Representación formal del conocimiento enciclopédico en FunGramKB:

el concepto PLACE

Universidad Nacional De Educación A Distancia

MÁSTER EN LINGÜÍSTICA INGLESA APLICADA

Trabajo de Investigación de Fin de Máster

Junio 2010

Tutor: Dr. Ricardo Mairal Usón

Alumna: María de los Llanos Carrión Varela

DNI 47.063.169-V

ÍNDICE

INTRODUCCIÓN

El presente trabajo de investigación tiene como objetivo crear y describir una metodología que permita la representación formal del conocimiento enciclopédico en el módulo Onomasticón de la base de conocimiento FunGramKB. Esta representación formal se realiza de manera optimizada mediante la población semi-automática del Onomasticón. El protocolo que posibilita dicha población semi-automática posee como punto de partida inicial la creación de reglas en lenguaje COREL basadas en plantillas, las cuales posteriormente permitirán la proyección de información tomada de herramientas accesibles online (la base de datos DBpedia) en el Onomasticón de FunGramKB. Con objeto de llevar a cabo el protocolo de creación de reglas, se tomará el concepto PLACE de la base FunGramKB y se crearán reglas en lenguaje COREL que permitan representar formalmente el conocimiento enciclopédico de las entidades que lo componen, tomando como referencia la clase "*Place*" de DBpedia a través de la herramienta creada ad hoc en el Onomasticón de FunGramKB (DBpedia Mapper). Finalmente, se realizará un análisis de los productos obtenidos (reglas) tras la aplicación del protocolo, así como se describirán las conclusiones y futuras implicaciones que se desprenden de dicho análisis.

En primer lugar, en el Capítulo 1 se mostrará una descripción y análisis del estado de la cuestión y las diferentes perspectivas dentro del ámbito de la creación de ontologías y bases de conocimiento, lo cual definirá el marco de trabajo en el cual se encuadra este estudio y la necesidad de ser desarrollado.

Posteriormente, en el Capítulo 2 se presentará de manera detallada el módulo, dentro de la base de conocimiento FunGramKB, en el que se centrará la descripción y posterior análisis de los protocolos llevados a cabo. Como se ha referido anteriormente, este módulo es el Onomasticón. También se detallará la naturaleza del mismo y su afinidad o discrepancia con similares estructuras creadas dentro de otros proyectos actuales (DBpedia, YAGO), así como su viabilidad para ser utilizadas como herramientas para la población semi-automática del Onomasticón. Se verá, de este modo, la manera en que el Onomasticón de FunGramKB es capaz de maximizar la capacidad de reusabilidad que dichas estructuras puedan brindar, en especial de la base de datos DBpedia.

Con posterioridad a esta descripción de las herramientas ontológicas que serán reutilizadas, en el Capítulo 3 se presentará de manera detallada la metodología y protocolo

seguidos de manera sistemática para la creación de estructuras (reglas) que permitan poblar el Onomasticón de FunGramKB. Esta fase correspondería a la parte práctica del estudio, ya que se trata de crear dichas reglas de manera manual para posteriormente poblar el Onomasticón de manera semi-automática mediante la implementación computacional (proyección) de las reglas en el DBpedia Mapper de dicho módulo. Asimismo, se incluirá también un análisis detallado de este proceso de creación de reglas, examinando de manera rigurosa la metodología de este procedimiento. A continuación, se presentarán los resultados obtenidos (reglas) tras la aplicación del protocolo y metodología descritos (los cuales se incluyen en el Apéndice I del presente trabajo) y se efectuará una valoración de los mismos, así como de la información que los resultados arrojan.

Finalmente, se presentarán las conclusiones obtenidas tras la creación y análisis de las reglas, además de las implicaciones de todo el proceso y las futuras líneas de investigación que se abren a través del protocolo desarrollado y sus productos, y que se estima que habrían de seguirse dentro de este campo tras la finalización del presente trabajo.

CAPÍTULO 1. ONTOLOGÍAS Y BASES DE CONOCIMIENTO: ESTADO DE LA CUESTIÓN. LA BASE DE CONOCIMIENTO FUNGRAMKB

1.1. PROBLEMÁTICA ORIGINAL

En el siglo en que vivimos y, más aún, en la sociedad actual, la información se perfila como el gran bien de nuestro tiempo. El mundo alfabetizado presume ser el más preparado de todos los tiempos, debido al masivo flujo de conocimiento y datos que, gracias a herramientas como Internet, circulan globalmente. Sin embargo, a pesar de esta ingente efusión de datos, las máquinas que los procesan no son capaces de comprenderlos. Se limitan a transfundirlos entre usuarios, a proporcionarlos a golpe de click, sin tener la capacidad de filtrar o depurar aquellos que en realidad son relevantes para el usuario que formula la consulta, con la consiguiente pérdida de tiempo. Los actuales motores de búsqueda en la red adolecen del llamado conocimiento semántico: son capaces de identificar y reconocer una cadena de símbolos gráficos, letras, que forman palabras, mas sin tener capacidad de acertar qué se esconde tras esos símbolos o qué persigue el usuario mediante su búsqueda. Obviamente, el dotar de contenido semántico a esa información, que para la máquina son meros símbolos despojados de significado, supone adentrarse en los dominios de la Inteligencia Artificial. No obstante, los usos inmediatos de los que la sociedad en general lograría beneficiarse podrían resumirse en palabras de Giles (2008): "*Users of the semantic web will be able to ask questions rather than search for phrases*".

No son pocos los intentos que se han realizado en los últimos tiempos para dotar de contenido semántico a textos escritos en lenguaje natural. Muchos de ellos se han vinculado a la creación de grandes repositorios de conocimiento del mundo en un lenguaje formal que la máquina pueda entender y procesar, lo que llamamos *ontologías* y *bases de conocimiento*, según las características que posean, y cuyas diferencias mutuas más adelante se discutirán con mayor detalle en este trabajo. Un claro ejemplo de estas tareas de recopilación de información universal se puede encuadrar dentro del titánico proyecto Cyc (Lenat et al. 1990), que dio comienzo a finales de 1984 y que todavía sigue en ejecución. Actualmente, según datos de Cycorp[1], la base de datos de Cyc contiene casi quinientos mil términos, incluyendo unos mil quinientos tipos de relaciones y cinco

[1] http://www.cyc.com/

millones de hechos (aserciones) que relacionan estos términos. El calificativo de titánico tiene que ver con el hecho de que la mayoría de estos datos han sido codificados e insertados manualmente en la base de datos de Cyc. Merece la pena mencionar el punto de partida de este proyecto: ya en 1984, los investigadores de la empresa norteamericana Microelectronics and Computer Technology Corporation (MCC) en Austin, Texas, habiéndose percatado de que las máquinas eran capaces de resolver complejas operaciones de diversos tipos, advirtieron que éstas carecían de respuestas para otras cuestiones mucho más básicas. A este fenómeno lo denominaron "[…] brittle *performance: when the system is confronted by some unanticipated situation, the program is likely to reach the wrong conclusion*" (Lenat et al. 1990: 32). Esto se puede explicar como una incapacidad de inferir datos, por simples que sean, ya que si no se han anticipado antes, la computadora ignora su existencia. O dicho de otro modo, "*The system breaks down because it cannot detect what, to humans, are very obvious contradictions*" (Panton et al. 2006: 2). Desde el primer momento, se puede advertir que las máquinas carecen de sentido de la "obviedad", de la capacidad de inferir, o de aprender a través de la experiencia: lo que los humanos conocen como *sentido común*. Se trataría de dotar a la máquina de sentido común para que fuese capaz de superar esa fragilidad ("*brittleness*", arriba mencionada) y ser capaz de razonar. La cuestión es, no obstante, cómo se puede dotar de sentido común a algo que no es humano, cuya capacidad de cognición no viene concedida de manera natural, ni parece que sea capaz de poder evolucionar sin la contribución del ser humano. He ahí, precisamente, la respuesta por la que muchos de los investigadores de este campo han abogado: el ser humano es el único que puede otorgar a la máquina ese sentido común. Solamente dotando de sentido común a la máquina, ésta será capaz de resolver cuestiones basadas en el conocimiento semántico de la consulta: la máquina ya no devolverá tan sólo secuencias de letras, sino respuestas.

A continuación, se presentarán diversos modos y teorías para entender la tarea de crear estos grandes repositorios de información sobre el mundo, de conocimientos básicos humanos: de sentido común. No obstante, antes de adentrarnos en repositorios concretos, se ha de definir qué son exactamente aquellos a los que nos referimos como ontologías y bases de conocimiento.

1.2. QUÉ ES UNA ONTOLOGÍA VS. UNA BASE DE CONOCIMIENTO

Posiblemente existen tantas definiciones de ontología como autores trabajando en este campo de estudio, lo que muchas veces provoca que algunas de ellas, cuando se posicionan frente a otras, resulten no sólo disímiles sino, incluso, contradictorias. La mayor parte de las veces, la definición que cada autor realiza del término ontología está íntimamente relacionada con la ontología en sí que éste haya creado o en la que esté trabajando, por lo que, de algún modo, se podría concluir que hay tantas definiciones posibles como ontologías existen, lo que nos impediría listar la totalidad de ellas en este trabajo por obvios motivos de espacio. No obstante, algo que parece común es que, tanto las ontologías como las bases de conocimiento, pretenden aglutinar información relevante acerca de un ámbito o dominio del mundo, las entidades que lo forman, las características que poseen estas entidades, los eventos que ocurren y las relaciones que tienen lugar entre todos estos conceptos. A modo de comienzo y en relación con este planteamiento, podríamos citar a Noy y McGuinness (2001), quienes expresan que:

> "*an **ontology** is a formal explicit description of concepts in a domain of discourse (**classes** (sometimes called **concepts**)), properties of each concept describing various features and attributes of the concept (**slots** (sometimes called **roles** or **properties**)), and restrictions on slots (**facets** (sometimes called **role restrictions**))*." Noy y McGuinness (2001: 3).

Otra definición podemos encontrarla en Uschold y Gruninger (1996):

> *"Ontology is the term used to refer to the shared understanding of some domain of interest which may be used as a unifying framework to solve* […] *problems* […]*"*. Uschold y Gruninger (1996: 103).

Una tercera definición que se puede presentar, entre las posibles y numerosas existentes, es la encontrada en Bateman (1991). Se trata de una definición bastante inclusiva, ya que define mostrando una lista de posibles funciones que una ontología puede o debe cumplir:

> *"The following list gives an idea of the range of functions adopted in NLP. Ontologies are often expected to fulfill at least one (and often more) of:*
>
> • *organizing 'world knowledge',*
>
> • *organizing the world itself,*

> *• organizing 'meaning' or 'semantics' of natural language expressions,*
>
> *• providing an interface between system external components, domain models, etc. and NLP linguistic components,*
>
> *• ensuring expressability of input expressions,*
>
> *• offering an interlingua for machine translation,*
>
> *• supporting the construction of 'conceptual dictionaries'.*
>
> *Moreover, an ontology is seen as a very general organizational device: i.e., one that provides a classification system for whatever area of application the ontology is applied to. The organizational resource offered by an ontology has to be re-usable.*" Bateman (1991:51).

Ésta última idea, la que alude a la reusabilidad de los recursos ontológicos, es una valiosa propiedad a la que más adelante volveremos.

Así pues, lo que parece ser compartido por muchas de las definiciones de ontología existentes en la literatura, es que una ontología puede definirse como una herramienta organizativa, una estructuración jerárquica de clases (conceptos) o categorías que han de estar representados semánticamente de manera formal para que la máquina sea capaz de almacenarlos y tratarlos. El lenguaje de representación que se utiliza para codificar esta información difiere entre las diferentes ontologías y bases de conocimiento, por lo que más adelante se tratará sobre este punto en concreto (cf. sección 1.5). Es importante reseñar que no se ha de confundir una ontología con una taxonomía léxica, la cual consta de una clasificación de elementos léxicos (palabras) relacionados. A este respecto, es fundamental remarcar la diferencia entre un concepto y una palabra. Un concepto es una representación mental del estado de las cosas en el mundo o en la mente, similar para todos los seres humanos o, en palabras de Guarino y Giaretta (1995:30), "***conceptualization****: an intensional semantic structure which encodes the implicit rules constraining the structure of a piece of reality*". Por contra, las palabras constituyen representaciones o instanciaciones léxicas de los conceptos, las cuales varían de una lengua a otra. De este modo, Gruber (1993:199) indica que "*an ontology is an explicit specification of a conceptualization"*, lo que se ha convertido en la definición de ontología más citada en la literatura (según Corcho et al. ,2001), aunque los mismos autores apuntan que otras definiciones son válidas también, como por ejemplo la de Guarino y Giaretta (1995), quienes afirman que "*an ontology is a formal theory that constrains the possible*

conceptualisations of the world" (Guarino y Giaretta, 1995, referenciados en Corcho et al. 2001:10). No obstante, nótese que la noción de conceptualización está presente en ambas definiciones, lo que pone de manifiesto la relevancia de la misma.

Por otro lado, una base de conocimiento se podría definir como una base de datos dinámica, representada formalmente (para que sea tratable por la máquina, como se ha mencionado anteriormente) y que ha de estar dotada de un motor de razonamiento por inferencia. Esta herramienta llamada motor de razonamiento sería la responsable del nivel heurístico del sistema, es decir, de razonar de manera eficiente con aquellos datos que el sistema conoce gracias a su previa introducción manual o automática. No obstante, al igual que ocurre con la noción de ontología, podemos encontrar toda una cornucopia de definiciones para el término *base de conocimiento*, que, de nuevo, suelen estar vinculadas al trabajo particular de cada autor con dichas herramientas. En Corcho et al. (2001) se explicita este hecho del siguiente modo:

> "*Other definitions emerge as a consequence of how their authors build and use ontologies. We can distinguish here between top-down or bottom-up approaches depending on what was first the ontology or the KB*." Corcho et al. (2001:10).

Con objeto de mantener el paralelismo con los autores anteriormente citados, encontramos en Noy y McGuinness (2001) la siguiente referencia, que enlaza con su definición de ontología y la mención de "clases":

> *"We can then create a knowledge base by defining individual instances of these classes filling in specific slot value information and additional slot restrictions*" (Noy y McGuinness, 2001:3).

Puesto que se ha observado la tendencia de que cada autor defina una base de conocimiento de la manera más coherente con su metodología de trabajo, esto conlleva que diferentes autores entiendan el proceso de generación y construcción de ontologías y bases de conocimiento de maneras inversas: la ontología puede ser el punto de partida sobre el que la base de conocimiento se construye, se sostiene y de donde se destilan los datos necesarios para el proceso de razonamiento; o bien la ontología puede ser un producto derivado de la depuración de elementos pre-existentes en la base de conocimiento. Esta última opción, si bien puede resultar contradictoria para algunos autores, es mencionada por Corcho et al. (2001):

> "*The opposite approach was taken at the KACTUS project [Bernaras et al., 96], where the ontology is built after a process of abstraction of the content already represented in a knowledge base*." Corcho et al. (2001:10).

El hecho de decantarse por una u otra opción no dependerá más que del tipo de herramienta que se esté perfilando y de las funciones que se persiga cumplir mediante su uso. Sin embargo, se podría argumentar que, a pesar de que esta creación de recursos *ad hoc* podría resultar eficiente para la herramienta en particular que se esté utilizando, dicho planteamiento va en detrimento de la deseada reusabilidad de la misma en diferentes contextos o dominios (Periñán Pascual y Arcas Túnez, 2006; Mairal Usón y Periñán Pascual, 2009b), lo que podría catalogarse de creación de contenidos poco eficiente o, simplemente, de un inconveniente de dicha acción.

