no será utilizada. Para más detalles se recomienda la lectura de [22]

<sup>2</sup>Según la tabla de composición de [24].

Sin embargo, como las reglas SWRL se basan en cláusulas de *Horn*, el consecuente de una regla no puede estar formado por una disjunción de átomos. Tomando esto en cuenta, los autores de [22] propusieron representar estas disjunciones como conceptos OWL. Por lo tanto, cada vez que una disjunción aparezca en el consecuente de una regla SWRL se deberá crear una nueva clase OWL. Continuando con el ejemplo anterior, la disjunción en el consecuente de la regla sería reemplazada por la siguiente equivalencia:

$$TPP\_NTPP\_PO \equiv TPP \vee NTPP \vee PO$$

La generación de reglas que cubran todas las entradas de la tabla de composición es una tarea necesaria, pero tediosa. Es por esto que en [22] se describe un algoritmo para la generación sistemática de un conjunto minimal de reglas SWRL que asegure un completo razonamiento sobre la tabla de composición. De esta forma, las reglas de composición, junto a las reglas de simetría e inversa, completan un modelo integro para el razonamiento espacial cualitativo.

La figura 3.3, representa gráficamente el látice final de conceptos, que contiene las ocho relaciones RCC8, las primitivas CM8 y todos los conceptos intermedios generados por las reglas de composición. Cada concepto en el látice está formado por la conjunción de sus ascendientes y por la disjunción de sus descendientes.

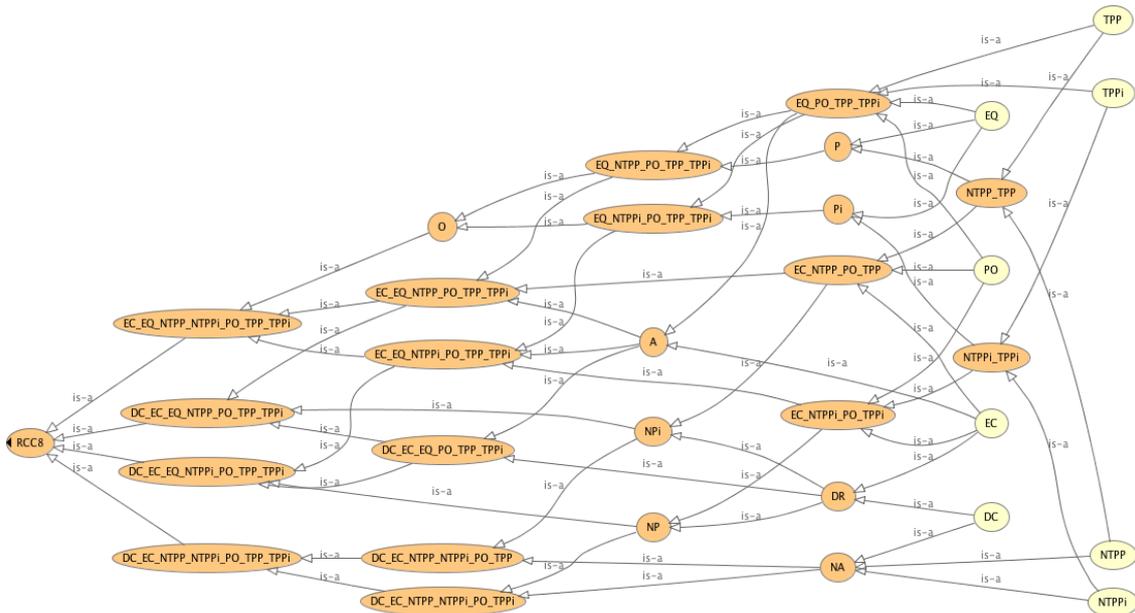


Figura 3.3: Látice de relaciones espaciales RCC8 con primitivas CM8

La ontología planteada en [22], demostró que es posible realizar una implementación de relaciones RCC8 basada en OWL y SWRL. Por otra parte, se ha mostrado que es posible expresar estas relaciones en términos de las primitivas CM8, las cuales permiten computar

las relaciones topológicas existentes entre los objetos de una imagen. Integrar este modelo con la ontología de objetos urbanos, es una tarea que deberá hacerse en dos etapas. La primera de estas se describirá en la siguiente sección y consistirá en la incorporación de los conceptos OWL que refieren a las relaciones RCC8, las primitivas CM8 y las reglas SWRL.

La segunda etapa, se tratará en el siguiente capítulo y consistirá en el desarrollo de una aplicación que funcionará como interfaz entre los usuarios y la ontología, y permitirá extraer automáticamente las primitivas CM8 existentes entre los objetos de una imagen.

### 3.3. Relaciones RCC8 en la ontología de objetos urbanos

Incorporar los conceptos de la ontología desarrollada en [22] a la ontología de objetos urbanos, es una tarea que puede ser realizada con **Protégé**. Mediante el asistente para realizar el *merge* de ontologías, **Protégé** proporciona una manera sencilla de mezclar ontologías, cuyo resultado es una nueva ontología que incluye tanto a las clases y propiedades, como a las reglas SWRL de las ontologías a unir. Por lo tanto consideramos apropiado utilizar dicha característica para este caso en particular.

Luego de la unión de ambas ontologías, el resultado es una nueva ontología que mantiene la taxonomía original de la ontología de objetos urbanos, pero que agrega las siguientes clases :

- Una clase por cada relación RCC8
- Una clase por cada primitiva CM8
- Una clase por cada entrada de la tabla de composición que sea el resultado de una conjunción de relaciones.
- La clase denominada *RCC8* definida en la sección 3.2.3.
- La clase denominada *SpatialRelation* definida en la sección 3.2.3.

Luego de esta unión, todas estas clases, a excepción de *RCC8* que se definió como subclase de *SpatialRelation*, se encuentran en el mismo nivel que *Obj\_Geo* ocupa en la jerarquía de clases. Para mayor comodidad y una mejor interpretación de esta nueva taxonomía, crearemos una clase *CM8*, como padre las clases correspondientes a las primitivas CM8, y como hijo de la clase *SpatialRelation* (figura 3.4), ya que como cada una de estas se definió como una disjunción de relaciones RCC8, son por lo tanto miembros de *SpatialRelation*. Las clases correspondientes a las relaciones RCC8 y las clases que se refieren a las composiciones entre estas, serán ordenadas como subclases de *RCC8*.

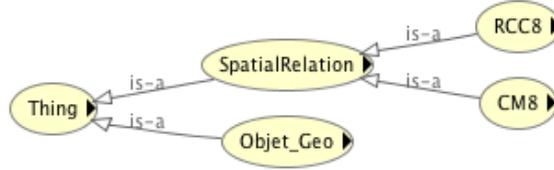


Figura 3.4: Taxonomía parcial de la unión de las ontologías

A pesar de esta organización, la taxonomía final del modelo de relaciones espaciales, o jerarquía de clases inferida por medio del razonamiento sobre las clases y sus definiciones, se sigue manteniendo igual al látice de la figura 3.3.

El último paso para completar la unión entre la ontología de objetos urbanos y la ontología de relaciones espaciales es adaptar la definición de los objetos urbanos a la notación reificada planteada en la sección anterior. Dado que el objetivo es realizar la menor cantidad de modificaciones posibles, cambiar la definición de las clases de objetos urbanos con la notación utilizada para describir relaciones reificadas no es una opción. En su lugar lo que proponemos es aprovechar las ventajas de las reglas SWRL y plantear reglas de equivalencia.

Si tomáramos la relación espacial RCC8, *EC* (conectado externamente), como el equivalente la propiedad **isAdjacentTo**, podríamos utilizar la siguiente regla SWRL para establecer esta equivalencia:

$$EC(?r1) \wedge from(?r1, ?x) \wedge to(?r1, ?y) \Rightarrow isAdjacentTo(?x, ?y)$$

La simetría de esta relación queda contemplada tanto en las reglas SWRL creadas por los autores de [22], como en la definición de la propiedad **isAdjacentTo** expresada en la ontología.

Para la relación de inclusión la situación es similar. Dado que para que un objeto esté incluido dentro de otro debe existir alguna de las relaciones RCC8, *EQ*, *NTPP* o *TPP* entre estos, la regla SWRL para establecer la equivalencia con nuestra *object property* **isIncludedIn** se torna un poco más compleja. En su lugar, podríamos utilizar la primitiva CM8, *P* (incluido en), la cual es equivalente a la disjunción de estas relaciones RCC8, para describir la regla SWRL de la siguiente manera:

$$P(?r1) \wedge from(?r1, ?x) \wedge to(?r1, ?y) \Rightarrow isIncludedIn(?x, ?y)$$

No es necesario escribir una regla para indicar la equivalencia entre  $P^{-1}$  (incluye a) y la propiedad **isComposedOf**, dado que como ya existe una regla que indica a  $P^{-1}$  como la inversa de *P*, y como **isComposedOf** se definió como la inversa de **isIncludedIn**, la equivalencia entre **isComposedOf** y  $P^{-1}$  se obtiene por inferencia<sup>3</sup>.

<sup>3</sup>Por limitaciones en el lenguaje, no es posible utilizar roles transitivos en una regla SWRL, ya que podrían ocurrir inferencias circulares. Esto obliga a quitar la característica transitiva de la propiedad **isIncludedIn**, sin embargo, la relación continuará siendo transitiva, ya que las reglas definidas por la tabla de composición así lo establecen.

De esta manera hemos logrado incorporar a nuestra ontología el modelo planteado en [22], el cual nos permite desde el uso de relaciones RCC8, obtener un mejor conocimiento sobre las relaciones espaciales de los objetos presentes en una imagen y en base a estas relaciones inferir nuevos resultados.

A continuación, veremos como desde un software diseñado especialmente para este trabajo, pueden obtenerse las relaciones RCC8 existentes entre los objetos de una imagen, mediante el cálculo automático de las primitivas CM8 correspondientes.