Tras haber descrito las principales diferencias entre ontologías y bases de conocimiento, con motivo de contextualizar el presente trabajo, a continuación iniciaremos la discusión sobre la herramienta concreta utilizada para este estudio: la base de conocimiento FunGramKB.

1.3. FUNDAMENTOS LINGÜÍSTICOS DE FUNGRAMKB: SEMÁNTICA PROFUNDA VS. ENFOQUE RELACIONAL

Antes de comenzar a ahondar en una descripción detallada de la estructura de la base de conocimiento FunGramKB, es pertinente mencionar los fundamentos teóricos sobre los que se sustenta dicha herramienta.

Es un hecho que no todas las herramientas ontológicas o de procesamiento del lenguaje natural (en adelante, PLN) se han provisto antes de su creación de una base lingüística robusta sobre la que apoyarse. Este suceso, si bien permite crear herramientas para el PLN con mayor rapidez y facilidad, aboca a éstas a una propensión a los fallos, ya que carecen de un sistema complejo y fundamentado de comprensión del lenguaje natural (Periñán Pascual y Mairal Usón, 2009). Asimismo, a la hora de representar el conocimiento en sistemas de PLN, se ha de elegir entre dos tipos de significado: el significado relacional (que pertenece a la semántica superficial o *surface semantics*) y el significado conceptual (perteneciente a la semántica profunda o *deep semantics*). El primero es aquel que podemos expresar mediante la asociación entre diferentes unidades

léxicas, mientras que el segundo alude a la descripción mediante rasgos semánticos o *primitivos* del contenido de cada unidad. Por consiguiente, el significado relacional no es puramente descriptivo, ya que se limita a señalar las conexiones entre unidades léxicas, sin ahondar en su contenido semántico individual (Mairal Usón y Periñán Pascual, 2009a), al contrario que el significado conceptual. De manera ilustrativa, se pueden citar herramientas como EuroWordNet, que utilizan un enfoque relacional (Periñán Pascual y Arcas Túnez, 2007b realizan una detallada comparación entre FunGramKB y EuroWordNet y las carencias que un enfoque relacional supone). Muchas de las herramientas creadas hasta ahora para sistemas de PLN sienten predilección por el enfoque relacional, puesto que las ventajas que éste ofrece a corto plazo son evidentes: relativa sencillez y rapidez de creación. Sin embargo, estas ventajas pueden resultar a la vez inconvenientes, puesto que esta sencillez y rapidez se transforman en imposibilidad de reusabilidad o traslación a otros sistemas, restricción de poder expresivo y alta redundancia (Mairal Usón y Periñán Pascual, 2009; Periñán Pascual y Arcas Túnez, 2007b). No obstante, pese a los aparentes impedimentos iniciales de adoptar un enfoque de semántica profunda (se requiere un gran esfuerzo en tiempo y especialización; como ejemplo, no olvidemos el anteriormente mencionado proyecto Cyc), la alta redundancia y restricción de poder expresivo se ven minimizadas significativamente, mientras que la reusabilidad se ve potenciada. Mairal Usón y Periñán Pascual (2010) también arguyen que algunas unidades conceptuales son muy difíciles de representar de manera relacional, como por ejemplo REMEMBER, LOVE o FORGET[2], por lo que en conjunto se podría decir que, tras sopesar ambas opciones, es más que razonable pensar que un modelado del conocimiento basado en la semántica profunda es un esfuerzo eficiente y amortizable.

De manera paralela a la decisión de adoptar un enfoque de semántica profunda, se ha de retornar al planteamiento enunciado al comienzo de esta sección: la necesidad de contar con una teoría lingüística válida sobre la que se sustente la base de conocimiento, con objeto de dotar a ésta de dicha validez y robustez. En el caso de FunGramKB, el modelo lingüístico que actúa como referencia para los niveles léxico y gramatical de la base de conocimiento es el Modelo Léxico Construccional[3] (MLC, o LCM por sus siglas en inglés, *Lexical Constructional Model*) (Ruiz de Mendoza Ibáñez y Mairal Usón, 2008;

[2] En mayúscula porque dichas expresiones aluden al concepto, no a la expresión léxica de cada unidad.

[3] Para más información acerca del MLC, puede visitarse la página web del grupo de investigación Lexicom, http://www.lexicom.es/

Mairal Usón y Ruiz de Mendoza Ibáñez, 2009). Este modelo constituye una reconciliación entre paradigmas funcionales y cognitivos del lenguaje, lo que supone un enriquecimiento mutuo entre ellos, al dar cabida a diversos niveles y aspectos del significado:

> "*The primary concern of the LCM is to develop a usage-based, comprehensive theory of meaning construction that aims to give explanations of how all aspects of meaning, including those that go beyond so-called core-grammar (e.g., traditional implicature, illocutionary force, and discourse coherence) interact among one another*." (Ruiz de Mendoza Ibáñez y Mairal Usón, 2008:355)

Este planteamiento es el que en la práctica une los postulados *proyeccionistas* de la Gramática del Papel y la Referencia (RRG, por sus siglas en inglés, *Role and Reference Grammar*) de Van Valin y LaPolla o de la Gramática Funcional (*Functional Grammar*, FG) de S.C. Dik, junto a la vertiente *construccionista* de Fillmore y Kay, entre otros (Mairal Usón y Ruiz de Mendoza Ibáñez, 2009). Se trata, por tanto, de combinar perspectivas de modelos funcionales como la RRG y FG mencionadas, junto a los postulados de la Lingüística Cognitiva de Lakoff y la Gramática de Construcciones de Goldberg (Ruiz de Mendoza Ibáñez y Mairal Usón, 2008). Tomando como punto de partida este enfoque hermanado de teorías lingüísticas a través del MLC, se detallará a continuación la descripción de la base de conocimiento FunGramKB.

1.4. ¿QUÉ ES FUNGRAMKB[4]?

Con el objetivo de comenzar mediante una definición explícita de FunGramKB, obsérvese el siguiente extracto de Periñán Pascual y Arcas Túnez (2004):

> "*FunGramKB is a lexicographical tool for the semiautomatic construction of a multipurpose lexico-conceptual knowledge base for a natural language processing (NLP) system* […]" (Periñán Pascual y Arcas Túnez 2004:38).

En primer lugar, FunGramKB se define como "*lexico-conceptual*" al incorporar componentes léxicos y conceptuales a través de los diferentes módulos, teniendo así cabida tanto los constructos conceptuales como las instanciaciones léxicas de estos constructos, en cada lengua natural de las incluidas en la base de conocimiento.

[4] http://www.fungramkb.com/

En segundo lugar, se puede apreciar que FunGramKB se define no como ontología, sino como base de conocimiento que integra los niveles léxico y conceptual, y que es multipropósito debido a la potencial reusabilidad de la herramienta. No en vano, se ha de recordar que la noción de reusabilidad se ha mencionado anteriormente como una característica que idealmente han de poseer este tipo de herramientas. La reusabilidad de FunGramKB alude a que se trata de una base de conocimiento ya no sólo multifuncional, que puede utilizarse en diversos ámbitos, como por ejemplo la creación de agentes o motores de búsqueda inteligentes, traducción automática o diccionarios que posibiliten una búsqueda conceptual (Mairal Usón y Periñán Pascual, 2010; Periñán Pascual y Arcas Túnez, 2006), sino también a que se trata de una base multilingüe, incluyendo en la actualidad lexicalización de conceptos en siete lenguas naturales (inglés, español, italiano, francés, alemán, búlgaro y catalán).

FunGramKB está estructurada en tres grandes niveles de información, que a su vez se subdividen en varios módulos independientes pero interrelacionados (Periñán Pascual y Arcas Túnez, 2006; Periñán Pascual y Arcas Túnez, 2010b). Estos tres grandes niveles son el **nivel léxico** (conocimiento lingüístico), el **nivel gramatical** (conocimiento acerca de esquemas construccionales) y el **nivel conceptual** (conocimiento no lingüístico). Como se ha mencionado, cada nivel se subdivide en varios módulos de la manera que muestran las figuras que se incluyen a continuación:

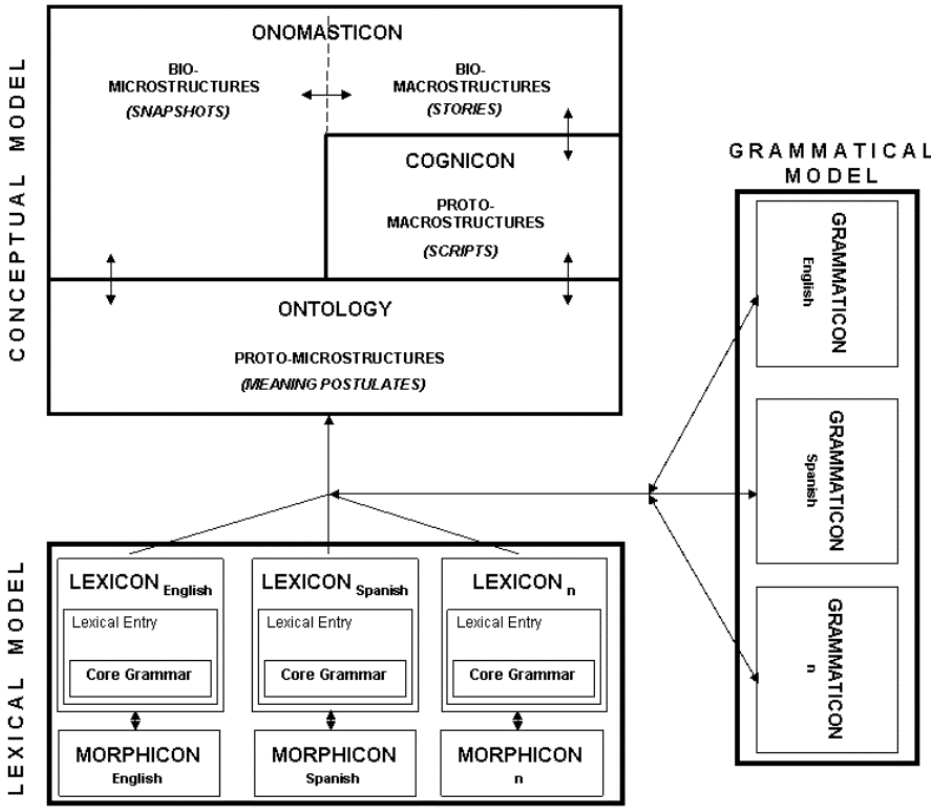

Figura 1. Arquitectura de FunGramKB[5]

Figura 2. Vista parcial de *FunGramKB editor*, mostrando los diferentes módulos y algunos recursos lexicográficos disponibles

[5] Figura tomada de http://www.fungramkb.com/

Si se presta atención a los diferentes módulos lingüísticos, comenzando por el *nivel léxico*, se puede explicar sus componentes como prosigue: el **Lexicón** es el módulo que contiene la información morfosintáctica, pragmática y colocacional de las unidades léxicas. Se trata, por tanto, de información que ayuda a utilizar dichas unidades de manera correcta formalmente en el discurso. El **Morficón** contiene aquellas reglas que afectan a la morfología flexiva, como por ejemplo las inflexiones verbales o conjugaciones, o las concordancias de género y número. El siguiente módulo lingüístico, el *nivel gramatical*, contiene el **Constructicón**, que almacena los diferentes Gramaticones de las diferentes lenguas naturales, los cuales constituyen esquemas de construcciones que permiten a la RRG (*Role and Reference Grammar*), teoría lingüística mencionada con anterioridad, construir el interfaz entre sintaxis y semántica, denominado enlace (*linking algorithm*), para representar un texto de entrada en lenguaje natural mediante una estructura lógica. Asimismo, en FunGramKB esta estructura lógica se ve mejorada mediante un nuevo formalismo denominado "estructura lógica conceptual" (CLS por sus siglas en inglés, *Conceptual Logical Structure*), que maximiza la carga informativa y reduce la redundancia (Periñán Pascual y Arcas Túnez, 2010b).

El otro gran módulo que compone FunGramKB es el *nivel conceptual*. Como se ha indicado con anterioridad, el nivel conceptual se refiere a aquellas representaciones prototípicas de la realidad que el individuo recrea en su mente, no de las palabras con que se describe de manera lexicalizada dicha realidad. De este modo, mientras que el módulo de nivel léxico es particular para cada lengua natural (ha de crearse, pues, un Lexicón para el inglés, otro para el español, y así sucesivamente para cada lengua con cada uno de los tres componentes del nivel léxico), el nivel conceptual es común a todos los lenguajes naturales, puesto que no está basado en conocimiento sobre las palabras sino en conocimiento sobre el mundo (Periñán Pascual y Arcas Túnez, 2006). Es por esto que se puede afirmar que el nivel conceptual de FunGramKB es universal, en tanto en cuanto los conceptos que recopila y categoriza son comunes al mundo: "*FunGramKB ontology takes the form of a universal concept taxonomy, where 'universal' means that every concept we can imagine has an appropriate place in this ontology*." (Periñán Pascual y Arcas Túnez, 2007a:199). Se trata de la lexicalización de estos conceptos, la manera de organizarlos y de aglutinarlos en expresiones lingüísticas en las diferentes lenguas naturales lo que difiere entre ellas. No obstante, el hecho de que el módulo conceptual de FunGramKB sea denominado universal no lo exime de ser lingüísticamente motivado, mas no

lingüísticamente dependiente. Esto significa que cada uno de los conceptos introducidos en el módulo conceptual (particularmente en la Ontología) tiene necesariamente al menos una unidad léxica cuyo significado no coincide con ninguno de los postulados de significado ya presentes en la base de conocimiento. Asimismo, este proceso asegura que cada nueva unidad léxica que pueda surgir en el futuro, como consecuencia de la introducción de nuevas lenguas naturales en FunGramKB, tendría cabida dentro de ella tras un proceso de negociación. La posibilidad de introducción de estos nuevos conceptos, lexicalizados de una manera particular en una lengua natural, además, contribuye a eliminar una posible carga subjetiva provocada por las propias lenguas maternas y contextos culturales de los ingenieros del conocimiento que pueblan la base (Periñán Pascual y Arcas Túnez, 2010b).