## Capítulo 4

# Una interfaz para la ontología de objetos urbanos

El uso de una ontología resulta incómodo sino impráctico para personas no afines al ámbito de la informática, como en este caso lo es para expertos geógrafos y urbanistas. Crear instancias de los objetos urbanos presentes en una imagen, ingresando la información necesaria de manera adecuada, y especificando las relaciones espaciales entre estos de manera reificada, es una tarea tediosa que consume mucho tiempo si la cantidad de objetos es elevada. Por estos motivos planteamos la necesidad de una interfaz que facilite la interacción de los usuarios con la ontología y que automatice la mayor cantidad de tareas posibles.

### 4.1. Tecnologías utilizadas

Para el desarrollo de la aplicación se optó por utilizar **Java** como lenguaje de programación. Esta elección se debió a varios motivos, entre los cuales se encuentran: la portabilidad que ofrece el lenguaje entre sistemas operativos, la practicidad para desarrollar interfaces gráficas, la amplia documentación y variedad de librerías e interfaces de programación, pero sobre todo, a la disponibilidad de API's<sup>1</sup> para OWL.

Para interactuar con la ontología desde Java, se necesita de alguna interfaz que permita leer, crear, manipular y visualizar ontologías OWL. De entre todas las API's de OWL que existen, se eligió utilizar la **OWL API v3.5.0** [28], debido a que posee una interfaz sencilla de utilizar, buena documentación con ejemplos de uso y soporta una amplia variedad de razonadores lógicos.

En cuanto a los razonadores lógicos, cabe mencionar que no todos soportan el uso de reglas SWRL. Durante el desarrollo de la ontología, fueron utilizados los razonadores **Pellet** y **Hermitt** que vienen incorporados en **Protégé v4.3.0**. Ambos razonadores proporcionan API's para Java compatibles con la interfaz **OWL API**. Luego de realizar

---

<sup>1</sup>API corresponde a las siglas en inglés de *Application Programming Interface* o Interfaz de Programación de Aplicaciones

diversas pruebas con las versiones de **Hermitt v1.3.8** y **Pellet v2.3.1**, se optó por el primero para ser utilizado como razonador de la aplicación, ya que a pesar de obtener los mismos resultados con ambas API's, el tiempo de ejecución durante las pruebas logrado por **Hermitt** fue siempre menor al de **Pellet**.

Para reconocer los objetos en una imagen, tal y como lo hiciera un software de reconocimiento especializado, la misma primero debe ser segmentada en regiones que delimiten a estos objetos. Dichas regiones pueden verse como figuras en el plano, y por ende, ser manipuladas de manera geométrica. Para manipular y realizar operaciones con estas figuras, necesarias para computar las primitivas CM8, se decidió utilizar la API **JTS v1.8** [29]. Esta librería *open source* proporciona una implementación de un modelo espacial<sup>2</sup> que cumple con lo especificado por la OGC<sup>3</sup>, una completa implementación de algoritmos espaciales fundamentales en dos dimensiones, métodos para análisis espacial y un conjunto de operaciones geométricas elementales.

Otra de las librerías utilizadas fue **GeoTools v12**, la cual se necesitó para manipular archivos de tipo *shapefile*, formato muy popular entre los sistemas de información geográfica, que almacena la localización de los elementos geográficos y los atributos asociados a ellos. Su uso en la aplicación se explicará más adelante.

## 4.2. Desarrollo

A continuación se describirá el desarrollo de los componentes más importantes de la aplicación. Para posterior referencia, la figura 4.1 muestra los diferentes paneles de la interfaz gráfica.

### Segmentación

Una de las principales funcionalidades de la aplicación es proporcionar al usuario la posibilidad de cargar una imagen satelital sobre la cual poder dibujar de manera interactiva las diferentes regiones de interés, simulando la segmentación que realiza un software de reconocimiento especializado. Para dibujar las geometrías que representan estas regiones, se consideró utilizar la clase "Polygon" de Java, cuyas instancias pueden ser representadas gráficamente de manera simple con métodos nativos de Java. Estos polígonos se componen de puntos, los cuales representan coordenadas en el plano, por lo tanto, para dibujar polígonos sobre una imagen, deberemos marcar puntos delimitando a los objetos de interés. Para tener una mejor percepción de lo que se está dibujando, la construcción de los polígonos será mediante segmentos, los cuales se dibujarán automáticamente entre cada último par de puntos marcados. Para finalizar un polígono, se proporciona una opción en el menú desplegable que permite al usuario cerrar el polígono que se está dibujando de manera automática, evitando así polígonos abiertos o errores en el dibujo que puedan producir futuras fallas en la detección de las primitivas CM8 (figura 4.2).

---

<sup>2</sup><http://www.vividsolutions.com/jts/discussion.htm>

<sup>3</sup>Open Geospatial Consortium. <http://www.opengeospatial.org/standards>

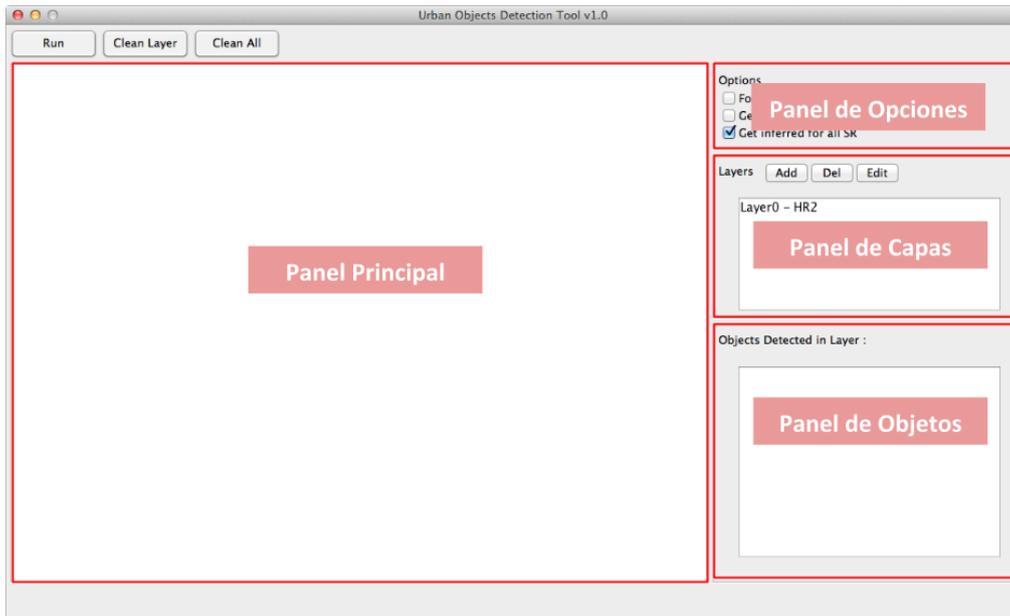


Figura 4.1: Paneles de la aplicación

Para iniciar una segmentación en una imagen ráster (o imagen de mapa de bits), se deberá cargar primero, dicha imagen en la aplicación. Mediante la opción “Open Image” del menú “File”, puede seleccionarse la ruta en la que se encuentra la imagen y la resolución de la misma. Una vez terminado este paso, la imagen será mostrada gráficamente en el panel principal y estará disponible para comenzar con el dibujado de polígonos.



Figura 4.2: Dibujado de polígonos

Como funcionalidades adicionales para el dibujo de polígonos tenemos:

- Un mecanismo para generar polígonos adyacentes de manera automática. Dado que dos polígonos se consideran adyacentes si comparten sus fronteras, cuando se intente marcar uno de los puntos de un polígono en proceso, muy cercano a la frontera de otro, con la intención de hacerlo adyacente, este mecanismo hará que ese punto que se marque pertenezca a la frontera de ambos.
- Deshacer segmento dibujado.
- Eliminar polígonos.

### Representación de los objetos urbanos en Java

Dijimos que para identificar los objetos urbanos presentes en una imagen, primero se debía segmentar en regiones, las cuales deben contener o representar un único objeto urbano. La clase “Polygon” de Java utilizada para dibujar estas regiones no proporciona las herramientas necesarias para calcular las primitivas CM8. Como se mencionó en la sección anterior, para computar efectivamente estas primitivas, nos apoyaremos en la librería **JTS**. Para utilizar los diferentes métodos de análisis espacial que nos proporciona esta librería, debemos escribir nuestros polígonos en términos de objetos espaciales compatibles. La clase “Geometry” de **JTS** permite representar polígonos, líneas, puntos, anillos, colecciones de geometrías, etc, por lo tanto se adapta perfectamente a nuestro modelo espacial. Sin embargo, no es representable gráficamente, o al menos no de una forma tan directa como lo es “Polygon”. Para utilizar las ventajas de la clase “Geometry” y no perder la capacidad de representación gráfica de la clase “Polygon”, se decidió crear una clase contenedora denominada “ObjGeom” que contiene dos instancias de polígonos creadas con el mismo conjunto de puntos, una de tipo “Polygon”, que es la que se utiliza para dibujar, y otra de tipo “Geometry”, utilizada durante el análisis espacial. Esta clase además contiene una variable para denotar al tipo de objeto urbano que representa, que inicialmente es *Object\_Geo*, pero que al finalizar el proceso contendrá el tipo correcto, una variable entera que sirve para identificar el objeto, y variables para todos los posibles atributos que un objeto urbano puede tener, como ser, resolución, longitud, amplitud, superficie, etc. Es decir, todos aquellos parámetros que fueron definidos en la ontología como *datatype properties*. Las relaciones con otras regiones/objetos, no son guardadas en la clase ya que se tratarán de manera diferente.

Así, cada vez que una región es dibujada, un objeto de tipo “ObjGeom” es creado, y con éste se crean sus correspondientes objetos de tipo “Geometry” y “Polygon”, y se inicializan todos sus parámetros como nulos para su posterior configuración.

### Capas de resolución

Durante el desarrollo de la ontología se hizo hincapié en la importancia de la resolución para la detección de objetos. Dado que una imagen se caracteriza en primera medida por su

resolución, todos los objetos que puedan identificarse en esta dependen de este parámetro. Es por esto que consideramos primordial establecer de antemano el tipo de resolución de la imagen que se cargará para la segmentación, ante la imposibilidad de detectar automáticamente este parámetro. Con esta información disponible desde el comienzo, es posible establecer automáticamente el parámetro de resolución de los objetos de tipo “ObjGeom” que son creados con cada región dibujada.

En ocasiones es necesario trabajar con una región observada a diferentes resoluciones o con diferentes grados de la misma resolución, lo que implica que se debe utilizar una imagen en cada caso. Esta situación puede ocurrir, por ejemplo, cuando se desea conocer con detalle los objetos que componen a un objeto compuesto, y una sola imagen no es suficiente para observar todo. Ante tal situación, consideramos que es de utilidad disponer de algún mecanismo que permita fácilmente manejar las diferentes imágenes de la región en estudio. Teniendo esto en cuenta, nuestra propuesta consiste en utilizar capas para manejar las diferentes resoluciones o vistas de una región, de manera que cada una tenga su propia imagen y segmentación. Con esta nueva funcionalidad, se permite que desde el panel de capas, un usuario pueda editar, eliminar y crear tantas capas como necesite, de la resolución que desee. Por cuestiones de simplicidad, sólo se permite que una capa esté activa a la vez, la cual es determinada desde la interfaz por el usuario en el momento de trabajar sobre esta. Cada vez que se realiza un cambio de capa, todas las regiones de la capa anterior se ocultan y las regiones de la capa que se convierte en activa se muestran.

Para implementar el uso de capas, se creó una clase denominada “Layer”, la cual contiene, entre otras cosas, el tipo de resolución, un nombre representativo y una lista de objetos de tipo “ObjGeom”, con lo cual cada capa tiene sólo acceso a sus objetos. Cada vez que una capa es creada, se crea un objeto de tipo “Layer” en la clase principal de la aplicación, y con esta, una lista de objetos vacía que será llenada a medida que se vayan dibujando regiones. Considerando que cada capa tiene acceso a sus objetos, en esta clase se definieron, con ayuda de la librería **JTS**, las operaciones fundamentales definidas en la sección 3.2.2 para obtener las primitivas CM8. Esta clase “Layer” tiene como función secundaria, reunir toda la información de cada objeto presente en la capa y de las primitivas que lo involucran, para luego, mediante métodos especializados que serán definidos en una clase que se verá más adelante, poder generar las instancias requeridas en la ontología para inferir las relaciones RCC8 entre estos y las clase de membresía. No debemos olvidar que todo el razonamiento espacial se lleva a cabo en la ontología.

### **Instanciación de objetos urbanos en la ontología**

Para crear instancias de los objetos de interés, se necesita ingresar toda la información posible de cada uno de estos en la ontología. Para lograr esto desde la interfaz, se proporciona un formulario (figura 4.3) para ingresar las características de los polígonos o regiones, el cual puede accederse desde la opción “Settings” del menú desplegable, haciendo click derecho sobre un polígono en particular, o haciendo doble click sobre un ítem del listado de objetos dibujados en el panel de objetos. En este formulario se solicita completar: el tipo de objeto urbano, si se conociera, del polígono a instanciar, y todos los parámetros que

fueron definidos como variables en la clase “ObjGeom”, y que corresponden a los atributos descritos por las *datatype properties* en la ontología. Para evitar errores, los atributos serán completados seleccionando de un listado de posibles opciones aquellas deseadas.

Además de completar estos atributos, el formulario también permite establecer equivalencia con otros polígonos que por lo general se encuentran en diferentes capas, lo que más tarde se traducirá en una igualdad entre individuos OWL. Dado que el computo de relaciones y clases de objetos, se produce de manera aislada en cada capa, esta opción permite a aquellos objetos que son identificables en más de una resolución, y que por lo tanto se encuentran en más de una capa, relacionarse para denotar al mismo objeto urbano. Esto permite establecer un vínculo entre las capas, para así extender el razonamiento más allá de los límites de una capa.

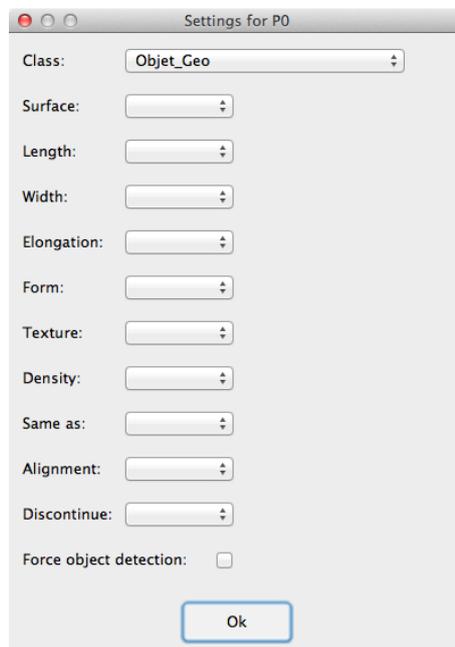


Figura 4.3: Formulario de características de un polígono

Esta opción es de utilidad, por ejemplo, cuando se tiene una capa, que llamaremos *Capa 1*, con varios objetos incluidos dentro de un objeto compuesto  $C$ , del cual se desconoce su tipo y características debido a que la resolución de la imagen no lo permite. Supongamos ahora que se desea conocer el tipo de objeto urbano de un objeto  $X$  incluido en  $C$ . Uno de los datos que se necesitará para inferir la membresía de  $X$ , es la inclusión, sin embargo, como el tipo de  $C$  se desconoce, la clasificación en este caso no tendrá éxito. Si tuviéramos otra capa, a la que llamaremos *Capa 2*, cuya resolución permitiera apreciar mejor las características y objetos adyacentes de  $C$ , sería posible inferir el tipo de  $C$ , lo que nos serviría para obtener el tipo del objeto  $X$ . Sin embargo, el objeto  $C$  de la capa *Capa 1*, es diferente del objeto  $C$  de la capa *Capa 2*, ya que pertenecen a diferentes capas. Por

lo tanto, el objeto  $C$  de la capa *Capa 2* no incluye a  $X$  y la inferencia del tipo de  $C$  no ayudará a obtener la membresía de  $X$ . A pesar de esto, lo anterior puede solucionarse si se hace explícita la igualdad entre los objetos  $C$  de ambas capas. Mediante el formulario de características de objetos, es posible seleccionar, de un listado de los polígonos presentes en todas las capas, aquel polígono que denota al mismo objeto urbano, con el cual se quiere establecer la igualdad.

### Conexión Java - OWL

Tanto las instancias de los objetos urbanos, como las instancias de las primitivas CM8, no pueden ser creadas ni utilizadas sino se disponen de las herramientas necesarias para ingresar esta información en la ontología. La clase *CM8toOWL*, creada con ayuda de la **OWL API**, contiene métodos para afirmar información en la ontología (crear instancias, definir relaciones, crear axiomas), obtener información afirmada y obtener información inferida.

Para crear instancias en la ontología de los polígonos dibujados, se utiliza la información contenida por los objetos “ObjGeom”, ingresada por medio del formulario descrito en el apartado anterior. La creación de estas instancias, se basa principalmente en utilizar métodos para la escritura de axiomas en la ontología. Algunos de estos afirman el tipo de objeto urbano, otros involucran al objeto en cuestión con otras instancias mediante *object properties*, como por ejemplo **hasSurface** con una instancia previamente creada de *Petit*, para especificar el tamaño de superficie; y otros utilizan *datatype properties* con valores de datos, para especificar alineación o discontinuidad. Para el caso de los axiomas que involucran instancias de clases cualitativas, se han creado en la ontología instancias de cada posible valor cualitativo, los cuales tienen un nombre representativo formado como “*i*” + valor cualitativo, como por ejemplo, *iTHR1*. Esto se implementó para no tener que crear un nuevo individuo cada vez que se necesite utilizar un determinado valor cualitativo.

Debido a que la resolución es un parámetro que está presente cada vez que se crea una instancia de “ObjGeom”, debemos tener cuidado en el momento de crear instancias en la ontología con axiomas que involucren a la resolución y ciertos objetos. Uno de los casos a considerar ocurre con los objetos que se definen como similares. Si estos objetos se encuentran en diferentes capas, se produciría una inconsistencia debido a que las resoluciones de ambos objetos, que ahora representan al mismo, son diferentes y disjuntas. Como buscamos automatizar todo el proceso de detección de objetos, se decidió que para aquellos objetos, cuyos atributos son distintos de nulo o que tienen tipo *Objet\_Geo*, deben existir un axioma que los involucre con la resolución. Esto se basa en la suposición de que aquellos objetos que tengan alguna información sobre sus atributos o que tengan un tipo genérico como *Objet\_Geo* son los que interesan ser identificados. Como puede haber casos excepcionales, se permite afirmar la resolución de todos los objetos desde la opción “Force detection for all objects” del panel de opciones, o de manera diferenciada en los objetos que se deseen, desde el formulario de características.

La reificación de relaciones RCC8 demanda la creación de individuos para su representación, utilizando los roles **from** y **to** como fue explicado anteriormente. Como aquí esta-

mos calculando las primitivas CM8, para posteriormente mediante razonamiento espacial en la ontología inferir las relaciones RCC8, debemos crear instancias de estas primitivas. De la misma forma que se construiría un individuo para una relación RCC8, se crearán dos individuos por cada relación entre dos objetos  $P1$  y  $P2$ , que se denominarán  $SR\_P1\_P2$  y  $SR\_P2\_P1$ , para indicar desde qué objeto está ocurriendo la relación. Es decir, si la relación ocurre desde  $P1$  a  $P2$ , el individuo a crear se llamará  $SR\_P1\_P2$ , y contendrá dos axiomas:  $SR\_P1\_P2$  **from**  $P1$  y  $SR\_P1\_P2$  **to**  $P2$ . El tipo de este individuo estará sujeto a las primitivas CM8 que se hayan calculado. Debemos tener en cuenta que por cada relación (en un sentido) entre dos objetos, se deben calcular cuatro primitivas. Por lo tanto, todo individuo que represente una relación espacial será miembro de cada una de las clases de estas primitivas calculadas durante el procedimiento. Luego, por medio del razonador, seremos capaces de inferir el tipo de relación RCC8 para esa relación espacial.