El nivel conceptual de FunGramKB se compone de tres módulos (Periñán Pascual y Arcas Túnez, 2007a; Periñán Pascual y Arcas Túnez, 2010b): la **Ontología**, que es una estructura jerárquica de conceptos usados por las personas para describir cualquier situación cotidiana; el **Cognicón**, que almacena conocimiento procedimental en forma de secuencias temporales o "guiones", y que está basado en el modelo temporal de Allen (Allen, 1983; Allen y Ferguson, 1994); y el **Onomasticón**, que almacena conocimiento episódico acerca de entidades (p. ej., personas, ciudades, lugares, acontecimientos, etc.) en forma de bio-estructuras (Periñán Pascual y Mairal Usón, 2010). Recordando lo mencionado al comienzo de este trabajo sobre las definiciones de ontología (cf. sección 1.2), se puede observar ahora que, en el ámbito de FunGramKB, *Ontología* se corresponde con uno de los módulos del nivel conceptual de la base de conocimiento: aquel módulo que recoge una jerarquía de los conceptos habituales que un hablante utiliza para expresar situaciones cotidianas. No obstante, no se ha de confundir esta serie de conceptos con las lexicalizaciones de los mismos. Esto es, los conceptos serían comunes a todas las lenguas naturales, pero la lexicalización de éstos es lo que variaría entre ellas. Sirva como ejemplo el concepto que en lenguaje natural podríamos definir como "*Mueble, por lo común de madera, que se compone de una o de varias tablas lisas sostenidas por uno o varios pies, y que sirve para comer, escribir, jugar u otros usos*"[6]. Una posible lexicalización en inglés podría ser *table* o *desk*; en español *mesa* (o también *escritorio*, por ejemplo), en alemán *Tisch*, y así sucesivamente. De este modo, *Ontología* en el marco de FunGramKB es uno de los elementos sobre los que se soporta la base de conocimiento, siendo así una parte

[6] Definición de "mesa" por el Diccionario RAE, 22ª edición, accesible online: http://buscon.rae.es/draeI/SrvltConsulta?TIPO_BUS=3&LEMA=mesa

más de dicha base. Esto enlaza con la idea mostrada anteriormente acerca de la concepción de *ontología* mantenida por cada autor, la cual habitualmente estará relacionada con el uso que éste haya hecho de las ontologías, o la herramienta particular que utilice o que haya creado.

A continuación se describirán brevemente los tres módulos del nivel conceptual de FunGramKB, puesto que este será el módulo en el que se centrarán las tareas del presente trabajo, más concretamente en el Onomasticón.

La **Ontología** es el elemento central de FunGramKB. Comprende tres tipos generales de conceptos (llamados metaconceptos y señalados con el símbolo (#)): entidades, (# ENTITY), eventos (# EVENT) y cualidades (# QUALITY). Estos tres conceptos organizan la dimensión cognitiva de los nombres, verbos y adjetivos, respectivamente (Periñán Pascual y Arcas Túnez, 2007a). Estas tres dimensiones conceptuales se relacionan entre sí a través de los llamados postulados de significado (*Meaning Postulates*, MPs), que más adelante detallaremos con más profundidad. El siguiente nivel bajo los metaconceptos se compone de los conceptos básicos, precedidos del símbolo (+), y por debajo se encuentran los conceptos terminales, identificados mediante el símbolo ($). A modo de aclaración, podemos observar la figura que ilustra la descripción de esta jerarquía de unidades conceptuales en Periñán Pascual y Arcas Túnez (2007a):

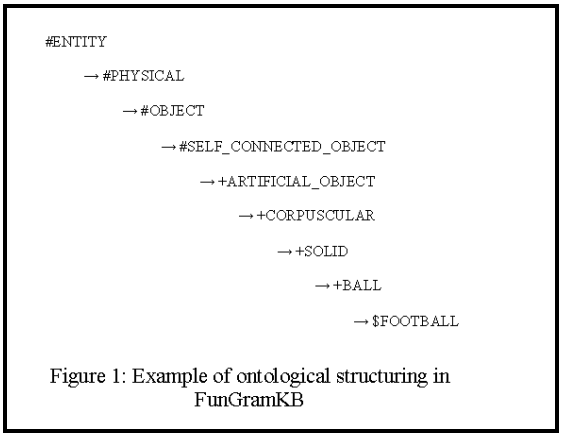

Figura 3. Ejemplo de estructuración ontológica en FunGramKB (Periñán Pascual y Arcas Túnez, 2007a:199)

El **Cognicón** es el módulo que almacena conocimiento procedimental en forma de secuencias temporales o "guiones", y que está basado particularmente en el modelo temporal de Allen (Allen, 1983). En FunGramKB, se estructura como una serie de predicaciones dentro de un marco lineal temporal. Un guión, por tanto, comprende varias predicaciones, y cada una de ellas se entiende como un evento E, tratado como un intervalo entre un par de puntos temporales: el punto temporal *i*, que es el inicio de la acción, y el punto temporal *t*, que es su fin. Así como Allen había previsto la posibilidad de diversas relaciones entre estos eventos (simultaneidad, posterioridad o solapamiento, entre otros), también quedan éstas reflejadas en FunGramKB (Periñán Pascual y Arcas Túnez, 2010b).

Finalmente, el **Onomasticón** está formado por aquellas entidades que componen nuestro conocimiento enciclopédico o histórico, por ejemplo personas, lugares, acontecimientos, etc., en forma de bio-estructuras. La naturaleza y características de este módulo serán desarrolladas manera extensa en la sección 2.1 del presente trabajo, motivo por el cual a continuación se trasladará la discusión al lenguaje de representación utilizado en FunGramKB.

1.5 COREL COMO LENGUAJE DE REPRESENTACIÓN: POSTULADOS DE SIGNIFICADO Y MARCOS TEMÁTICOS EN FUNGRAMKB

Como se ha mencionado anteriormente, las dimensiones conceptuales dentro de FunGramKB se relacionan a través de los llamados postulados de significado. La creación de estos constructos conceptuales responde a la necesidad de representar el conocimiento, sin perjuicio de que en los módulos lingüísticos de FunGramKB el Modelo Léxico Construccional siga, como modelo teórico escogido, impregnando estos módulos de la base de conocimiento. No obstante, además de la importancia que supone el contar con un modelo teórico robusto tras el sistema, FunGramKB ha de plantearse, como toda base de conocimiento, el lenguaje de representación mediante el cual introducirá los constructos conceptuales en dicha base. Este punto también supone una inflexión clave a la hora de perfilar la herramienta, por lo que merece la pena reseñar las diferencias principales con otras bases de conocimiento y ontologías existentes.

Es necesario hacer un pequeño ejercicio de reflexión histórica antes de comprender la elección de un lenguaje de representación u otro para la arquitectura de una base de conocimiento u ontología. Como se ha señalado más arriba, la necesidad de contar con una teoría lingüística que actúe de fuste para la creación de la herramienta no siempre ha sido una realidad. Dada la naturaleza de estas herramientas, las ontologías y las bases de conocimiento han sido iniciadas desde su etapa germinal más por ingenieros de la computación y profesionales de la informática que por lingüistas o, al menos, pocas han sido fruto de la cooperación entre ambas disciplinas si miramos desde una perspectiva diacrónica. Esta necesidad obvia de aunar las diferentes perspectivas es algo que a día de hoy puede parecer natural, pero que no siempre ha sido así (cf. Mairal Usón y Periñán Pascual, 2009a). Puesto que los ingenieros que han creado las herramientas y su arquitectura pertenecían al campo de la computación, parece coherente que los lenguajes de representación[7] escogidos pertenezcan al conjunto de los usados tradicionalmente en este campo de estudio, así como el hecho de contar con una base teórica más computacional que lingüística. Es así que la lógica de primer orden, por ejemplo, ha sido utilizada en numerosas ontologías y bases de conocimiento. Sin ir más lejos, CycL, el lenguaje formal utilizado por el mencionado proyecto Cyc, basa su sintaxis en el cálculo de predicados de primer orden, así como en KIF (*Knowledge Interchange Format*). Otros lenguajes formales están basados también en la Semántica de Marcos (*Frame Semantics*) de Minsky (1974)[8], como por ejemplo FLogic, siendo esta teoría combinada también con KIF para lenguajes como Ontolingua. Incluso, el proyecto ConceptNet (Liu y Singh, 2004a; Liu y Singh, 2004b) da un paso más allá y postula la representación del conocimiento utilizando lenguaje natural, debido a la gran flexibilidad que éste ofrece frente a la lógica, visión ésta que podría ser rebatida por muchos. Si bien es cierto que la lógica parece conllevar, dentro de su rigor y exactitud, la desventaja anexa de ser demasiado estricta en algunos contextos (especialmente si no permite un razonamiento no-

[7] No se ha de confundir lenguaje de representación con lenguajes de programación sobre los que se sostiene el sistema. Los segundos serían los que sirven para implementar la herramienta computacionalmente (*caja negra*), mientras que los primeros son los que el ingeniero del conocimiento utiliza para introducir constructos conceptuales relacionados con las unidades, y los que, a su vez, el usuario utilizará en su interfaz con la base de conocimiento a la hora de recuperar la información que ésta pueda ofrecerle.

[8] Si bien el propio Minsky (2000) recomienda el uso de múltiples lenguajes para representar el conocimiento: "*There is no best way to represent knowledge. The present limitations of machine intelligence stem largely from seeking unified theories incapable of reasoning well, while our purely symbolic logical systems lack the uncertain, approximate linkages that can help us make new hypotheses. Human versatility must emerge from a large-scale architecture in which each of several different representations can help overcome the deficiencies of the other ones*." (Minsky, 2000: 71).

monotónico), el extremo de utilizar lenguaje natural para la representación del conocimiento podría dar pie a una excesiva ambigüedad, lastre adherido a la cualidad de poder expresivo y flexibilidad. Periñán Pascual y Arcas Túnez (2007a) lo expresan de este modo:

> "*Dik (1997) proposes using words from the own language when describing meaning postulates, since meaning definition is an internal issue of the language. However, this strategy contributes to lexical ambiguity due to the polysemic nature of the defining lexical units. In addition, describing the meaning of words in terms of other words leads to some linguistic dependency (Vossen, 1994). Instead, FunGramKB employs concepts for the formal description of meaning postulates, resulting in an interlanguage representation of meaning*." (Periñán Pascual y Arcas Túnez, 2007a:200).

Mediante esta mención a FunGramKB, se retomará a continuación la discusión y definición del lenguaje formal mediante el cual se representan los constructos conceptuales en esta base de conocimiento.

El lenguaje de representación del conocimiento en FunGramKB responde al nombre de **COREL** (***Co****nceptual* ***Re****presentation* ***L****anguage*). Se trata de un lenguaje de interfaz para la representación formal del conocimiento conceptual. Esta característica hace que COREL sea, a la vez que un lenguaje formal robusto tratable por la máquina, un código fácil de interpretar por el usuario, aunando de este modo rigor sistemático y funcionalidad. La manera de representar mediante COREL los constructos cognitivos que son los conceptos, se compone de dos estructuras complementarias: los marcos temáticos y los postulados de significado (Periñán Pascual y Mairal Usón, 2009; 2010).

Los **marcos temáticos** (*Thematic Frames*, TF, en FunGramKB) son estructuras conceptuales que explicitan el número y tipo de participantes que intervienen en la situación prototípica[9] retratada por el concepto, lo que también incluye las preferencias de selección de los participantes (Periñán Pascual y Mairal Usón, 2009). Gracias a los marcos temáticos, el concepto encuentra delimitado el marco en el que la situación tiene lugar. Para ilustrar esto, podemos tomar el ejemplo mostrado en Periñán Pascual y Mairal Usón

[9] Recordemos que el MLC, marco teórico relacionado con FunGramKB, contempla teorías de la lingüística cognitiva, por lo que esta *prototipicidad* de la que se habla ha de entenderse en el sentido de la teoría de prototipos, tratada, entre otros, por Lakoff y Langacker (Lakoff, 1987; Langacker, 1987; 1991) e iniciada por Rosch (1973).

(2009): el concepto +PULL_00, el cual quedaría lexicalizado en inglés como *pull* o *draw*, y en español con unidades léxicas como *tirar* o *estirar*:

(1)

(x1: +HUMAN_00 ^+ANIMAL_00)Agent (x2:+CORPUSCULAR_00)Theme (x3)Location (x4)Origin (x5)Goal

La equivalencia, según explican Periñán Pascual y Mairal Usón (2009), en lenguaje natural de esta estructura sería la siguiente:

> *"Thus, the thematic frame (1) describes a prototypical cognitive scenario in which "entity$_1$ (Agent) moves entity$_2$ (Theme) from one place (Origin) to another (Goal), there also being a place (Location) along which entity$_2$ moves"."* (Periñán Pascual y Mairal Usón, 2009:267).

Podemos añadir que, asimismo, según las preferencias de selección presentes en el marco temático, se sabe que la entidad$_1$ (el agente) es o bien un humano, o bien un animal, y que la entidad$_2$ (el tema) ha de ser una entidad sólida tridimensional ("+CORPUSCULAR_00").

Por otro lado, una vez que se han detallado los participantes en la situación prototípica que el concepto evoca, se requiere el **postulado de significado** (*Meaning Postulate*, MP, en FunGramKB) de dicho concepto, que es la estructura que define el mismo mediante una semántica profunda. Un postulado de significado es una secuencia de una o varias predicaciones, lógicamente conectadas, que son constructos conceptuales que evocan los rasgos genéricos del concepto (cf. Periñán Pascual y Arcas Túnez, 2004). Si continuamos con el ejemplo del concepto +PULL_00, observamos que Periñán Pascual y Mairal Usón (2009) expresan el postulado de significado de este modo:

(2)

+((e1: +MOVE_00 (x1)Agent (x2)Theme (x3)Location (x4)Origin (x5)Goal (f1: +HAND_00 ^ +MOUTH_00)Instrument (f2: (e2: +SEIZE_00 (x1)Theme(x2)Referent)) Condition) (e3:+BE_00 (x1)Theme (x5)Referent))

La estructura en (2), como se puede observar, incluye una co-indización de las entidades (llamadas *argumentos*) señaladas en el marco temático mostrado en (1), que podemos identificar mediante la letra *x* seguida de un número. Cada una de las

predicaciones que enumeran una característica genérica del concepto está designada por la letra *e*, seguida de un número. En caso de que aparezca el signo (+) precediendo a una de estas predicaciones, fuera del paréntesis, significará que dicha predicación es estricta, la cual permitirá exclusivamente la herencia monotónica. Por monotónica entendemos que no hay excepciones al contenido de dicha predicación, dentro de todas las situaciones, prototípicas o no, que puedan identificarse mediante el concepto que se está describiendo. Otra de las características de los postulados de significado es que pueden contener elementos *satélites*, designados por la letra *f*, que aportan información periférica acerca de la situación conceptual descrita por las predicaciones. Dependiendo de la centralidad del papel en el evento que se está describiendo, se le asignará el papel de *argumento* o de *satélite* dentro del postulado de significado (Periñán Pascual y Mairal Usón, 2010), por lo que se observa que, nuevamente, nociones como centralidad hacen acto de presencia, en clara referencia a la teoría de prototipos anteriormente mencionada (cf. Nota 9). De este modo, la equivalencia del postulado de significado del ejemplo (2) en lenguaje natural sería la siguiente: "*A person or animal moves something towards themselves with their hand or mouth, providing that they hold it firmly*." (Periñán Pascual y Mairal Usón, 2009:268).

Por consiguiente, se manifiesta que cualquier concepto, previamente expresado de manera lexicalizada en una lengua natural, puede ser representado en FunGramKB mediante el lenguaje COREL, lo que constituye un reflejo del compromiso de universalidad de esta base de conocimiento mencionado anteriormente (Periñán Pascual y Arcas Túnez, 2010a).

CAPÍTULO 2. MARCO TEÓRICO Y HERRAMIENTAS UTILIZADAS

2.1. QUÉ ES EL ONOMASTICÓN: ARQUITECTURA

Como se ha indicado en el anterior capítulo, el presente estudio versará acerca de la metodología y sistema que permitan poblar de manera semi-automática el Onomasticón de FunGramKB. A continuación se detallará la estructura y arquitectura del mismo, con objeto de definir el marco teórico del trabajo que es necesario llevar a cabo para poblarlo.