Por defecto, todos los individuos que se crean en la ontología, tanto objetos urbanos como relaciones espaciales, se declaran como diferentes entre sí mediante un axioma generalizado. Esto se hace para evitar situaciones indeseables, producto de la hipótesis de mundo abierto. Sólo en el caso mencionado anteriormente se hará explícita la igualdad entre individuos.

### Importar segmentación

Esta aplicación no ofrece todas las herramientas que un software profesional puede ofrecer para la segmentación de una imagen en regiones. Para obtener mejores resultados, la aplicación brinda soporte a archivos con formato *shapefile*, permitiendo así importar una segmentación.

Utilizando algún sistema de Información Geográfica, como por ejemplo **QGIS**<sup>4</sup>, es posible, mediante una imagen ráster previamente georreferenciada, crear una capa de polígonos sobre esta de manera sencilla, gracias a las diferentes herramientas que se proporcionan para esta tarea. Esta capa se almacena en un archivo *shapefile*, el cual contiene, entre otras cosas, las coordenadas de los puntos de los objetos espaciales. Para poder leer este formato de archivos se utilizó la librería **GeoTools**. Para que los polígonos del archivo *shapefile* puedan dibujarse en nuestra aplicación de manera correcta, se necesita contar con la información de georreferenciado. Esta información se guarda en un archivo de texto, denominado archivo de puntos, que contiene datos de correspondencia entre los píxeles de los cuatro extremos de la imagen y las coordenadas correspondientes. Mediante la opción “Import Segmentation” del menú “File” (figura 4.4), un usuario puede cargar un archivo *shapefile*. La nueva ventana emergente permite establecer la ruta del archivo *shapefile*, la ruta de la imagen que será mostrada como capa en la aplicación y la ruta del archivo de puntos. Con todos estos datos y por medio de una transformación de coordenadas en píxeles, se podrán dibujar los polígonos en su ubicación correspondiente.

---

<sup>4</sup>Sistema de Información Geográfica libre y de Código Abierto. <http://www.qgis.org/es/site/>

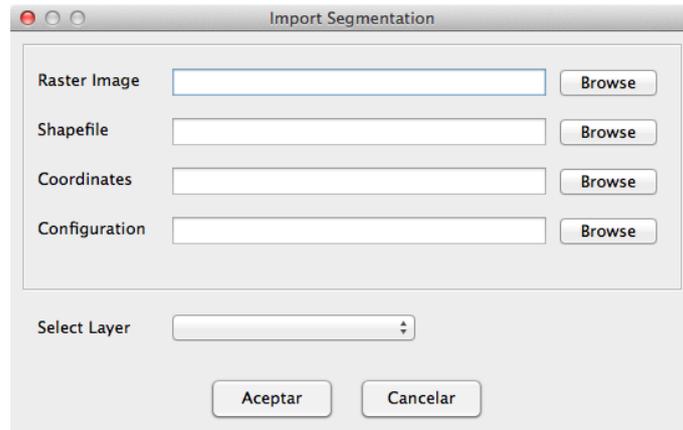


Figura 4.4: Ventana de importación de segmentación

Opcionalmente también puede cargarse un archivo de configuración que contiene información guardada sobre las características de los objetos dibujados en una determinada capa. Este archivo puede generarse mediante la opción “Save” del menú “File”, en cualquier capa, siempre y cuando haya objetos presentes.

## Resultados

Los resultados arrojados luego de la ejecución de la aplicación pueden observarse de dos maneras diferentes. La forma más directa, se encuentra en el panel de objetos de la capa activa. El listado de objetos en este panel, al inicio contiene los polígonos de la capa activa junto con los tipos de objetos urbanos afirmados por el usuario. Luego de que la clasificación es llevada a cabo, los tipos de los polígonos son actualizados automáticamente con la información inferida por el razonador. Como el razonador obtiene todas las clases de pertenencia del objeto, incluyendo subclases y superclases, se decidió sólo mostrar la clase más concreta de todas estas.

La otra manera de obtener resultados es mediante la opción “Show Results” del menú “Debug”. Esta ventana emergente contiene información detallada, diferenciada por capas, de cada polígono instanciado y cada relación espacial instanciada. Dado un polígono cualquiera, la información que se puede obtener corresponde a la información afirmada sobre ese polígono, a las primitivas CM8 que lo relacionan con los otros polígonos de la capa y a la información inferida sobre ese polígono, que incluye a todas las clases de pertenencia.

Para las relaciones espaciales, es posible observar la información afirmada sobre las relaciones espaciales, la cual involucra a las primitivas utilizadas y los objetos que relaciona. Como información inferida, tenemos el tipo de relación RCC8 que se obtuvo del razonamiento con esas primitivas, y como información adicional, el resultado de la composición de esas primitivas en diferentes combinaciones.

Dado que tanto la inferencia de objetos espaciales como de relaciones espaciales, consume demasiados recursos si la cantidad de polígonos es elevada, para agilizar el proceso

de clasificación, se proporciona desde el panel de opciones, una opción denominada “Get inferred data for all objects”, para habilitar o deshabilitar la inferencia para todos los objetos. Del mismo modo, para la inferencia de todas las relaciones espaciales existentes, la opción “Get inferred data for all SR”, cumple con la misma función. Cuando la opción de inferencia de objetos se encuentra deshabilitada, como lo es por defecto, sólo se inferirá nueva información en aquellos objetos cuyos atributos han sido especificados o cuyo tipo de objeto sea *Objet\_Geo*.

## Ejecución

A continuación se muestra el diagrama de flujo de una ejecución típica de la aplicación.

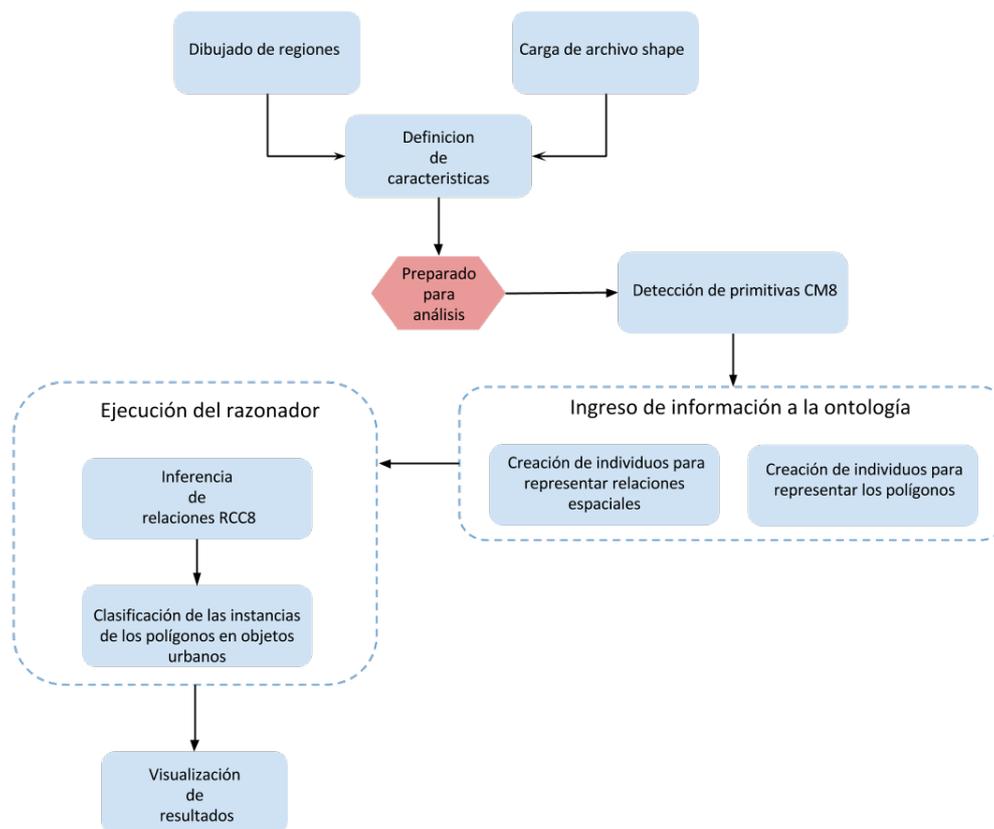


Figura 4.5: Diagrama de flujo de una ejecución típica

Para comenzar con la detección de objetos urbanos en una imagen, tenemos dos alternativas: cargar una imagen ráster y dibujar los polígonos (o regiones), o importar una segmentación mediante un archivo *shapefile*.

Una vez la imagen cargada y los polígonos dibujados se deberá ingresar toda la información que el usuario pueda aportar sobre los mismos mediante el formulario de características. Cuando finaliza esta etapa la aplicación se encuentra lista para la ejecución de los diferentes métodos de análisis. Al presionar el botón “Run” se comienza, en primer término, con la computación de las primitivas CM8. Seguidamente se produce la creación de individuos en la ontología, tanto para representar las relaciones espaciales RCC8, como para las regiones o polígonos que más tarde serán identificados como objetos urbanos. En la etapa de “Ejecución del razonador”, la aplicación invoca al razonador para inferir las relaciones RCC8 y las clases de pertenencia de las regiones instanciadas. Finalmente, los resultados están disponibles para su visualización.

### 4.3. Ejemplo de uso

Para demostrar el uso la aplicación, consideraremos el ejemplo que se describe a continuación.