El Onomasticón de FunGramKB está encuadrado dentro del nivel conceptual de la base de conocimiento, junto a la Ontología y al Cognicón. El Onomasticón, como se ha reseñado con anterioridad de manera breve, almacena la información relativa a las instancias de entidades y eventos, en forma de bio-estructuras. Con la finalidad de comprender qué es a lo que se apunta de manera precisa mediante la expresión *bio-estructura*, es necesario regresar al carácter prototípico del conocimiento almacenado en los módulos conceptuales que son la Ontología y el Cognicón. Dada la prototipicidad que poseen, llamamos a las estructuras que forman la Ontología y el Cognicón *proto-estructuras*, ya que se deben a la generalidad y no a la particularidad. No obstante, a pesar de reflejar esta generalidad, en el marco de una situación prototípica, cuando cabe la posibilidad de que alguna de las características de la misma pueda, en algún momento hipotético del tiempo, ser diferente de las expresadas en la predicación, se utiliza la etiqueta de "rebatible", lo que permitirá una herencia no-monotónica (y que se indica en lenguaje COREL mediante el empleo del signo (*) delante de la predicación en cuestión). Por ejemplo, si hablamos de un pájaro, una característica prototípica de este tipo de animales es "puede volar", pero, puesto que es posible que exista un pájaro que, a pesar de serlo, no sea capaz de volar (p.ej. un pingüino o un avestruz), ésta será entonces una característica rebatible.

Por el contrario, el Onomasticón refleja la situación opuesta, donde la prototipicidad deja paso a la especificidad, y puesto que se trata de entidades precisas existentes, se utiliza la etiqueta de *bio* para nombrar a estas estructuras. De este modo,

mientras que un concepto (entidad) de la Ontología es +SONG_00, una entidad del Onomasticón es %HEY_JUDE_00[10].

Además del parámetro de la prototipicidad que acaba de mostrarse, los esquemas conceptuales en FunGramKB también se clasifican conforme a otro parámetro: la temporalidad (Periñán Pascual y Arcas Túnez, 2010b). Teniendo en cuenta esta característica, el conocimiento dentro de los esquemas conceptuales puede presentarse de manera temporal o atemporal. Si se hace de manera temporal, significará que ese conocimiento se presenta dentro de un marco de tiempo (macroestructuras). Por ejemplo, la biografía de una persona (entidad del Onomasticón) o un guión de los reflejados en el Cognicón poseerían la propiedad de ser estructuras temporales. Por otro lado, cabe la posibilidad de que las estructuras conceptuales representen el conocimiento de manera atemporal (microestructuras), como sucedería con una característica aislada de una entidad del Onomasticón o un postulado de significado de la Ontología. La convergencia de estos dos parámetros, prototipicidad y temporalidad, resultan en la creación de una tipología de estructuras conceptuales compuesta por proto-microestructuras, proto-macroestructuras, bio-microestructuras y bio-macroestructuras. Si se coloca en una matriz esta combinación de parámetros, se obtendrá la siguiente tabla que lo ilustra:

		TEMPORALIDAD	
		-	+
PROTOTIPICIDAD	+	Proto-microestructura (Postulado de significado)	Proto-macroestructura (Guión)
	-	Bio-microestructura (Retrato)	Bio-macroestructura (Historia)

Tabla 1. Tipología de esquemas conceptuales en FunGramKB (Periñán Pascual y Arcas Túnez, 2010b)

[10] Nótese el símbolo (%) utilizado para identificar las entidades del Onomasticón, diferente de los símbolos (#), (+) o ($), que identifican a las entidades pertenecientes a la Ontología como metaconceptos, conceptos básicos o conceptos terminales, respectivamente.

Además de estar organizadas de la manera arriba ilustrada, estas cuatro estructuras se integran en la base FunGramKB, lo que puede observarse representado de manera gráfica en la figura siguiente:

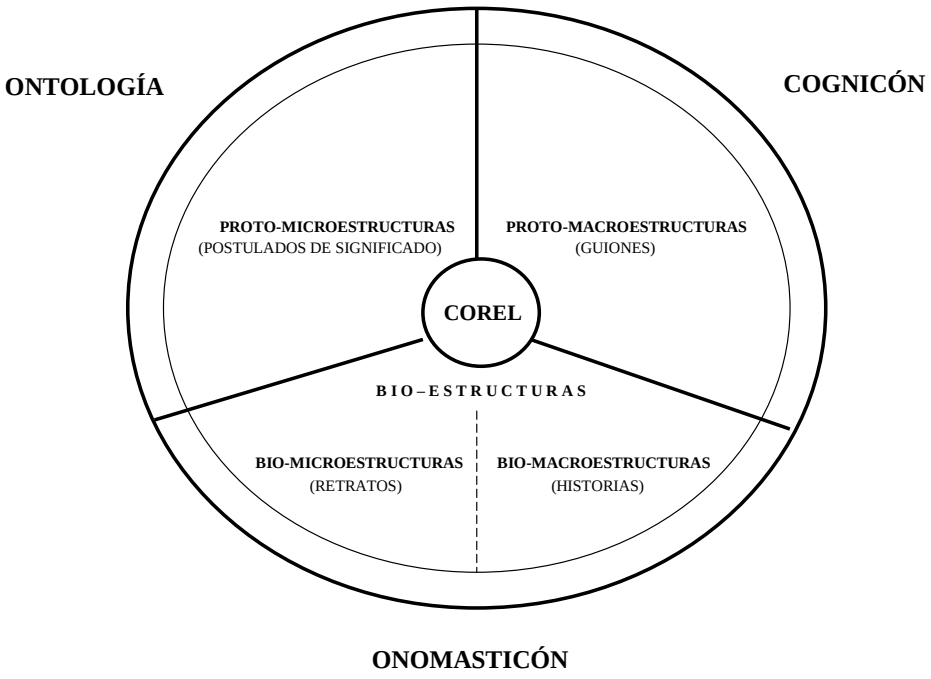

Figura 4. El Planeta Cognitivo (Periñán Pascual y Mairal Usón, 2010)

Esta figura muestra la necesaria interrelación entre los tres módulos del nivel conceptual de FunGramKB a través de un mismo lenguaje de representación, COREL (cf. sección 1.5), ya que, al igual que ocurre con los componentes de la memoria humana a largo plazo, un sistema de PLN que persiga permitir con éxito el razonamiento ha de prever que los componentes que lo forman puedan relacionarse entre ellos a través de un mismo lenguaje (Periñán Pascual y Arcas Túnez, 2010b; Periñán Pascual y Mairal Usón, 2010).

La Tabla 1 y Figura 4 arriba mostradas sirven de contexto clave a la hora de describir el Onomasticón y los elementos que éste alberga. De este modo, si se toma una entidad como, por ejemplo, *Elvis Presley* (%ELVIS_PRESLEY_00), dependiendo del tipo de característica que se desee reflejar, se necesitará un tipo u otro de estructura conceptual. Por tanto, si se desea describir la profesión de cantante de Elvis, será preciso emplear una

bio-microestructura, ya que dicha propiedad puede encuadrarse como uno de los muchos rasgos que, de manera relativamente aislada del resto de características, describen a Elvis en un momento concreto del tiempo, como serían, entre otras muchas, la profesión de actor, su color de pelo o su estatura (cf. Periñán Pascual y Arcas Túnez, 2010b). Es por ello que este tipo de microestructuras en el Onomasticón se denominan "retratos" (*snapshots* en su versión en inglés), por el hecho de asemejarse a una fotografía estática que se toma en un momento particular. Sin embargo, si se desea describir la biografía de Elvis, es necesario hacer uso de una bio-macroestructura, las llamadas "historias" (*stories*) por obvias razones: contienen elementos que han de ser integrados dentro de un esquema temporal determinado. En el ejemplo que nos ocupa, se podrían enumerar como historias, entre otras, la fecha de nacimiento o fallecimiento del cantante, ya que se trata de elementos cuya existencia se encuadra en un marco temporal establecido, y cuyo orden no es posible alterar.

Otro ejemplo al que se puede aludir es al de una entidad como el Taj Majal. De nuevo, si se persigue describir una propiedad como "hecho de mármol" o "mausoleo", se haría mediante una bio-microestructura (retrato), mientras que el relato de su proceso de construcción habría de realizarse mediante una historia (bio-macroestructura). Esto se puede ilustrar en lenguaje COREL a través del ejemplo (3), originalmente mostrado en Periñán Pascual y Arcas Túnez (2010b) acerca de la entidad "%TAH_MAHAL_00". El equivalente en lenguaje natural de las predicaciones representadas en (3) se muestra en (3b):

(3)

+(e1: +BE_02 (x1: %TAH_MAHAL_00)Theme (x2: %INDIA_00)Location)
*(e2: +BE_01 (x1)Theme (x3: +WHITE_00 & $MARBLE_00)Attribute)
*(e3: +COMPRISE_00 (x1)Theme (x4: 1 $DOME_00 & 4+TOWER_00)Referent)

(3b)

El Taj Majal está ubicado en India.
Su principal material es el mármol blanco.
El Taj Majal tiene una cúpula principal y cuatro torres.

Estas tres predicaciones acerca del Taj Majal pertenecen a la categoría "retrato", ya que las tres enumeran rasgos atemporales de la entidad. Sin embargo, apreciamos que sólo la primera de ellas porta el símbolo (+) al frente, lo que indica que tan sólo esa predicación es estricta. Las otras dos predicaciones son rebatibles, ya que si el Taj Majal encontrara destruida una de sus torres o la cúpula principal por cualquier circunstancia, o si alguna de éstas se sustituyera por un elemento hecho de otro material diferente del mármol blanco, el Taj Majal seguiría considerándose la misma entidad. No obstante, si el Taj Majal no se encontrase en la India, entonces ya no se trataría de la misma entidad, sino de una reproducción del mausoleo en otra ubicación.

Por otro lado, y continuando con el mismo ejemplo, es posible también representar características temporales del Taj Majal. Es el caso de las siguientes predicaciones en (4) y su equivalente en lenguaje natural en (4b) (Periñán Pascual y Arcas Túnez, 2010b):

(4)

+(e1: past +BUILD_00 (x1)Theme (x2: %TAH_MAHAL_00)Referent (f1: 1633)Time)
+(e2: past +BE_00 (x2)Theme (x3:%WORLD_HERITAGE_SITE_00)Referent (f2: 1983)Time)

(4b)

El Taj Majal fue construido en 1633.
El Taj Majal se convirtió en Patrimonio de la Humanidad por la UNESCO en 1983.

Los conceptos representados en (4) constituyen historias, ya que, además de ser predicaciones estrictas, indicado por el signo (+) frente a cada una de ellas (no es posible cambiar el año de construcción, ni aquel en el que fue declarado Patrimonio de la Humanidad), pertenecen a un esquema temporal en el que están encuadradas de una manera precisa y determinada. Por consiguiente, se observa a través de los ejemplos anteriormente mencionados cómo una misma entidad puede compartir retratos e historias que representen rasgos conceptuales que la definan.

Mediante la comparación de los dos tipos de estructuras del Onomasticón (retratos e historias) con los elementos de los otros dos módulos conceptuales de FunGramKB, la Ontología y el Cognicón, es comprensible que los postulados de significado de la Ontología se consideren proto-microestructuras (por su carácter estático y atemporal), mientras que los guiones del Cognicón son proto-macroestructuras (ya que los eventos dentro de estos guiones siguen un esquema temporal determinado).

No obstante, y paralelamente a las similitudes entre módulos conceptuales que han sido enumeradas hasta ahora, uno de los principales rasgos distintivos del Onomasticón frente a Ontología y Cognicón es que la manera de poblar este módulo no es enteramente manual, como sucede en los otros dos módulos, sino que se realiza de manera semi-automática (Periñán Pascual y Arcas Túnez, 2010b). Si bien más adelante se dedicará un capítulo entero a describir la metodología y protocolo para iniciar esta población semi-automática, así como el análisis del producto resultante, es pertinente en este momento hacer una mención a los sistemas y estructuras pertenecientes a otros proyectos que podrían asemejarse al Onomasticón de FunGramKB y, lo que es más importante, cuáles de ellos van a desempeñar un papel relevante como recursos para la población semi-automática del Onomasticón de FunGramKB.

2.2. ESTRUCTURAS SIMILARES: YAGO Y DBPEDIA

Dentro del vasto universo ontológico mencionado en el capítulo 1 del presente trabajo, podemos destacar dos proyectos concretos que se relacionan con el Onomasticón de FunGramKB, en tanto en cuanto ambos versan sobre entidades y la manera de categorizarlas: el proyecto YAGO y el proyecto DBpedia. Una importante razón por la cual se han escogido estos dos proyectos en particular, en lugar de otros similares, responde a que ambos toman la información acerca de las entidades de fuentes ricas en datos y, lo que es primordial, que se hallan en permanente actualización, lo que permite que se dé cabida en dichas fuentes a nuevas entidades que puedan crearse, así como que la información relativa a las entidades existentes pueda ser actualizada a medida que ésta cambie. Una de las más profusas fuentes de información que en la actualidad almacena conocimiento sobre millones de entidades, en forma de más de 15 millones de artículos, es el proyecto Wikipedia.

Wikipedia[11], un proyecto de la Fundación Wikimedia, es una enciclopedia multilingüe (posee artículos en 271 lenguas naturales), libremente accesible en la web, y editada de manera cooperativa por voluntarios de todo el mundo, bajo la supervisión de un grupo de administradores de cada edición que velan por el mantenimiento de los artículos y el cumplimiento de las normas. Este conocimiento, no obstante, a pesar de su dimensión y organización jerárquica, carece de estructuración conceptual, por lo que parece razonable pensar que reutilizarlo para dotarlo de este tipo de contenido e integrarlo en una base de conocimiento es una tarea que merece la pena acometer. Partiendo de esta base, se abordarán a continuación los proyectos YAGO y DBpedia, los cuales también reconocen desde el momento de su creación el gran valor informativo que Wikipedia puede ofrecer, así como la utilidad de organizarlo dentro de una ontología o de una base de conocimiento.

Se comenzará por el proyecto YAGO (Suchanek et al., 2007). YAGO (acrónimo de *Yet Another Great Ontology*) es una ontología de entidades y relaciones que contiene más de un millón de entidades y más de 5 millones de relaciones. Se trata de un proyecto que pretende unificar información extraída de manera automática de Wikipedia y enriquecida con la base de datos léxica WordNet[12], de manera que esta última se vea enriquecida en tamaño y en contenido mediante la inclusión de información relativa a entidades, tales como personas, lugares, etc. Las relaciones en YAGO se explicitan mediante la estructura Is-A, así como otras relaciones no taxonómicas (por ejemplo, relación de transitividad). Se muestran a continuación algunos ejemplos que ilustran la manera en la que YAGO codifica la información (Suchanek et. al, 2007):

(5)

a)"*Einstein*" MEANS AlbertEinstein
b) AlbertEinstein TYPE physicist
c) physicist SUBCLASSOF scientist
d) subclassOf TYPE transitiveRelation

En (5a), se observa una redirección desde el texto que el usuario introduce en el momento de hacer la consulta, "Einstein", hacia la entidad almacenada en YAGO,

[11] http://www.wikipedia.org/

[12] http://wordnet.princeton.edu/

"AlbertEinstein"[13]. Puesto que las palabras también se consideran entidades[14], es por ello que YAGO permite esta redirección mediante la relación "MEANS". En los ejemplos (5b) y (5c), se reflejan dos de las múltiples propiedades de la entidad "AlbertEinstein" que YAGO almacena: en primer lugar, la entidad "AlbertEinstein" es una instanciación de la clase "physicist", una relación que se expresa mediante la relación denominada "TYPE", mientras que, a su vez, la clase "physicist" está encuadrada bajo la clase más numerosa "scientist", expresado mediante la relación denominada "SUBCLASSOF". Finalmente, en (5d) se observa que las relaciones también se consideran entidades, por lo que la relación del tipo "subclassOf" arriba mostrada, en minúscula en esta ocasión por ser considerada entidad, es una instanciación de la clase "transitiveRelation".