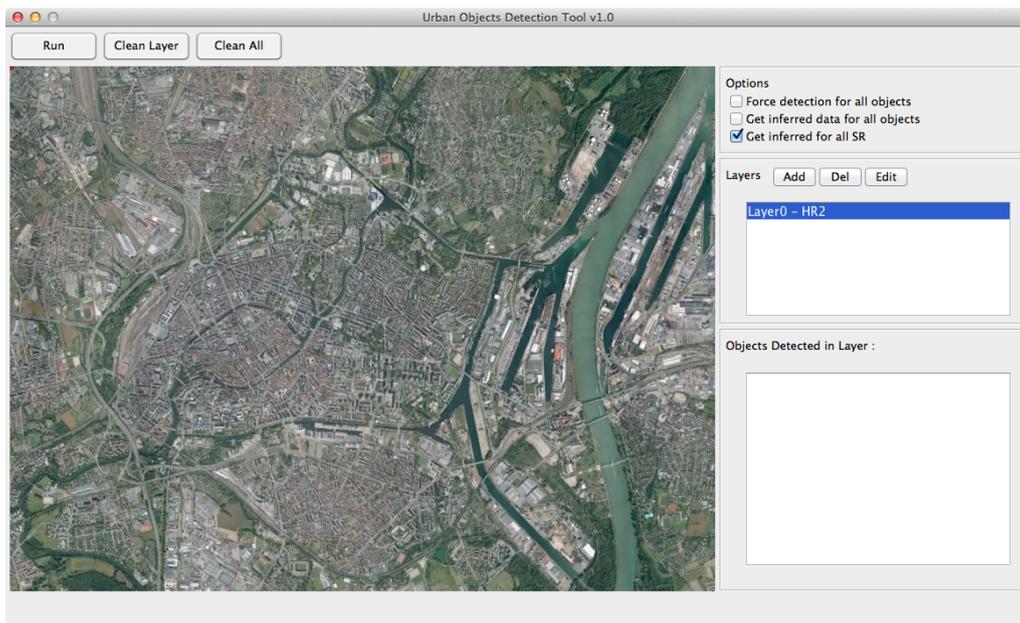


Figura 4.6: Aplicación cargada con imagen de resolución HR2

La figura 4.6 muestra una instancia de ejecución de la aplicación, la cual contiene en su panel principal una imagen satelital con resolución HR2 de la parte central de la ciudad de Strasbourg.

Supongamos que se dibujan los siguientes polígonos, representados en la figura 4.7 y que además, como un primer intento de clasificarlos manualmente, se les asignan los siguientes tipos de objetos urbanos:

- Polígono P0 : Surface Vegetal
- Polígono P1 : TU
- Polígono P2 : TU
- Polígono P3 : TU
- Polígono P4 : TU

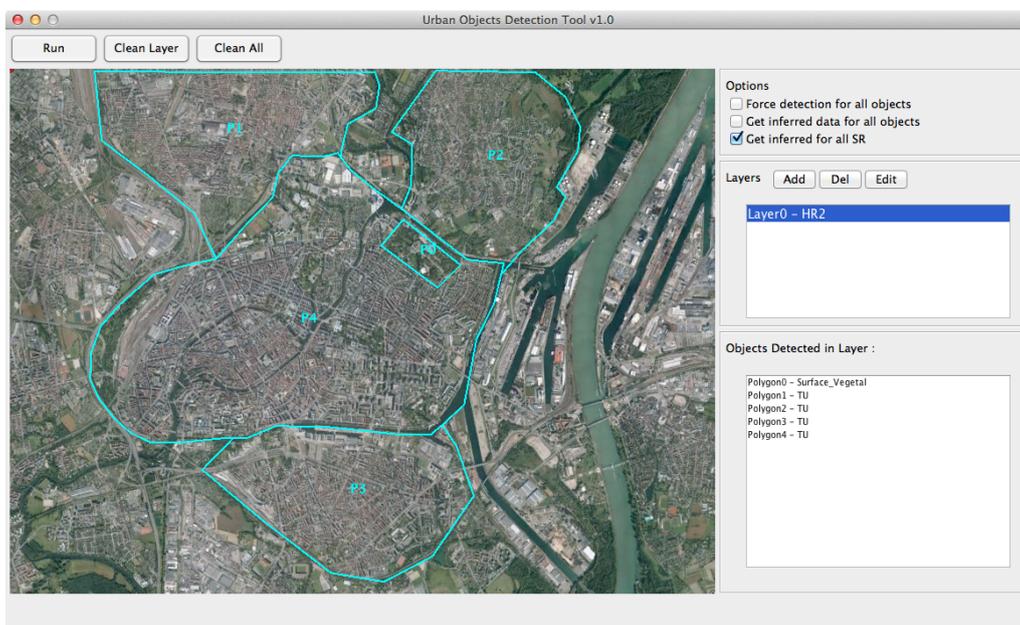


Figura 4.7: Segmentación de la imagen de resolución HR2

Supongamos además que nos interesa conocer con más detalle que tipo de *TU* representa P4. Para esto completaremos algunas de las características del objeto que representa este polígono mediante su formulario correspondiente.

Observando con más detenimiento la imagen, podemos apreciar que el polígono P4 posee una textura heterogénea y una superficie algo discontinua, por lo tanto reflejaremos dichas características en el formulario como se muestra en la figura 4.8.

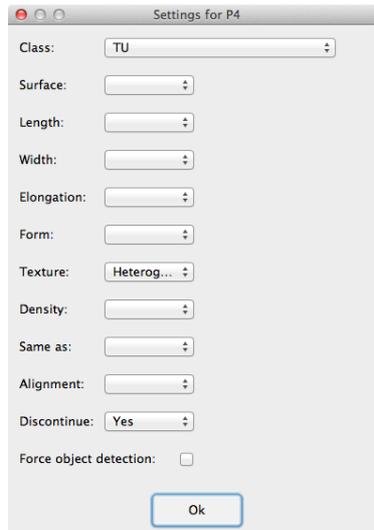


Figura 4.8: Formulario de características del polígono P4

Una vez finalizado este paso, presionando el botón “Run” daremos comienzo al procesamiento de la información ingresada. Inmediatamente, un cuadro de diálogo aparecerá mostrando el progreso de la clasificación de objetos y la inferencia de relaciones. Luego de alcanzar el 100 % del progreso, podemos observar en el panel de objetos de la capa activa que P4 aparece clasificado ahora como *TU\_Discontinua* (figura 4.9).

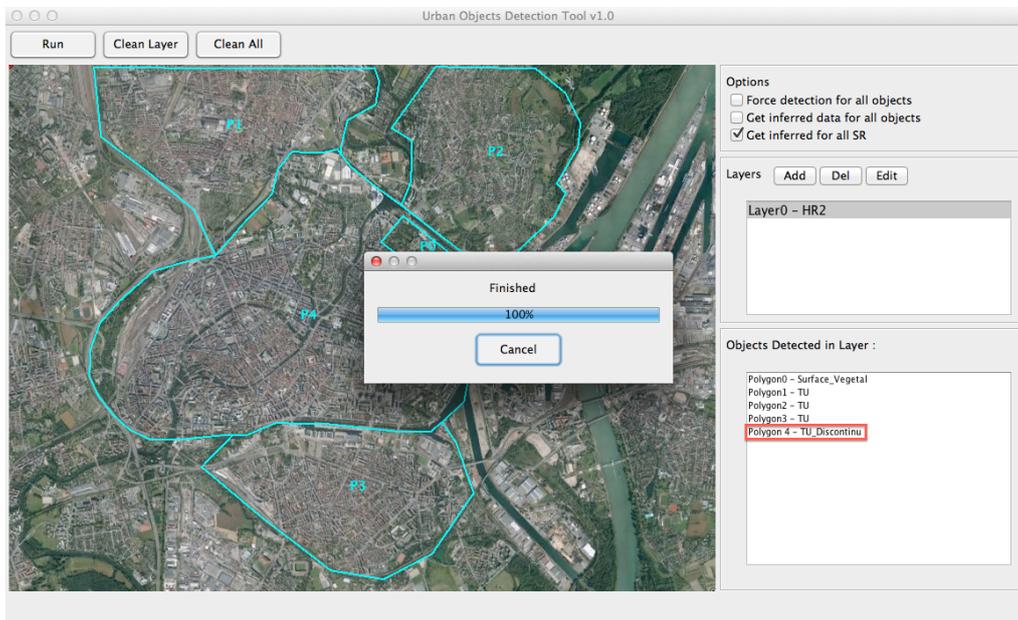


Figura 4.9: Tipo inferido para el polígono P4

Para saber si la clasificación fue correcta deberemos revisar las relaciones espaciales inferidas que involucran a este objeto. Mediante la opción “Show Results” del menú “Debug” podremos acceder a esta información. Para mayor claridad, a continuación se muestra un extracto muy resumido de la información presentada en la ventana de resultados, correspondiente al polígono P4, con las secciones más relevantes para este ejemplo. Primitivas CM8 calculadas entre el polígono 4 y el resto de los polígonos:

$x = \text{Polygon } 4, y = \text{Polygon } 0 :: \text{NP}(x,y) \mid \text{P-1}(x,y) \mid \text{O}(x,y) \mid \text{NA}(x,y)$

$x^\circ - y^\circ \neq \emptyset \implies x \text{ is not contained in } y$

$y^\circ - x^\circ = \emptyset \implies x \text{ contains } y$

$x^\circ \text{ INTERSECT } y^\circ \neq \emptyset \implies x \text{ overlaps } y$

$\partial x \text{ INTERSECT } \partial y = \emptyset \implies x \text{ does not share boundary with } y$

$x = \text{Polygon } 4, y = \text{Polygon } 1 :: \text{NP}(x,y) \mid \text{NP-1}(x,y) \mid \text{DR}(x,y) \mid \text{A}(x,y)$

$x^\circ - y^\circ \neq \emptyset \implies x \text{ is not contained in } y$

$y^\circ - x^\circ \neq \emptyset \implies x \text{ does not contain } y$

$x^\circ \text{ INTERSECT } y^\circ = \emptyset \implies x \text{ interior disjoint from } y$

$\partial x \text{ INTERSECT } \partial y \neq \emptyset \implies x \text{ share boundary with } y$

$x = \text{Polygon } 4, y = \text{Polygon } 2 :: \text{NP}(x,y) \mid \text{NP-1}(x,y) \mid \text{DR}(x,y) \mid \text{A}(x,y)$

$x^\circ - y^\circ \neq \emptyset \implies x \text{ is not contained in } y$

$y^\circ - x^\circ \neq \emptyset \implies x \text{ does not contain } y$

$x^\circ \text{ INTERSECT } y^\circ = \emptyset \implies x \text{ interior disjoint from } y$

$\partial x \text{ INTERSECT } \partial y \neq \emptyset \implies x \text{ share boundary with } y$

$x = \text{Polygon } 4, y = \text{Polygon } 3 :: \text{NP}(x,y) \mid \text{NP-1}(x,y) \mid \text{DR}(x,y) \mid \text{A}(x,y)$

$x^\circ - y^\circ \neq \emptyset \implies x \text{ is not contained in } y$

$y^\circ - x^\circ \neq \emptyset \implies x \text{ does not contain } y$

$x^\circ \text{ INTERSECT } y^\circ = \emptyset \implies x \text{ interior disjoint from } y$

$\partial x \text{ INTERSECT } \partial y \neq \emptyset \implies x \text{ share boundary with } y$

Relaciones RCC8 inferidas entre el polígono 4 y el resto de los polígonos :

SR\_P4\_P0 : NTPPi

SR\_P4\_P1: EC

SR\_P4\_P2 : EC

SR\_P4\_P3 : EC

Información inferida para el polígono P4:

Classes:

Objet\_Geo

```
TU_Discontinuu
Objet_Geo_Construit
Surface_Artificiel
.....
Object Properties: isComposedOf P0

isAdjacentTo P1
isAdjacentTo P2
isAdjacentTo P3

Datatype Properties:

None
```

Como podemos ver, las primitivas CM8 computadas por la aplicación, son correctas y la deducción de las relaciones RCC8 a partir de estas primitivas también lo son, de acuerdo con la tabla 3.1. De la información inferida en el polígono P4, podemos observar que las reglas SWRL establecieron correctamente la equivalencia entre las relaciones espaciales y nuestras *object properties*. Con esta notación es más fácil apreciar que la aplicación dedujo correctamente las relaciones entre los polígonos y P4.

De acuerdo a la definición de la clase *TU\_Discontinuu*, este objeto es apreciable a resolución HR2, tiene una textura heterogénea, es discontinuo, incluye entre otros objetos a una superficie vegetal y es adyacente a otros objetos de tipo *TU*. Esto concuerda con la información ingresada sobre P4, con las características del resto de los objetos y con las relaciones espaciales inferidas. Por lo tanto podemos afirmar que la inferencia de la membresía del polígono P4 fue correcta.

Para extender nuestro ejemplo, supongamos ahora que disponemos de una imagen con resolución THR1 de una sección más ampliada del polígono P4, más precisamente, una vista en detalle del polígono P0. Para utilizar esta imagen en conjunto con la anterior, la misma debe ser cargada en una nueva capa de resolución THR1 (figura 4.10).

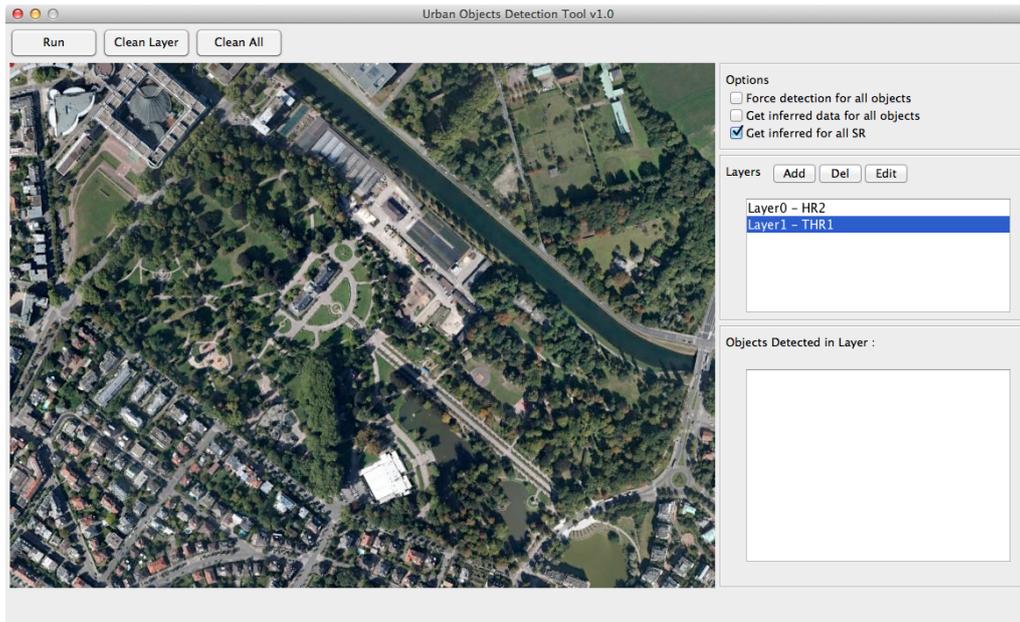


Figura 4.10: Imagen de resolución THR1 cargada en la nueva capa

Consideremos los siguientes polígonos dibujados, representados en la figura 4.11, de los cuales fueron detectados por el usuario los siguientes tipos de objetos urbanos:

- Polígono P5 : Route Autre
- Polígono P6 : Route Autre
- Polígono P7 : Groupe Arbres
- Polígono P8 : Groupe Arbres
- Polígono P9 : Alignemet Arbres
- Polígono P10 : Surface Vegetal

Supongamos además, que nos interesa conocer de manera mas detallada que tipo de superficie verde representa el polígono P10. En este caso, dada las características de la imagen, sólo podemos afirmar con seguridad que P10 contiene a sus objetos de manera no alineada. Sin embargo, analizando las dos capas en conjunto, podemos notar que P10 y P0 refieren al mismo objeto. Con lo cual, podemos expresar dicha igualdad desde el formulario de características de P10, seleccionando a P0 del listado de probables objetos similares (figura 4.12).

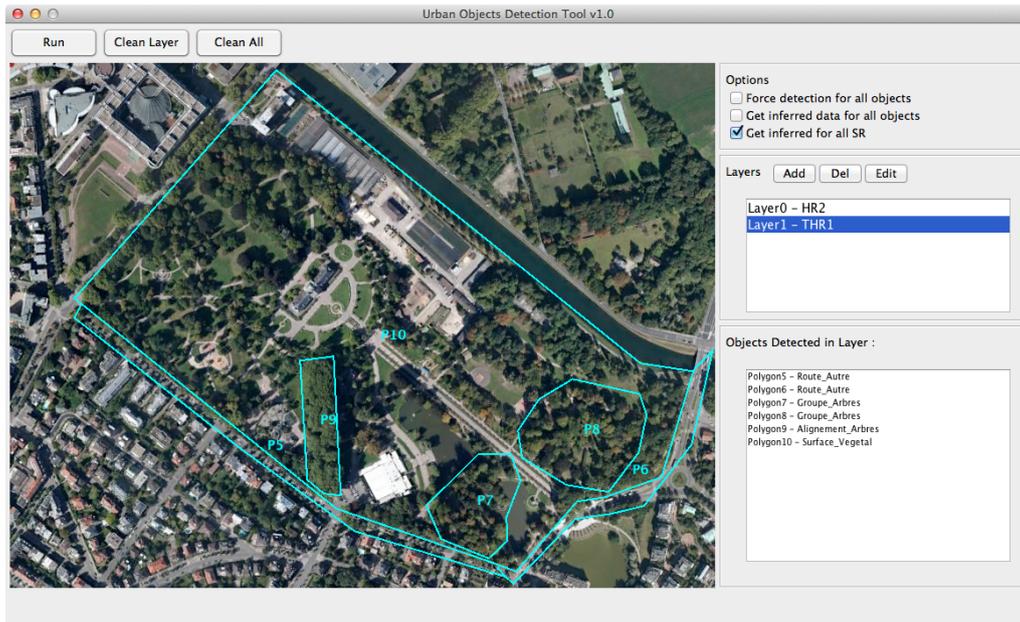


Figura 4.11: Segmentación de la imagen de resolución THR1

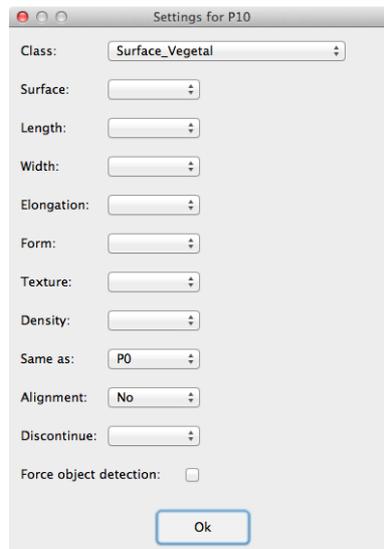


Figura 4.12: Formulario de características del polígono P10

Si comenzamos con el procesamiento de la información, luego de unos minutos tenemos que P10 es clasificado como *Parc* (figura 4.13), lo cual pareciera ser correcto de acuerdo a lo que puede apreciarse en la imagen.

Para saber si esta clasificación es correcta, al igual que hicimos en el ejemplo anterior,

deberemos analizar las relaciones espaciales inferidas para esta capa. Nuevamente, sólo presentaremos de manera resumida la información relevante al polígono P10.