Si comparamos YAGO con el Onomasticón de FunGramKB, encontramos diferencias notorias: en primer lugar, YAGO es una base de datos relacional, no de semántica profunda, lo que desde el primer momento sitúa a FunGramKB en una posición frontal a ella. A pesar de que el hecho de que se trate de una base de conocimiento relacional presenta ventajas a la hora de representar el conocimiento (dada la sencillez de representación), se puede objetar que un sistema que define en relación a otros términos puede correr el riesgo de caer en la circularidad definitoria, así como en problemas derivados de la polisemia de las unidades léxicas, lo que en última instancia deriva en una robustez reducida a la hora de inferir información, disminuyendo la eficiencia global de la herramienta. Obviamente, esto ocurre debido al empleo de unidades léxicas en lugar de conceptos: un concepto es unívoco, pero una unidad léxica siempre corre el riesgo de ser polisémica. Los propios creadores de YAGO transmiten algunos casos de necesidad de desambiguación en procesos automáticos, casos que han de corregir de manera manual[15]. Otra diferencia relevante de YAGO frente a FunGramKB es que la base de datos WordNet, de la cual se extrae información, consiste en una red léxica, no en una base de conocimiento basada en el análisis de semántica profunda, lo que nos devuelve al

[13] Si bien los autores reconocen la posibilidad de que una misma palabra, en este caso "Einstein", se refiera a más de una entidad, como por ejemplo el músico Alfred Einstein, ejemplo citado por los autores (Suchanek et. al, 2007), de lo que se deriva que en algún punto del proceso sería preciso efectuar una desambiguación.

[14] "Einstein" se encuentra entrecomillado en YAGO, como muestra el ejemplo (5a), para expresar que se trata de una palabra.

[15] Un ejemplo de este fenómeno mostrado en Suchanek et al. (2007) menciona a la palabra "*capital*", entendida principalmente como "*capital city*" en Wikipedia y confundida frecuentemente con "*financial assets*", el cual resulta ser el sentido más frecuente en WordNet.

planteamiento ya mencionado de las bases relacionales. Consecuentemente, y a pesar de reconocer los esfuerzos realizados por la ontología YAGO, finalmente no se ha estimado que ésta resulte adecuada para que FunGramKB pueda utilizarla como recurso para la población semi-automática del Onomasticón.

Sin embargo, es pertinente mencionar otro proyecto de alta relevancia para el Onomasticón de FunGramKB, denominado DBpedia (Auer et al., 2007; Bizer et al., 2009). Si bien DBpedia tiene en común con YAGO el hecho de que obtiene datos de manera automática de Wikipedia, el resultado y la presentación de esta información son diferentes. El proyecto DBpedia[16] persigue extraer información estructurada procedente de Wikipedia para hacerla accesible en la web, después de haberla transformado en una rica base de conocimiento. La principal fuente de información para DBpedia, dentro de cada artículo de Wikipedia, son los *info-boxes*. Un *info-box*, situado generalmente a la derecha en la parte superior de un artículo, contiene resumida la información más relevante que se puede encontrar dentro del mismo, de manera estructurada en forma de una relación Is-A (p. ej., para una persona figurará: lugar de nacimiento, año de nacimiento; para un lugar, su ubicación, extensión, etc.). La Figura 5 que se incluye a continuación muestra uno de estos *info-boxes*:

Figura 5. Artículo sobre una ciudad en Wikipedia[17], con el info-box indicado a la derecha

[16] http://dbpedia.org/

[17] http://es.wikipedia.org/wiki/Albacete, fecha de consulta 2 de junio de 2010, 18:21 UTC.

Actualmente, DBpedia contiene más de 2,6 millones de entidades, que incluyen personas, lugares, obras musicales, películas y empresas (entre otros), tomadas de las versiones de Wikipedia en 35 lenguas naturales[18]. Resulta obvio que esta cantidad de entidades, ya clasificadas de manera sistemática, resulta de extremo interés para el Onomasticón de FunGramKB. Asimismo, otra de las características interesantes de DBpedia es que su base de conocimiento enriquece la información extraída de Wikipedia (particularmente, de los *info-boxes* de cada artículo) con datos procedentes de otras fuentes, lo que se consigue mediante 3,1 millones de enlaces a páginas externas y 4,9 millones de vínculos RDF a otras fuentes web de datos. Esto contribuye a que la información almacenada sea, además de más completa que la disponible en Wikipedia, constantemente actualizada a la vez que las fuentes hacen lo propio, ya que DBpedia mantiene actualizada no sólo aquella información que extrae de Wikipedia, sino también del resto de fuentes.

2.3. USO DE DBPEDIA PARA LA POBLACIÓN SEMI-AUTOMÁTICA DEL ONOMASTICÓN DE FUNGRAMKB

Quizás una de las aportaciones más valiosas que DBpedia puede ofrecerle al Onomasticón de FunGramKB sea la enumeración de propiedades de las entidades para que después dichas propiedades puedan trasladarse a FunGramKB. Si se observa el proceso de población del módulo conceptual Ontología de FunGramKB, éste se realiza de una forma manual a partir del conocimiento almacenado en diversos recursos lexicográficos (diccionarios, tesoros de la lengua, corpus lingüísticos, etc.) y la compleja posterior toma de decisiones que permite vincular en un mismo concepto diversas unidades léxicas. Ahora bien, y regresando al concepto de *proto*-estructuras mencionado con anterioridad (cf. sección 2.1), estas representaciones conceptuales responden a una realidad prototípica y, en el caso de la Ontología, atemporal. Se debe a esto que dichas representaciones conceptuales en forma de postulados de significado, sin perjuicio de contar en bastantes ocasiones con la etiqueta de "rebatibles", son en su gran mayoría propiedades estáticas, que aluden a un prototipo no concreto de la realidad. Sin embargo, algunas de las propiedades de las entidades del Onomasticón, puesto que dichas entidades son

[18] Inglés, alemán, francés, español, italiano, portugués, polaco, sueco, neerlandés, japonés, chino, ruso, finés, noruego, catalán, ucraniano, turco, checo, húngaro, rumano, volapük, esperanto, danés, eslovaco, indonesio, árabe, coreano, hebreo, lituano, vietnamita, esloveno, serbio, búlgaro, estonio y galés.

instanciaciones concretas de la realidad o *bio*-estructuras, tienen la posibilidad de variar en el tiempo sin afectar a la identificación de la entidad como tal. Las entidades del Onomasticón, pues, parecen poseer una mayor capacidad de dinamismo, al tratarse de entidades reales y, en numerosos casos, vivas o existentes, lo que podría apuntar a que la etiqueta de *bio* no es casual. Con objeto de ilustrar esto, véase el concepto +EAT_00, tomado de la Ontología de FunGramKB. En (6) se muestra el marco temático del concepto, así como su postulado de significado:

(6)

CONCEPT: +EAT_00

THEMATIC FRAME:
(x1: +HUMAN_00 ^ +ANIMAL_00)Agent
(x2: +SOLID_00 ^ +LIQUID_00)Theme
(x3)Location
(x4)Origin
(x5: +STOMACH_00)Goal

MEANING POSTULATE:
+(e1: +INGEST_00 (x1)Agent (x2)Theme (x3)Location (x4)Origin (x5)Goal (f1: +MOUTH_00)Means (f2)Instrument (f3: (e2: +CHEW_00 (x1)Theme (x2)Referent))Manner)

El ejemplo (6) muestra que hay ciertas condiciones que delimitan el concepto, como el mismo marco temático ilustra. Por ejemplo, según reflejan los esquemas COREL en (6), es preciso que el agente que ingiera el alimento sea un humano o un animal, y se requiere que lo haga llegar a su estómago (meta) a través de la boca (medio). Las propiedades (o valores) de animal o humano no pueden variar en el mundo que conocemos, por lo que los valores que forman el marco temático del concepto +EAT_00 son inamovibles. Sin embargo, si tomamos una entidad del Onomasticón, como sería una ciudad, su número de habitantes, extensión, bandera… podrían variar en el tiempo, sin significar la desaparición de la entidad ni un cambio imposible en el mundo que conocemos: las bio-macroestructuras, como son las *historias*, son capaces de tener una evolución, lo que las diferencia de las proto-microestructuras que componen la Ontología. Podría argumentarse, no obstante, que algunas de las propiedades de las estructuras del

Onomasticón son también invariables, como por ejemplo aquellos eventos acaecidos en el pasado, lo que haría necesario concluir que, si bien las entidades del Onomasticón poseen algunas propiedades invariables, también poseen numerosas propiedades susceptibles de evolución, mientras que las estructuras de la Ontología parecen estar *fosilizadas* en una mayor medida, a pesar de que este planteamiento siempre podría relativizarse a la luz de ejemplos concretos.

Retomando de nuevo la base DBpedia, otro de los principales valores que ésta ofrece a FunGramKB para la población del Onomasticón consiste en la estructuración de los datos que aquélla posee. DBpedia almacena los datos procedentes de los *info-boxes* de Wikipedia acerca de las entidades, completa esta información mediante el uso de otros recursos y, lo que resulta de mayor interés para FunGramKB, codifica de manera estructurada esta información. En el capítulo 3 del presente trabajo se describirá de manera detallada la metodología del proceso mediante el cual se crean, en primer lugar, reglas dentro de FunGramKB, que posteriormente permitan importar esta información estructurada de DBpedia para poblar el Onomasticón.

Resulta obvio el hecho de que, mediante esta importación a FunGramKB de datos procedentes de DBpedia, se está contribuyendo a la reusabilidad de recursos, una valiosa cualidad que ya se ha mencionado en diversas ocasiones. Esto es un hecho de evidente importancia, ya que permite la optimización de dichos recursos y la explotación de la eficiencia de la herramienta de origen, junto a la adaptación de la arquitectura de la plataforma de destino. Igualmente, se podría considerar que Wikipedia, una herramienta que posee tal ingente cantidad de información, recopilada y organizada, debería ser reutilizada para ser codificada de manera más semántica y menos jerárquica, una tarea que se puede considerar que FunGramKB, a través de los valores contenidos en estructuras de DBpedia, intenta acometer. No en vano, el propio proyecto DBpedia reconoce el potencial y reutiliza la información de Wikipedia junto a otras muchas fuentes. A modo ilustrativo, se puede observar la siguiente imagen que muestra todas las fuentes que actualmente DBpedia incorpora en sus estructuras, información que podrá ser a su vez integrada en FunGramKB cuando se haya completado la población semi-automática del Onomasticón con los datos procedentes de DBpedia:

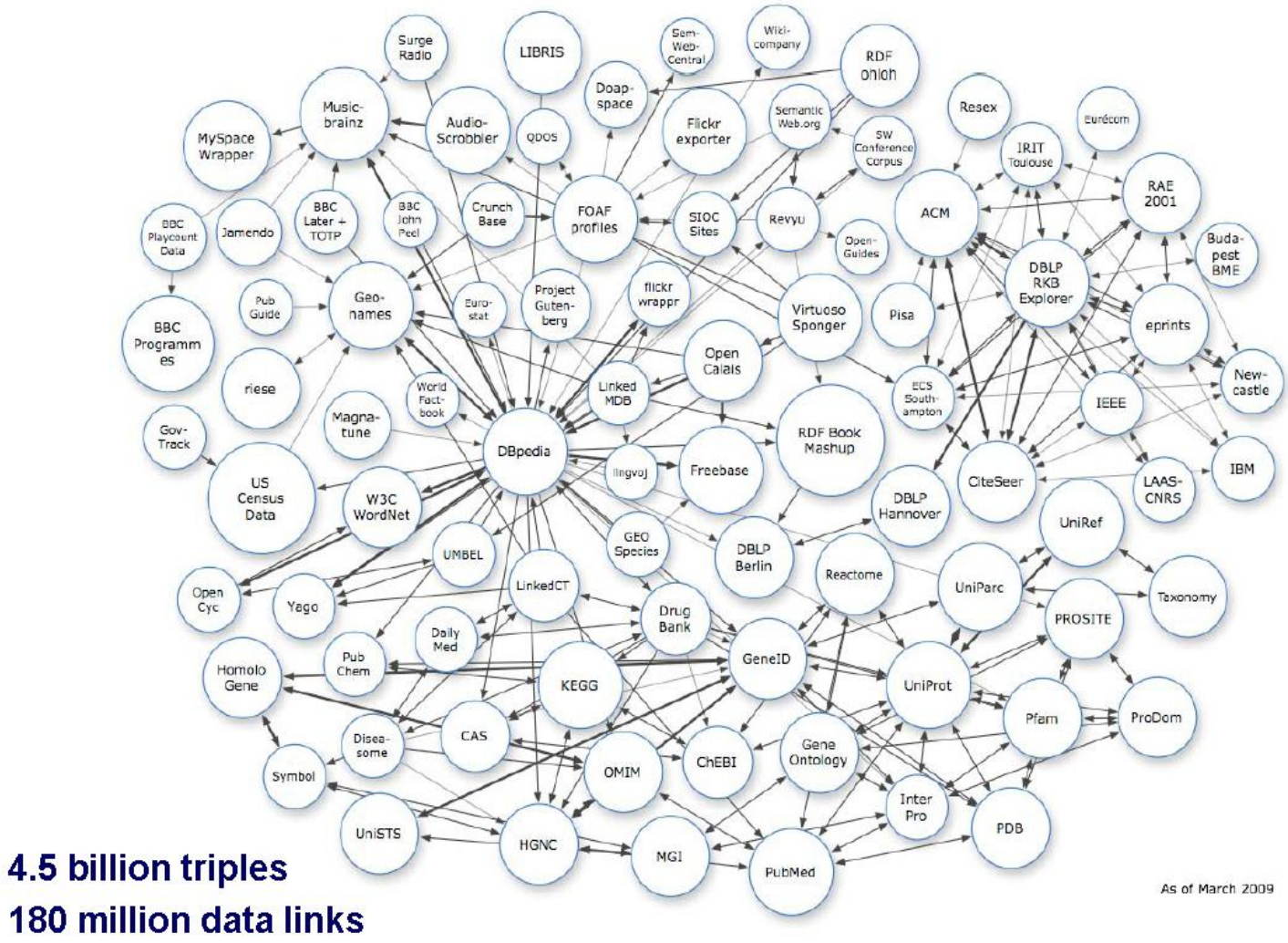

Figura 6. "Linked Open Data" sets en la web (Jentzsch, 2009)

Por tanto, a continuación se procederá a la descripción del proceso de creación de reglas que permitan la población semi-automática del Onomasticón y que, en última instancia, permitan la transmisión e integración en FunGramKB de la información y conocimiento de las fuentes arriba mostradas.

CAPÍTULO 3. METODOLOGÍA DE POBLACIÓN SEMI-AUTOMÁTICA DEL ONOMASTICÓN DE FUNGRAMKB

3.1. PRESENTACIÓN DE LA METODOLOGÍA

Como se ha mencionado con anterioridad, la reusabilidad de las herramientas ontológicas y de PLN es una característica clave a la hora de optimizar recursos y maximizar los resultados obtenidos mediante el empleo de dichas herramientas, a la vez que se trataría de potenciar un uso automático de las mismas que permita ambos efectos.