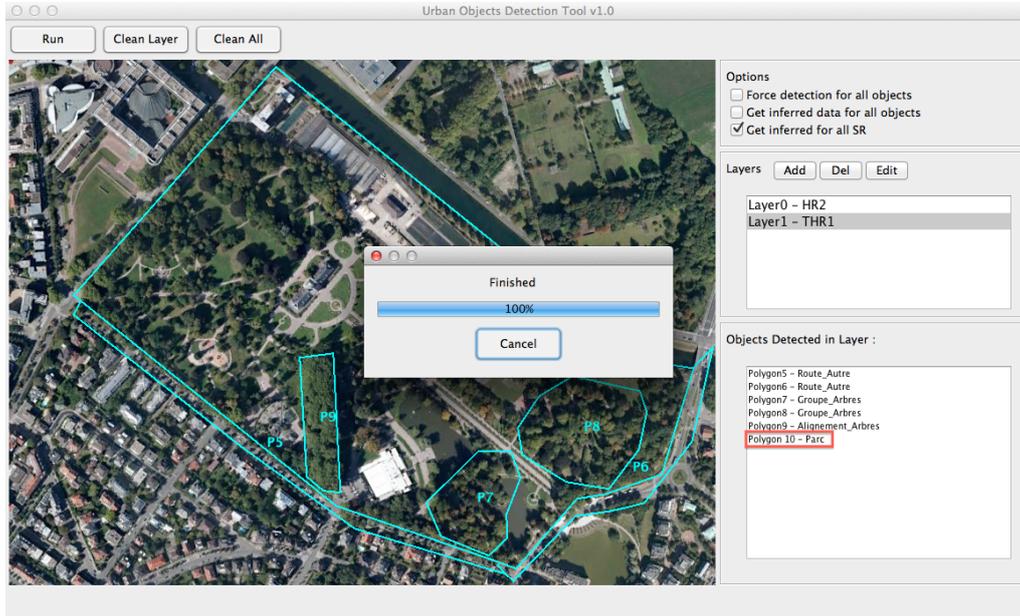


Figura 4.13: Tipo inferido para el polígono P10

Primitivas CM8 calculadas entre el polígono 10 y el resto de los polígonos:

$x = \text{Polygon 10}, y = \text{Polygon 5} :: \text{NP}(x,y) - \text{NP-1}(x,y) - \text{DR}(x,y) - \text{A}(x,y)$

$x^\circ - y^\circ \neq \emptyset \implies x \text{ is not contained in } y$   
 $y^\circ - x^\circ \neq \emptyset \implies x \text{ does not contain } y$   
 $x^\circ \text{ INTERSECT } y^\circ = \emptyset \implies x \text{ interior disjoint from } y$   
 $\partial x \text{ INTERSECT } \partial y \neq \emptyset \implies x \text{ share boundary with } y$

$x = \text{Polygon 10}, y = \text{Polygon 6} :: \text{NP}(x,y) - \text{NP-1}(x,y) - \text{DR}(x,y) - \text{A}(x,y)$

$x^\circ - y^\circ \neq \emptyset \implies x \text{ is not contained in } y$   
 $y^\circ - x^\circ \neq \emptyset \implies x \text{ does not contain } y$   
 $x^\circ \text{ INTERSECT } y^\circ = \emptyset \implies x \text{ interior disjoint from } y$   
 $\partial x \text{ INTERSECT } \partial y \neq \emptyset \implies x \text{ share boundary with } y$

$x = \text{Polygon 10}, y = \text{Polygon 7} :: \text{NP}(x,y) - \text{P-1}(x,y) - \text{O}(x,y) - \text{NA}(x,y)$

$x^\circ - y^\circ \neq \emptyset \implies x \text{ is not contained in } y$   
 $y^\circ - x^\circ = \emptyset \implies x \text{ contains } y$

$x^\circ \text{ INTERSECT } y^\circ \neq \emptyset \implies x \text{ overlaps } y$   
 $\partial x \text{ INTERSECT } \partial y = \emptyset \implies x \text{ does not share boundary with } y$

$x = \text{Polygon } 10, y = \text{Polygon } 8 :: \text{NP}(x,y) - \text{P-1}(x,y) - \text{O}(x,y) - \text{NA}(x,y)$

$x^\circ - y^\circ \neq \emptyset \implies x \text{ is not contained in } y$   
 $y^\circ - x^\circ = \emptyset \implies x \text{ contains } y$   
 $x^\circ \text{ INTERSECT } y^\circ \neq \emptyset \implies x \text{ overlaps } y$   
 $\partial x \text{ INTERSECT } \partial y = \emptyset \implies x \text{ does not share boundary with } y$

$x = \text{Polygon } 10, y = \text{Polygon } 9 :: \text{NP}(x,y) - \text{P-1}(x,y) - \text{O}(x,y) - \text{NA}(x,y)$

$x^\circ - y^\circ \neq \emptyset \implies x \text{ is not contained in } y$   
 $y^\circ - x^\circ = \emptyset \implies x \text{ contains } y$   
 $x^\circ \text{ INTERSECT } y^\circ \neq \emptyset \implies x \text{ overlaps } y$   
 $\partial x \text{ INTERSECT } \partial y = \emptyset \implies x \text{ does not share boundary with } y$

Relaciones RCC8 inferidas entre el polígono 10 y el resto de los polígonos :

- SR\_P10\_P5 : EC
- SR\_P10\_P6 : EC
- SR\_P10\_P7 : NTPPi
- SR\_P10\_P8 : NTPPi
- SR\_P10\_P9 : NTPPi

Información inferida para el polígono P10:

Classes:

- Parc
- Objet\_Geo
- Objet\_Geo\_Construit
- .....

Object Properties:

- isComposedOf P8
- isComposedOf P9
- isComposedOf P7
- isAdjacentTo P5
- isAdjacentTo P6
- isIncludedIn P4

Datatype Properties:

None

De acuerdo a lo observado en la imagen, podemos confirmar que tanto las primitivas CM8 computadas, como la posterior inferencia de las relaciones RCC8 a partir de estas, son correctas. En cuanto a la información inferida para el polígono P10, podemos asegurar que las *object properties* se corresponden correctamente con las relaciones espaciales, pero el tipo de objeto urbano todavía debe ser constatado con la definición de la ontología.

Según la definición de la clase *Parc*, este objeto es identificable bajo resolución THR1, es adyacente a objetos de tipo *Autre.Route* y se encuentra incluido en objetos de tipo *TU\_Discontinuu*. Entre los diferentes tipos de objetos que componen a *Parc* se encuentran los grupos y alineamientos de árboles. Sin embargo, como detalle importante, debemos mencionar que estos objetos no se encuentran alineados entre sí.

A excepción de la inclusión, que no se encuentra especificada de manera explícita en P10, la definición de la clase *Parc* coincide parcialmente con la información inferida y afirmada del polígono, sin embargo, no es suficiente para establecer la membresía a esta clase. Entonces, por qué el razonador infirió que P10 es un objeto de tipo *Parc*? La respuesta está en la igualdad entre individuos establecida entre P10 y P0. Dado que en el ejemplo anterior P0 se había inferido como incluido en un *TU\_Discontinuu*, la igualdad establecida entre P0 y P10 le permite al razonador deducir que P10 se encuentra incluido en un *TU\_Discontinuu*, y por lo tanto, junto a la información anterior, inferir que es de tipo *Parc*.

La igualdad anterior funciona en ambos sentidos, lo que significa que si P10 es de tipo *Parc*, P0 también lo es. Esto parecería ser una contradicción a la definición del objeto *Parc*, ya que P0 se encuentra en una capa de resolución HR2 y *Parc* no es identificable en esta resolución. Sin embargo, que un objeto no sea identificable en una imagen con una determinada resolución, no significa que no exista en esta, sino que la resolución de la misma no permite caracterizar sus atributos y por lo tanto hace más difícil su detección. En nuestro primer ejemplo, la resolución no era lo suficientemente alta como para reconocer con mayor detalle al objeto denotado por P0, por lo que fue declarado como *Surface\_Artificiel*. Posteriormente, la inclusión de la capa THR1 permitió deducir que en realidad P0 se trataba de un parque, el cual al ser un tipo de *Surface\_Artificiel*, no contradice con la anterior definición y no genera ninguna modificación en la inferencia de clases de los objetos de la capa HR2.

Con este conjunto de ejemplos se demostró como, mediante la multi-resolución, la cual es “simulada” mediante la vinculación adecuada de objetos de capas diferentes, es posible inferir resultados que normalmente no son posibles de obtener con una única imagen. Esto deja en evidencia la importancia de la aplicación, no sólo para crear instancias de manera más sencilla o para obtener automáticamente relaciones espaciales, sino para obtener resultados que antes no eran posibles, o muy difíciles de obtener debido a los pasos involucrados.

## Capítulo 5

# Conclusiones y trabajos futuros

### 5.1. Conclusiones

La formalización del conocimiento del dominio, mediante la construcción de un modelo basado en lógica, asegura la consistencia del mismo. Teniendo esto en cuenta, una ontología fue desarrollada a partir de un diccionario de objetos urbanos como una alternativa interesante para formalizar el conocimiento.

La elección de OWL como lenguaje de ontologías se basó en su capacidad de razonamiento. Esto permitió que fuera posible obtener información que usualmente se encuentra implícita en el dominio cuando fue necesario.

Mediante la integración del modelo propuesto por [22] a la ontología, se logró incorporar razonamiento espacial cualitativo, lo que nos permitió, entre otras cosas, inferir nuevas relaciones entre objetos a partir de relaciones conocidas. Este modelo introdujo el uso de primitivas CM8 para expresar relaciones RCC8. Dichas primitivas se componen de operaciones geométricas elementales, las cuales pueden ser calculadas por medio de librerías para el análisis topológico.

Mediante la aplicación desarrollada, no sólo fue posible computar automáticamente las primitivas a partir de polígonos dibujados en una imagen, sino que además, se logró implementar una interfaz amigable y funcional para el usuario común, que permite interactuar con la ontología de manera transparente. El uso de valores cualitativos en la ontología y posteriormente en la aplicación, facilitó aún más el trabajo de los expertos para reconocer objetos cuando los valores numéricos no están presentes. Por otra parte, los resultados provistos por la aplicación permiten realizar un completo análisis espacial de los objetos presentes en una imagen.

Los ejemplos de la sección 4.3 demuestran la efectividad del modelo para identificar objetos basados en sus características espaciales y atributos. Dado que este modelo está basado en la información extraída de un diccionario de objetos urbanos escrito por expertos, existen algunos resultados inesperados, como los que aparecen cuando ocurre una “multi-clasificación”. Sin embargo, cada inferencia, incluidas las de este tipo, está justificada por la información que puede encontrarse allí, ya sea implícita o no.

Este trabajo establece las bases para la formalización del conocimiento urbanístico de

una ciudad mediante ontologías. El enfoque utilizado para el desarrollo del modelo permite fácilmente adaptar el mismo a diferentes dominios, permitiendo expandir el ámbito de aplicación no sólo a la ciudad de Strasbourg sino a cualquier ciudad de la cual se tenga una completa descripción de sus objetos urbanos.

## 5.2. Trabajos futuros

Cualquier esquema de representación, destinado a modelar las acciones y procesos del mundo real, debe ser capaz de hacer frente a los efectos de fenómenos inciertos. Casos como los expuestos durante el desarrollo de la ontología, donde un objeto puede ser miembro de múltiples clases al mismo tiempo, representa cierta incertidumbre por parte del modelo. Una forma de sacar ventaja de esta situación, es utilizar un enfoque basado en ontologías probabilísticas. La idea detrás de esta propuesta es, dada una instancia de un objeto urbano desconocido, obtener con que probabilidad este objeto se clasifica efectivamente como un objeto urbano determinado, dada cierta información estadística. Sin embargo, la habilidad de OWL para representar y razonar con incertidumbres representa un problema. Afortunadamente, existen extensiones para el lenguaje, como por ejemplo **PR-OWL** [30] y **PRONTO** [31], que posibilitan el razonamiento probabilístico en OWL y por lo tanto merecen ser estudiadas.

Expresar conceptos como muy cercano, cercano, medianamente cercano, alejado, muy alejado, etc, que resultan normales para un humano, en este trabajo serían de gran utilidad para tener otra forma de caracterizar espacialmente a los objetos urbanos. Pero, para esto, es necesario contar con algún tipo de razonamiento que utilice *Fuzzy Logic* (lógica difusa), el cual OWL no posee. Sin embargo, en [32], se expone una extensión del lenguaje OWL, denominada **Fuzzy OWL**, empleada para definir ontologías con elementos de la lógica difusa, que podría considerarse en este caso.

La aplicación todavía tiene mucho trabajo por delante. Debido a la falta de tiempo, quedaron algunas funcionalidades por mejorar y otras por implementar. Una de estas está referida a mejorar el dibujado de polígonos. Se necesitan más y mejores herramientas para dibujar las regiones, como así también para manipular los polígonos una vez dibujados.

Otra falencia es la falta de un mecanismo que permita guardar una segmentación hecha en la aplicación en formato *shape*. Esto complicaría un poco el proceso actual del dibujado de polígonos debido a que se necesitaría tener conocimiento de la geolocalización de la imagen para construir el archivo *shape*.

El mayor punto débil, y por ende, de mayor limitación, quizás sea el manejo de memoria y la *performance* de la aplicación. Uno de los principales responsables es la utilización del razonador lógico. Cuando la cantidad de individuos es elevada, el espacio en memoria se reduce, la *performance* del razonador se ve afectada y por consecuencia, también la *performance* de la aplicación. Para disminuir estos efectos, una opción a considerar podría ser la reducción en la frecuencia con la que se invoca al razonador para inferir resultados. De no ser posible, una alternativa sería intentar con otra API para OWL, como por ejemplo **Apache Jena** [33].

# Bibliografía

- [1] Forestier G., Puissant A., Wemmert C., Gañçarski P. “*Knowledge-based Region Labeling for Remote Sensing Image Interpretation*”. Universidad de Strasbourg, Francia. LSIIT - Laboratoire des Sciences de l’Image, de l’Informatique et de la Télédétection. LIVE - Laboratoire Image, Ville, Environnement. 2011.
- [2] Lillesand T.M., Kiefer W., Chipman J.W. “*Remote Sensing and Image Interpretation*”. Wiley, 2003.
- [3] Herold M., Scepan J., Muller A. Gunter S. “*Object-oriented mapping and analysis of urban land use/cover using ikonos data*”. In Proc. 22nd. Earsel Symposium Geoinformation for European-Wide Integration, pages 531-538. Prague, June 2002.
- [4] Benz U, Hofmann P., Willhauck G., Lingenfelder I., Heynen M. “*Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information*”. ISPRS Journal of Photogrammetry and Remote Sensing, 58:239-258, 2004.
- [5] Cleve C., Kelly F., abd Kearns M., Moritz M. “*Classification of the wild-land urban interface: A comparison of pixel and object-based classifications using high resolution aerial photography*”. Computers, Environment and Urban Systems, doi:10.1016/j.compenvurbsys.2007.10.001, 2008.
- [6] Gruber, T. R. “*Toward Principles for the Design of Ontologies Used for Knowledge Sharing*”. International Journal Human-Computer Studies, 43(5-6):907-928, 1995.
- [7] Gruber, T.R. “Ontology” *Encyclopedia of Database Systems*. Ling Liu and M. Tamer Özsu (Eds), Springer-Verlag, 2008.  
<http://http://tomgruber.org/writing/ontology-definition-2007.htm>
- [8] Baltasavias E. “*Object extraction and revision by image analysis using existing geo-data and knowledge: current status and steps towards operational systems*”. ISPRS Journal of Photogrammetry and Remote Sensing. 58:129-151, 2004.
- [9] World Wide Web Consortium.  
“*The Resource Description Framework*”. <http://www.w3.org/RDF/>  
“*RDF Primer*”. <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/>

- [10] World Wide Web Consortium. “*The Resource Description Framework Schema*”.  
<http://www.w3.org/TR/rdf-schema/>
- [11] World Wide Web Consortium. “*OWL Web Ontology Language Overview*”.  
<http://www.w3.org/TR/owl-features/>
- [12] World Wide Web Consortium. “*OWL 2 Web Ontology Language Primer (Second Edition)*”.  
<http://www.w3.org/TR/2012/REC-owl2-primer-20121211/>
- [13] World Wide Web Consortium. “*OWL 2 Web Ontology Language Document Overview (Second Edition)*”.  
<http://www.w3.org/TR/owl2-overview/>
- [14] Clementini E., Di Felice P., Van Oosterom, P. “*A Petit set of formal topological relationships suitable for end-user interaction*”. Proceedings of the Third International Symposium on Advances in Spatial Databases. 277-295, 1993.
- [15] Hogenboom F., Frasincar F., Kaymak U. “*A Review of Approaches for Representing RCC8 in OWL*”. Econometric Institute Erasmus University Rotterdam. Rotterdam, the Netherlands. 2010.
- [16] Protégé. “*A free, open source ontology editor and knowledge-base framework*”.  
<http://protege.stanford.edu/>
- [17] Fact++. <http://owl.man.ac.uk/factplusplus/>
- [18] Pellet. <http://clarkparsia.com/pellet/>
- [19] Hermit. <http://hermit-reasoner.com/>
- [20] PR-OWL. “*An extension to the OWL language to represent probabilistic ontologies*”.  
<http://www.pr-owl.org/>
- [21] PRONTO. “*An extension of Pellet that enables probabilistic knowledge representation and reasoning in OWL ontologies*”. <http://pellet.owldl.com/pronto>
- [22] Marc-Zwecker S., de Bertrand de Beuvron F., Zanni-Merk C., Le Ber F. “*Qualitative Spatial Reasoning in RCC8 with OWL and SWRL*”. In KEOD 2013-International Conference on Knowledge Engineering and Ontology Development. 2013.
- [23] Randell D.A., Cohn A.G., Cui Z. “*A spatial logic based on regions and connection*”. Proceedings of the 3rd International Conf. Knowledge Representation and Reasoning, Morgan Kaufmann, Los Allos, CA, pp.165-176. 1992.
- [24] D.A. Randell, Z. Cui, A.G. Cohn. “*Computing transitivity tables: A challenge for Automated Theorem Provers*”. 11th International Conference on Automated Deduction Saratoga Springs, NY, USA, June 15–18, 1992. Automated Deduction—CADE-11. Lecture Notes in Computer Science. Volume 607, pp 786-790.

- [25] World Wide Web Consortium. “*SWRL: A Semantic Web Rule Language Combining OWL and RuleML*”.  
<http://www.w3.org/Submission/SWRL/>
- [26] Protégé’s SWRL Language FAQ.  
<http://protege.cim3.net/cgi-bin/wiki.pl?SWRLLanguageFAQ#nid8JQ>.
- [27] Le Ber F. and Napoli A. “*Design and comparison of lattices of topological relations for spatial representation and reasoning*”. *Journal of Experimental and Theoretical Artificial Intelligence*, 15(3):331–371. 2003.
- [28] The OWL API.  
<http://owlapi.sourceforge.net/>
- [29] Java Topology Suite.  
<http://www.vividsolutions.com/jts/JTSHome.htm>
- [30] PR-OWL: A Bayesian extension to the OWL Ontology Language.  
<http://www.pr-owl.org/>
- [31] Introducing Pronto: Probabilistic DL Reasoning in Pellet.  
<http://weblog.clarkparsia.com/2007/09/27/introducing-pronto>
- [32] G. Stoilos, G. Stamou, V. Tzouvaras, J.Z. Pan and I. Horrocks. “*Fuzzy OWL: Uncertainty and the Semantic Web*”. In *Proc. of the International workshop on OWL: Experience and Directions (OWL-ED2005)*. 2005.
- [33] A free and open source Java framework for building Semantic Web and Linked Data applications.  
<http://jena.apache.org/>