En relación a este planteamiento, en el presente capítulo se describirá la metodología mediante la cual el Onomasticón de FunGramKB, descrito en previas secciones, se puebla, con ayuda de otras herramientas ontológicas, de manera semi-automática mediante un proceso que se divide en tres pasos (Periñán Pascual y Arcas Túnez, 2010b):

1) Se han de crear, de manera manual, reglas basadas en plantillas que permitan plasmar en esquemas en lenguaje COREL el conocimiento almacenado en DBpedia.

2) Estas reglas se implementan en FunGramKB Suite, donde la proyección se produce de manera automática.

3) Puesto que DBpedia evoluciona a la vez que sus fuentes lo hacen (entre ellas, Wikipedia), el Onomasticón será actualizado también a través del servicio web.

A continuación, se comenzará describiendo en detalle el punto a) enumerado arriba, para lo cual es necesario, en primer lugar, mostrar la estructuración de las propiedades de las entidades presentada en DBpedia. Con posterioridad a que se conozca la estructuración que presentan estas propiedades, se describirá el proceso que comienza mediante la elección de cada uno de ellos y la subsiguiente codificación en lenguaje COREL del concepto que evoca, así como los aspectos más significativos a los que se ha de prestar atención durante este proceso.

3.2. LAS PROPIEDADES EN DBPEDIA: JERARQUÍA DE CLASES E INSTANCIACIÓN DE UNA CLASE: "*PLACE*"

La base de conocimiento DBpedia almacena información clave acerca de entidades tomada principalmente de Wikipedia, y almacena esta información de manera jerárquica. Esta jerarquía de clases se organiza en la llamada *Ontología de DBpedia*. En ella, todas las clases se aglutinan en torno a una sola matriz común, llamada "*Thing*", superordinado del cual parten todos las demás clases. La siguiente imagen muestra un fragmento (por motivos de espacio) de la vasta jerarquía de clases de la ontología de DBpedia:

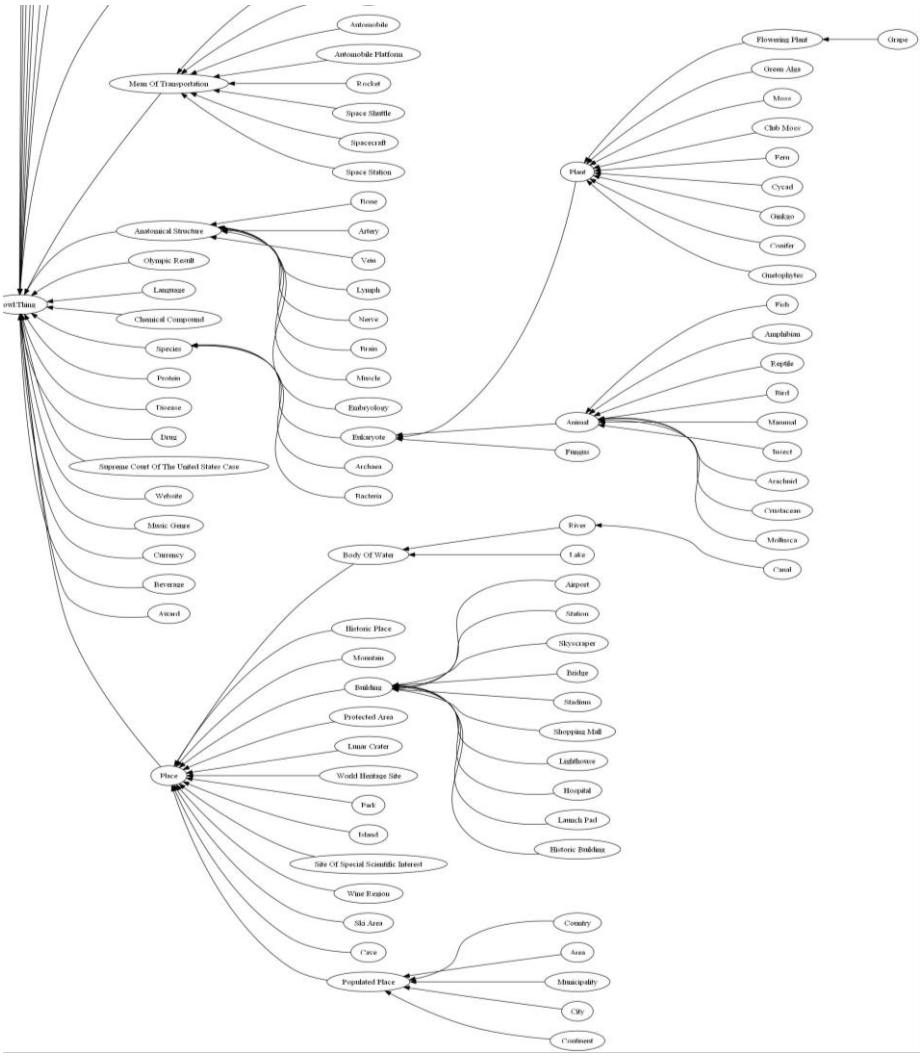

Figura 7. Fragmento de la jerarquía de clases de la ontología de DBpedia[19]

[19] Imagen original en http://www4.wiwiss.fu-berlin.de/dbpedia/dev/ontology.htm, fecha de consulta 8 de febrero de 2010

Como puede observarse, la jerarquía de clases de la Ontología de DBpedia es una extensísima red de clases de entidades, que abarca un total de más de 259 clases. Dentro de cada una de ellas se encuentra una serie de propiedades que definen a cada instanciación perteneciente a estas clases. Por ejemplo, si se toma la clase "*Place*", encontraremos diversas entidades dentro de ella, a la vez sub-organizadas (lo que DBpedia identifica como "*subClassOf*") en los diferentes tipos de lugar que se pueden identificar en la figura anterior (p.ej., "*populated place*" o "*body of water*"). Con carácter general, las entidades que se encuentren aglutinadas dentro de la misma clase tenderán a compartir una serie de propiedades que las definen, cuyo valor es obtenido de la información estructurada de los mencionados *info-boxes* de Wikipedia. De este modo, una propiedad de las entidades que componen la clase "*Place*" puede ser la altitud a la que un lugar se encuentra, la profundidad que posee o el área total que ocupa. Estas propiedades obtienen un valor según esté indicado en el *info-box* correspondiente de Wikipedia, por lo que cada instanciación dentro de la clase "*Place*" tendrá un valor asignado a cada una de sus propiedades, p.ej., los metros de altitud o de profundidad a la que se encuentra, o el número de metros cuadrados que ocupa su área total. La Ontología de DBpedia almacena actualmente 1200 propiedades asignadas a las diferentes clases[20]. Para ilustrar esta descripción, se puede observar en la Figura 8 a continuación una breve vista de una clase tomada de DBpedia[21]:

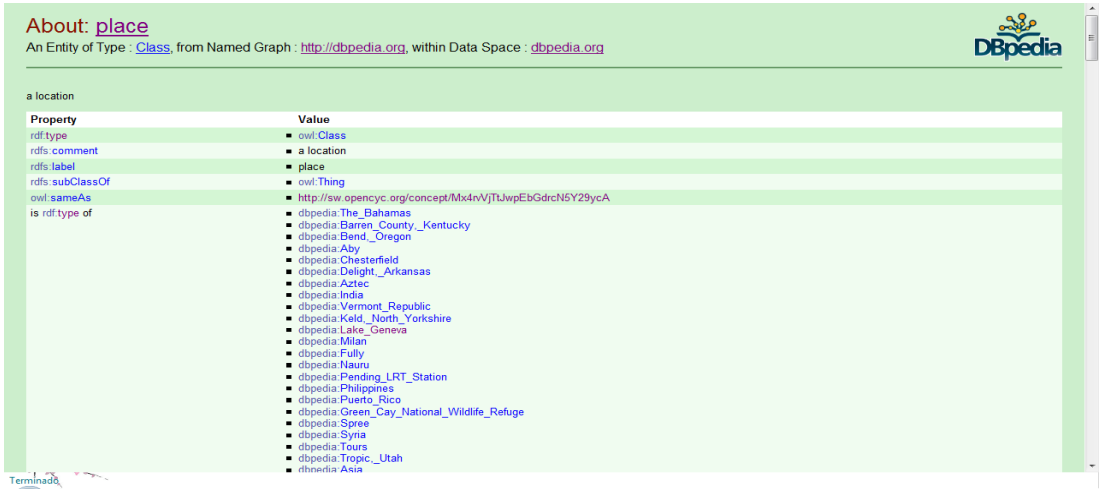

Figura 8. Vista parcial de la clase "*Place*" de DBpedia, donde se puede observar que "*Place*" parte del padre "*Thing*" (indicado por la etiqueta "*subClassOf*")

[20] http://wiki.dbpedia.org/Ontology, fecha de consulta 2 de junio de 2010

[21] http://dbpedia.org/ontology/Place, fecha de consulta 2 de junio de 2010

Dentro de cada clase de DBpedia existe la mencionada serie de propiedades que cada entidad perteneciente a dicha clase poseerá, y cuyo valor será tomado de la información que DBpedia extraiga de los *info-boxes* de Wikipedia. Si se accede a cada una de estas propiedades a través de DBpedia, se obtendrá más información acerca de las mismas, como puede ser una breve definición de ésta ("*comments*") o la unidad de medida (si procede) en la que se expresa su valor. Esta unidad de medida resultará relevante a la hora de crear manualmente la regla que definirá la propiedad dentro del Onomasticón de FunGramKB, algo que más adelante se pondrá de manifiesto cuando se describa el proceso de creación de reglas. A continuación se muestra, a modo de ejemplo, las imágenes que ilustran los datos que DBpedia proporciona acerca de tres de las propiedades, "*elevation*", "*areaTotal*" y "*nearestCity*", pertenecientes a la clase "*Place*":

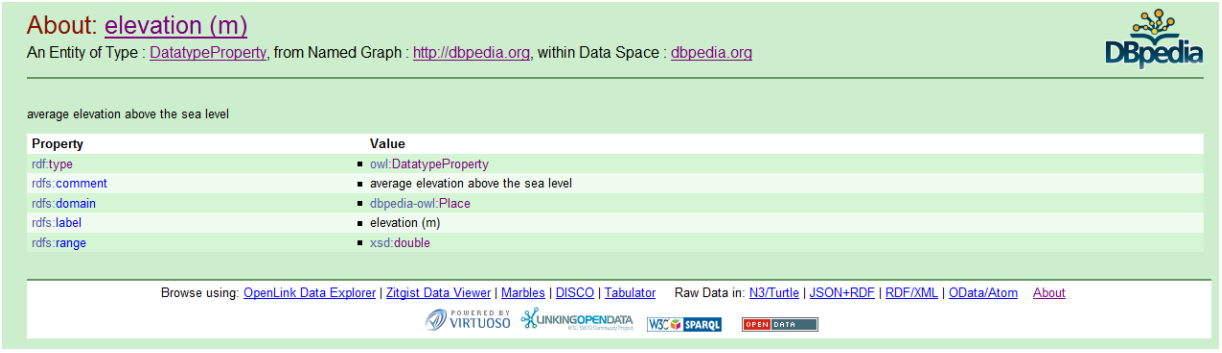

Figura 9. Propiedad "*elevation*", dentro de la clase "*Place*"[22]

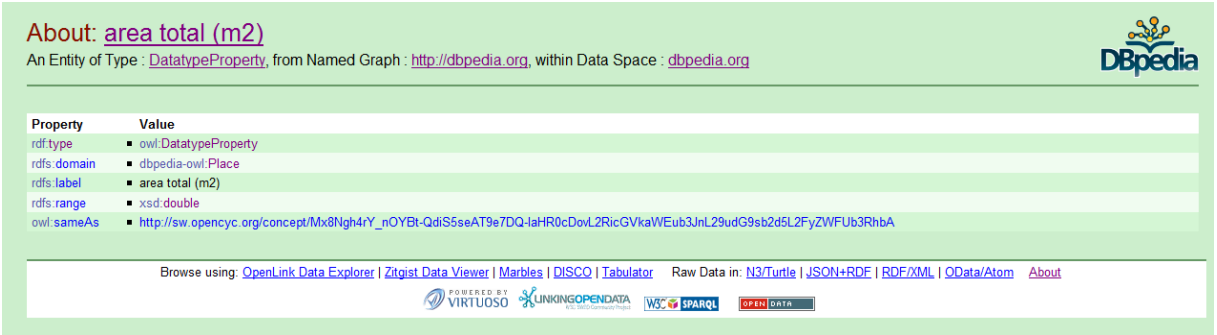

Figura 10. Propiedad "*areaTotal*", dentro de la clase "*Place*"[23]

[22] http://dbpedia.org/ontology/elevation, fecha de consulta 2 de junio de 2010

[23] http://dbpedia.org/ontology/areaTotal, fecha de consulta 2 de junio de 2010

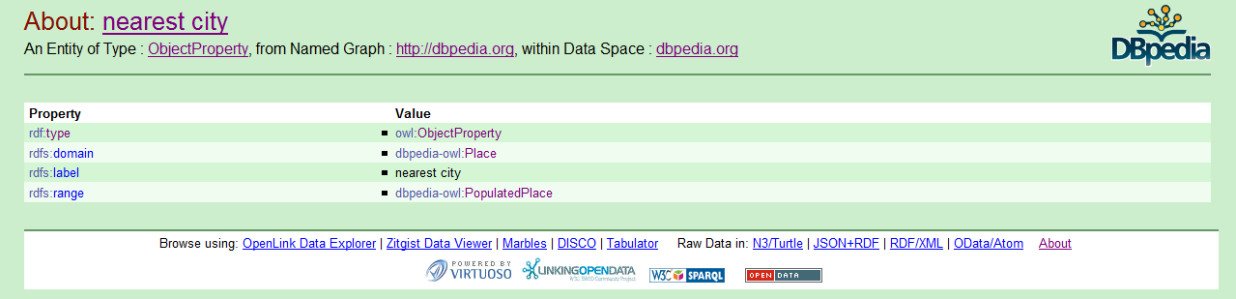

Figura 11. Propiedad "*nearestCity*", dentro de la clase "*Place*"[24]

Como se observa en las Figuras 9, 10 y 11, cada propiedad puede expresar su valor en una unidad de medida diferente, además de distinguirse, por otra parte, el hecho de que DBpedia puede proporcionar diferente cantidad de información acerca de cada una de las propiedades. En el caso de "*elevation*", se aprecia en la Figura 9 que se proporciona una breve descripción o aclaración de la propiedad en forma de "*comments*" ("*average elevation above the sea level*"), además de indicarse que esta propiedad se encuentra dentro de la clase "*Place*", y que su valor vendrá expresado en metros (m). Por otro lado, se observa en la Figura 10, correspondiente a la propiedad "*areaTotal*" que, si bien se mantiene la indicación de la clase "*Place*", desaparecen los comentarios aclaratorios, aunque la unidad de medida se sigue mostrando. En este caso, se indica que los valores mostrados por esta propiedad se expresarán en metros cuadrados (m^2). Finalmente, si se presta atención a la propiedad "*nearestCity*", ilustrada en la Figura 11, no aparece ninguna unidad de medida dentro de un sistema o clasificación internacional, puesto que esta propiedad carece de ella y, por tanto, no procede indicar nada al respecto (el valor vendrá expresado con un topónimo), mientras que algo que sí se indica es que se encuentra dentro de la clase "*Place*". Toda esta información añadida acerca de las propiedades de una clase es de gran importancia para la codificación en lenguaje COREL de cada propiedad mediante la creación de una regla, como más adelante se explicará.

Con objeto de contextualizar el entorno desde donde se obtendrá información para crear las reglas anteriormente mencionadas, han sido presentados diversos ejemplos tomados de la clase "*Place*" de la Ontología de DBpedia. Para mostrar de manera más concreta el proceso de creación manual de reglas, se continuará trabajando con esta clase a lo largo de las posteriores secciones del presente trabajo.

[24] http://dbpedia.org/ontology/nearestCity, fecha de consulta 2 de junio 2010

3.3. CREACIÓN MANUAL DE REGLAS: PROTOCOLO

Regresando al protocolo en tres pasos enumerado anteriormente para la población semi-automática del Onomasticón (Periñán Pascual y Arcas Túnez, 2010b), retomaremos la descripción detallada del punto a), que se corresponde con la creación, de manera manual, de reglas basadas en plantillas que permitan extraer mediante esquemas COREL el conocimiento almacenado en DBpedia.

Los pasos, a grandes rasgos, que se pueden enumerar dentro del protocolo para la creación manual de reglas, son los siguientes:

1) Pertinencia de la propiedad para FunGramKB
2) Elección de una propiedad dentro de una clase de DBpedia
3) Identificación del tipo de bioestructura (i.e. *retrato* o *historia*) a la que pertenecerá la predicación en la que se proyecte el atributo.
4) En caso de valor numérico, identificación de la unidad de medida
5) Construcción de las predicaciones de la regla en lenguaje COREL

Tras realizar el esbozo de la metodología llevada a cabo, mediante la enumeración de los pasos que componen el procedimiento, se procederá a continuación a describir en detalle los protocolos seguidos en cada uno de ellos para completar la creación de reglas en forma de esquemas COREL:

1. Pertinencia de la propiedad para FunGramKB

Con anterioridad a acometer la creación de una regla para cualquier propiedad, es necesario realizar un ejercicio previo que descarte, dentro del listado de propiedades que DBpedia proporciona para cada clase o subclase, cuáles de ellas resultarán pertinentes y relevantes para FunGramKB. Esta relevancia puede relacionarse con la reusabilidad y la pertinencia de la propiedad para el PLN, con su grado de generalidad o prototipicidad, con su capacidad expresiva o con motivos técnicos de la herramienta utilizada, entre otras razones que pudieran surgir ad hoc para cada caso concreto. Por ejemplo, a pesar de que FunGramKB fomenta la incorporación de diferentes conceptos culturales, se debería de sopesar la opción de que una propiedad vinculada fuertemente a una cultura en particular

pueda resultar poco rentable para FunGramKB, dada su reducida capacidad expresiva para el resto de culturas.

En el ejemplo que nos ocupa, puesto que la clase "*Place*" de DBpedia se considera análoga al concepto PLACE (+PLACE_00) de la Ontología de FunGramKB, se realizará una valoración de las propiedades que DBpedia proporciona para esta clase, eliminando aquellas que no resulten relevantes o pertinentes para el PLN y considerando las características peculiares del concepto PLACE de la Ontología de FunGramKB. Este proceso se llevará a cabo aduciendo las razones enumeradas en el párrafo anterior u otras que resulten necesarias para cada ejemplo concreto. En el Apéndice I del presente trabajo se incluyen las razones por las que aquellas propiedades para las que se prescinde de creación de regla han sido tales.

2. Elección de una propiedad dentro de una clase de DBpedia

Tras la clasificación previa descrita en el paso 1, se ha de proseguir con el siguiente eslabón de la cadena que permitirá crear la regla. A modo ilustrativo, tomaremos la propiedad "*elevation*" mostrada anteriormente, que hace referencia a una característica geográfica de las entidades que la clase "*Place*" engloba.

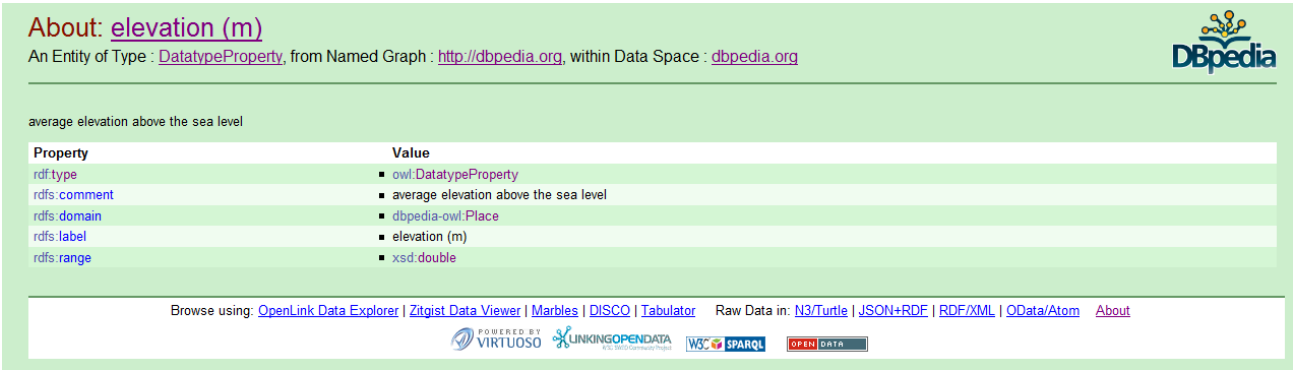

Figura 12. Propiedad "*elevation*" dentro de la clase "*Place*"[25]

Como se observa en la Figura 12, gracias a la información que DBpedia aporta acerca de la propiedad "*elevation*", es manifiesto que ésta, en primer lugar, se expresa en metros, por lo que esa será la medida que se codificará en COREL a la hora de elaborar la regla. En segundo lugar, para la propiedad "*elevation*", DBpedia aporta una breve reseña aclaratoria, llamada "*comment*", que, en este caso, se podría considerar una indicación

[25] http://dbpedia.org/ontology/elevation, fecha de consulta 2 de junio 2010

relativa a la desambiguación de la propiedad "*elevation*". Como resultado de esta desambiguación, se descubre, claramente, que la propiedad "*elevation*" en DBpedia se refiere al nivel promedio de altitud sobre el nivel del mar de un determinado lugar. Mediante esta aclaración se descarta que la propiedad llamada "*elevation*" en DBpedia se refiera a la altura, la cual es la distancia vertical entre dos puntos de la superficie terrestre, y no a partir del nivel del mar. Asimismo, también se explicita mediante este breve comentario que el valor asignado a la propiedad "*elevation*" de una entidad será un valor promedio ("*average*"), puesto que debido a las diferencias geológicas y geográficas del relieve de un determinado lugar, es posible que en superficies de extensión amplia tengan lugar variaciones en altitud entre dos puntos de la zona. En caso de que lo que se deseara reflejar fuese la altitud máxima o mínima de un lugar, DBpedia también cuenta con otras propiedades que reflejan estas características de manera separada, i.e., las propiedades "*maximumElevation*" y "*minimumElevation*". Una vez que se ha escogido la propiedad a analizar, y se ha hecho una observación pormenorizada de la información que DBpedia aporta acerca del mismo, se procedería al siguiente paso dentro del protocolo de creación de reglas.

3. Identificación del tipo de bioestructura (i.e. *retrato* o *historia*) a la que pertenecerá la predicación en la que se proyecte la propiedad

Como se ha mencionado con anterioridad en secciones previas del presente trabajo (cf. secciones 2.1 y 2.3), las entidades o bio-estructuras del Onomasticón de FunGramKB pueden ser clasificadas como retratos (*snapshots*) o historias (*stories*), en función de si se trata de propiedades estáticas en un momento particular (retratos), o bien de propiedades que han de encajarse dentro de un esquema temporal determinado (historias). Por tanto, existirán propiedades de DBpedia que pueden contribuir a la construcción de retratos o de historias. En el ejemplo que nos ocupa, la propiedad "*elevation*" se consideraría un retrato, ya que el valor que esta propiedad adopte en relación a una entidad determinada será el reflejo de dicha propiedad en un momento aislado del tiempo, además de estar sujeto a cambio[26]. Un ejemplo de propiedades que adoptan el perfil opuesto –historias– serían aquellas tales como "*birthDate*" o "*deathDate*" que pueden encontrarse dentro de la clase

[26] P.ej., si se considera el hipotético caso de que por causas naturales, como un terremoto, un corrimiento de tierra o la subida del nivel del mar la altitud de un lugar sufra modificaciones.

"*Person*"[27] de DBpedia, ya que se trata de propiedades que han de ser encajadas dentro de un marco temporal determinado de manera cronológica.

4. En caso de valor numérico, identificación de la unidad de medida

A continuación, y como también se ha indicado en el paso 2, antes de proceder a elaborar la regla en lenguaje COREL, es importante identificar los parámetros o medidas que se necesitarán a la hora de hacer la codificación. En el ejemplo que nos ocupa, gracias a la información que DBpedia aporta sobre la propiedad "*elevation*" (como se ha mostrado en las Figuras 9 y 12), se puede identificar el metro como unidad de medida de los valores que "*elevation*" adquiere para cada entidad. Puesto que en la regla que se cree en lenguaje COREL se habrá de indicar esta unidad, procedería, llegados a este punto, hacer una búsqueda en la Ontología de FunGramKB del concepto que se corresponda con la unidad que se necesita. En el ejemplo actual, se buscaría en la Ontología de FunGramKB el concepto que se correspondiese con la unidad de medida "metro", ya que los conceptos que se usen en la regla en lenguaje COREL han de estar previamente incluidos en la Ontología de FunGramKB. Se trata en este caso, nuevamente, de otra manifestación de la interrelación necesaria entre los diferentes módulos de FunGramKB, que contribuye a su consistencia e interoperabilidad. Con objeto de conocer el hecho de si la unidad de medida "metro" está ya incluida como concepto dentro de la base de conocimiento FunGramKB, se ha de recurrir a la herramienta de búsqueda ("SEARCH") dentro de la Ontología de FunGramKB. A tal efecto, y por razones prácticas de optimización de recursos, no es necesario acceder a la Ontología de FunGramKB de manera independiente, ya que se puede encontrar un acceso directo a la herramienta de búsqueda dentro de la aplicación "FunGramKB Editor: Onomasticon", como se ilustra en la Figura 13:

[27] http://dbpedia.org/ontology/Person

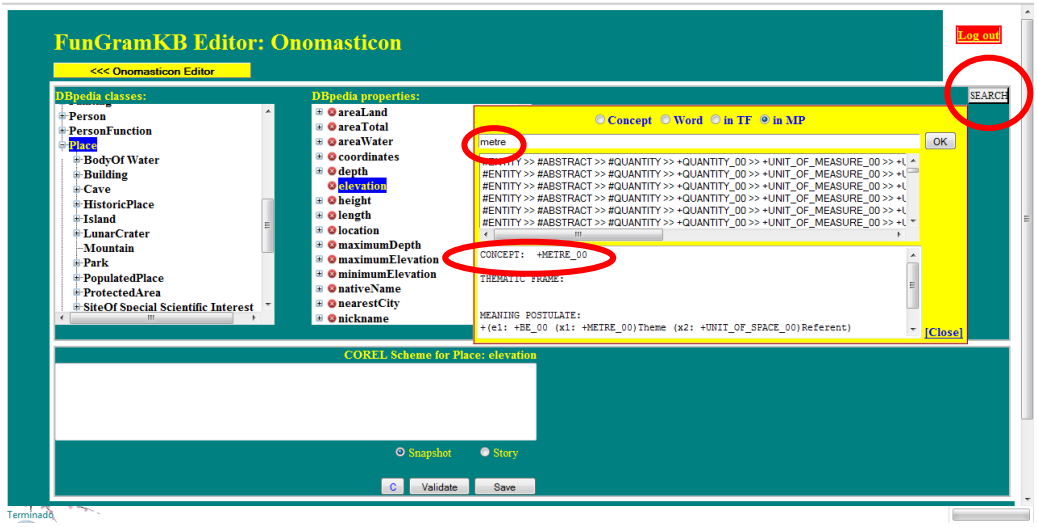

Figura 13. Vista de "FunGramKB Editor: Onomasticon", tras haber realizado la búsqueda de la unidad léxica "metre", obteniendo como resultado el concepto básico +METRE_00.

En caso de que se hubiera realizado la búsqueda del concepto utilizando la unidad léxica "metro", en español, se habría llegado al mismo resultado, ya que los conceptos son independientes de la lengua natural en que estén expresadas las lexicalizaciones que se hagan de los mismos. Además, esto es posible gracias al mencionado carácter multilingüe de FunGramKB, que permite asignar diferentes lexicalizaciones en varias lenguas naturales a un mismo concepto expresado en lenguaje COREL. La Figura 14 ilustra este fenómeno:

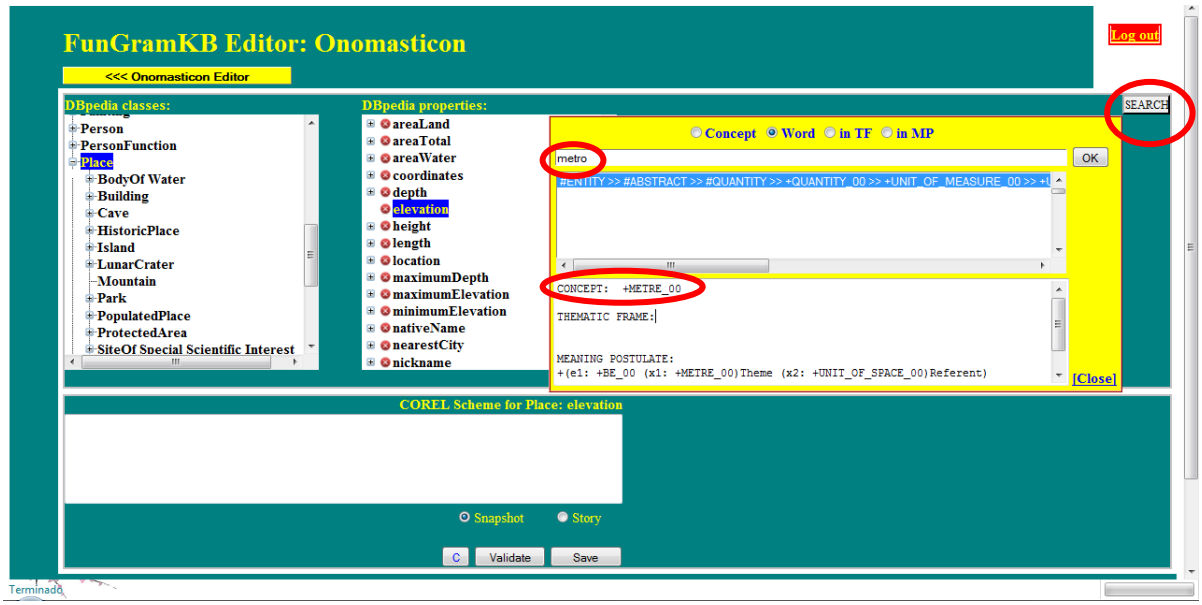

Figura 14. Vista de "FunGramKB Editor: Onomasticon", tras haber realizado la búsqueda de la unidad léxica "metro", obteniendo el mismo concepto básico +METRE_00.

En el potencial caso de que la unidad de medida que se necesita emplear en la creación de la regla no apareciera en la búsqueda realizada, sería necesario crearla en la Ontología de FunGramKB antes de utilizarla en el Onomasticón. Esta tarea tiene como objetivo mantener la correlación entre módulos, ya que en las reglas introducidas en el Onomasticón no se han de emplear conceptos inexistentes en la Ontología de FunGramKB. Para llevar a cabo la tarea de introducción del nuevo concepto, se requeriría acceder a la aplicación "FunGramKB Editor" de la Ontología de FunGramKB, y crear el concepto terminal[28] que indicaría la unidad de medida dentro de este módulo, siguiendo el procedimiento pertinente a tal fin, no sin antes realizar todas las comprobaciones posibles de que dicho concepto efectivamente no existe en FunGramKB. En caso de que la unidad de medida no pertenezca a un sistema internacional establecido (como, p.ej., el sistema métrico decimal) y en su lugar se trate de otro concepto (una ciudad, un año, etc.), también habría de comprobarse que dicha unidad existe como concepto en la Ontología de FunGramKB, antes de avanzar al siguiente paso. En caso de que esta fase ya esté completada, se procederá al cuarto y más complejo paso: la elaboración de la regla en lenguaje COREL.

5. Construcción de las predicaciones de la regla en lenguaje COREL

Una vez que se han realizado las comprobaciones anteriores, acerca de la existencia en la Ontología de FunGramKB de los conceptos necesarios para expresar el valor de la propiedad, se procederá a elaborar la regla, basada en plantillas, que posteriormente será utilizada para importar información de DBpedia.

Con el fin de ilustrar de manera más clara la elaboración de la regla, se continuará con la propiedad "*elevation*", perteneciente a la clase "*Place*" escogida con anterioridad.

Una regla es una plantilla que expresa en lenguaje COREL, mediante una secuencia de predicaciones, al estilo de un postulado de significado, la relación existente entre una entidad y la clase al que pertenece, una propiedad de esta entidad y los posibles valores que entidad y propiedad pueden adoptar. A la entidad poseedora de la propiedad se le denominará <C> ("concept"), y al valor que ostenta la propiedad se le denominará <VAL> ("value"). Dentro del esquema COREL que compone la regla, se ha de encontrar

[28] Puesto que el inventario de conceptos básicos es cerrado, es por ello que el nuevo concepto a crear debe ser un concepto terminal.

una primera predicación, indicada como (e1), que refleje la clase a la cual pertenece la propiedad cuya regla se esté elaborando. En el caso de la propiedad "*elevation*", se representaría de la siguiente forma:

(7)

+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent)

El ejemplo (7) muestra la primera predicación del esquema COREL de "*elevation*". El equivalente en lenguaje natural de la predicación (e1) podría expresarse como "existe una entidad x1:<C> y esta entidad es siempre un lugar". El símbolo (+) que la predicación porta al frente indica que se trata de una predicación estricta, ya que una regla para propiedades que se encuentren dentro de la clase "*Place*" siempre requerirá que la entidad encajada en la posición <C> sea un lugar. La entidad concreta de la que se trate aparecerá de manera automática en la posición <C> una vez que se realice la proyección desde DBpedia, una tarea que tendrá lugar tras la introducción de la regla en la herramienta DBpedia Mapper del Onomasticón, la cual permite la proyección de las clases y propiedades de DBpedia en el Onomasticón de FunGramKB.

A continuación, y una vez que se ha indicado en la predicación inicial (e1) la clase a la que pertenece la propiedad y a la que ha de pertenecer la entidad, se procederá a codificar la propiedad en sí. En el ejemplo que nos ocupa, se codificaría de la siguiente forma:

(8)

*(e2:+BE_01 (x1)Theme (x3: $ALTITUDE_00)Attribute (f1: <VAL>
+METRE_00)Quantity)

El ejemplo (8) muestra la segunda predicación que compone la regla de "*elevation*". En primer lugar, se observa que el símbolo (*) al frente de la predicación indica que ésta es rebatible, ya que el valor de esta predicación es susceptible de variar (por causas naturales, p.ej.). En segundo lugar, se puede identificar que el concepto escogido para reflejar "*elevation*" se corresponde con el concepto terminal, indicado por el símbolo ($), $ALTITUDE_00. Gracias a los comentarios incluidos por DBpedia en la propiedad "*elevation*", como se indicaba en el paso 2, es posible escoger el concepto que mejor se corresponde con las características de la propiedad. La manera de buscar el

concepto más apropiado coincide con el método de búsqueda de la unidad de medida que se ha descrito en el paso 4 e ilustrado en las Figuras 13 y 14. En este caso, al introducir "*elevation*" como unidad léxica ("Word") en la casilla de búsqueda, ésta no devuelve resultados, como se ilustra a continuación:

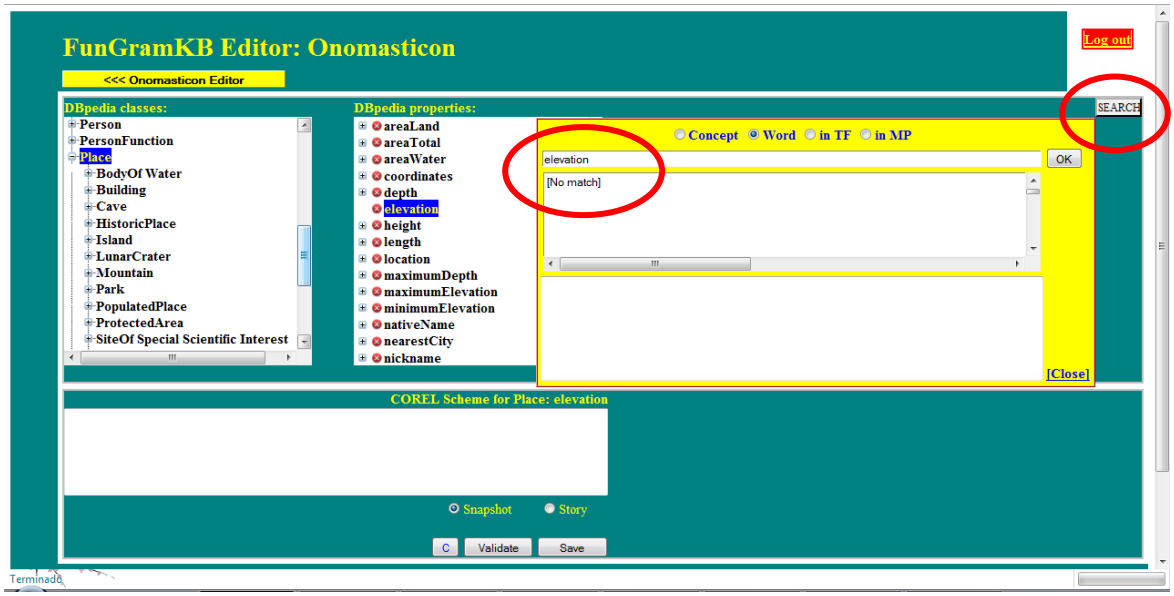

Figura 15. Vista de "FunGramKB Editor: Onomasticon", tras haber realizado la búsqueda de la unidad léxica "*elevation*", sin obtener resultado.

A pesar de no haber obtenido resultado en la búsqueda de la unidad léxica "*elevation*", sería poco prudente concluir sin mayor comprobación que el concepto que designa no existe en la Ontología de FunGramKB. Se ha de recordar que la Ontología de FunGramKB está estructurada en torno a conceptos, y no en torno a unidades léxicas o palabras, de lo que se desprende que es posible que el concepto denotado por la propiedad "*elevation*", descrita en los comentarios de DBpedia como "*average elevation above the sea level*", (como muestran las Figuras 9 y 12), exista en la Ontología de FunGramKB lexicalizado de manera diferente. Con el propósito de averiguar este hecho, una opción es acudir a recursos lexicográficos que aporten una descripción extendida de la unidad léxica "*elevation*", como por ejemplo la definición que se obtiene tras la consulta del diccionario on-line *Longman Contemporary Dictionary*[29], el cual define "*elevation*" como "*a height above the level of the sea*"[30]. Al hallar la unidad léxica "*height*" dentro de la definición, se podría acometer una búsqueda de esta unidad en la Ontología de FunGramKB, gracias a lo

[29] http://www.ldoceonline.com/

[30] http://www.ldoceonline.com/dictionary/elevation, fecha de consulta 4 de junio de 2010.

cual se posibilita el encuentro con el concepto que mejor refleja la propiedad que se desea plasmar en la regla: el concepto $ALTITUDE_00, como muestra la Figura 16:

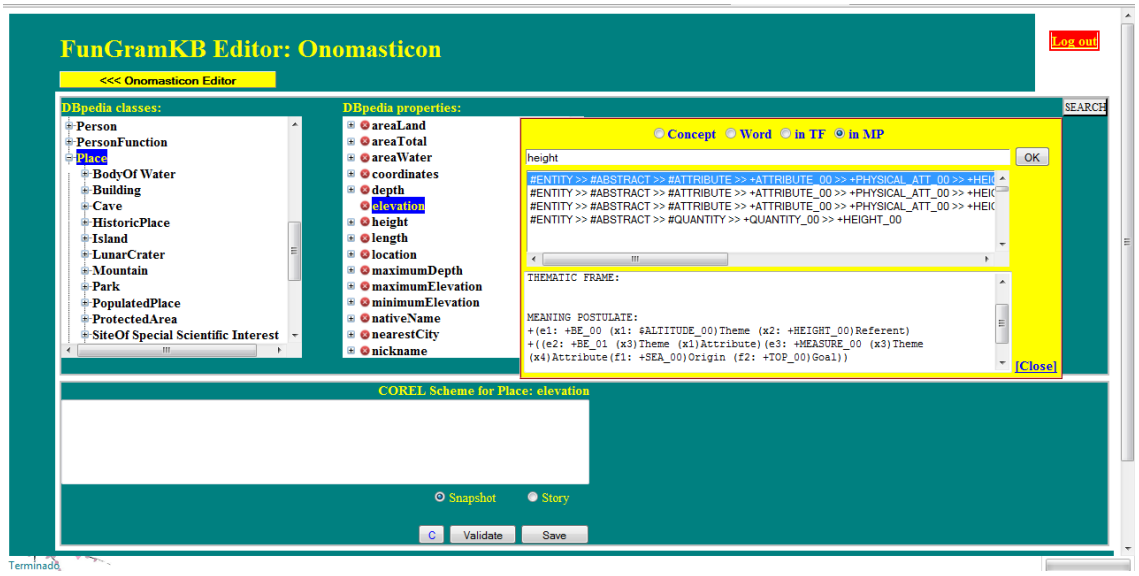

Figura 16.

Una vez que se ha hallado el concepto más cercano a la propiedad cuya regla está siendo creada, se procederá a su inserción de la manera más adecuada en la predicación. No obstante, es posible que en otras ocasiones no sea posible hallar un concepto equivalente a la propiedad de DBpedia de manera exacta en la Ontología de FunGramKB. En estos casos, se tendrá que describir este concepto mediante el empleo de otros que permitan la construcción conceptual de la propiedad que se esté buscando. Este hecho no debería ser considerado una desventaja, sino todo lo contrario, en aras a tener una representación de significado más rica dentro del Onomasticón.

Retornando a la predicación (e2) que aparece en el ejemplo (8), se advierte también la presencia del concepto +METRE_00, que indica la unidad de medida en la que se expresa el valor <VAL>. Es en este momento cuando se observa la necesidad de indicar la unidad de medida, como ya se había anticipado en previos apartados de la presente sección (pasos 2 y 4). Si se realiza una equivalencia en lenguaje natural de lo que la predicación (e2) refleja en lenguaje COREL, podría expresarse como: "la entidad (x1)[31] posee una propiedad que es la altitud y que obtiene el valor de <VAL> metros". Puesto que la predicación es rebatible, nótese que no se indica ningún adverbio absoluto como "siempre"

[31] La entidad (x1) aparece de nuevo, y por co-indización se refiere a la misma entidad (x1:<C>) de la anterior predicación, (e1), que ya se ha indicado que es siempre un lugar.

o "nunca" en la equivalencia en lenguaje natural del esquema COREL. Mediante la combinación de ambas predicaciones, (e1) y (e2), se obtiene la regla final, que para la propiedad "*elevation*" de la clase "*Place*" sería la siguiente:

(9)

+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent)
*(e2:+BE_01 (x1)Theme (x3:$ALTITUDE_00)Attribute
(f1:<VAL>+METRE_00)Quantity)

Tras la elaboración de la regla, procedería introducirla en el correspondiente recuadro de "COREL scheme for Place: elevation" dentro del "FunGramKB Editor: Onomasticon" y realizar las pertinentes operaciones de validación sintáctica de la regla, una posibilidad que ofrece la aplicación mediante la selección de la opción "*validate*" presente bajo el recuadro, como se puede observar en la Figura 17:

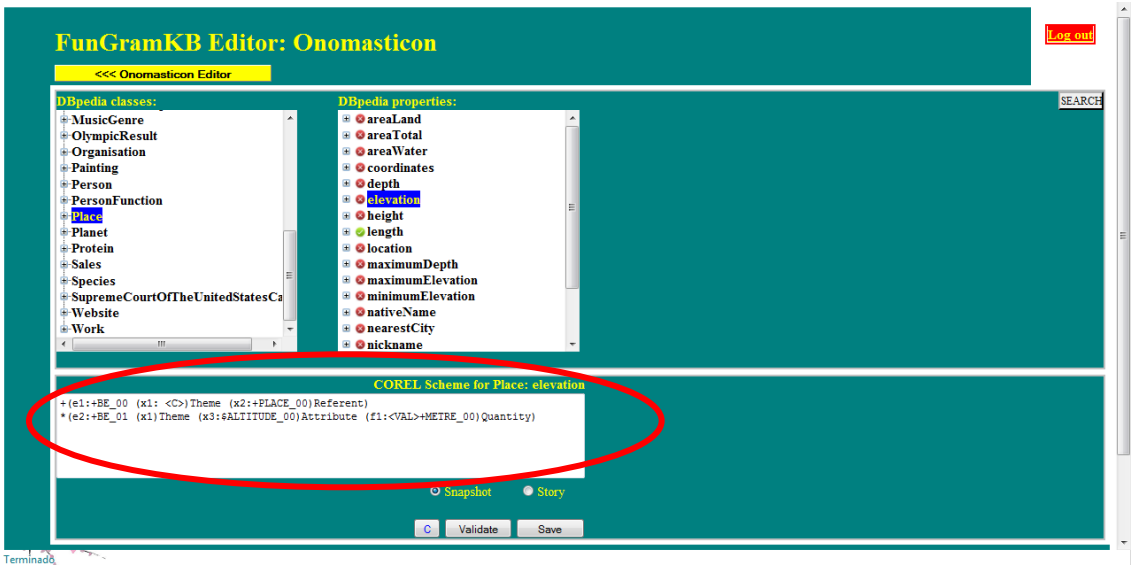

Figura 17. Introducción de la regla para la propiedad "*elevation*" en "FunGramKB Editor: Onomasticon"

Después de presionar la opción "*save*", ubicada a la derecha de la opción de validación, como se muestra en la mencionada figura, habrá quedado almacenada finalmente la regla en el Onomasticón de FunGramKB.

Con posterioridad a la introducción de cada regla en el Onomasticón de FunGramKB, tiene lugar un proceso computacional que permite la proyección de las reglas en formato COREL con la información tomada de DBpedia, hacia predicaciones

que configuren los retratos o historias correspondientes. Mediante la regla creada en la presente sección para la propiedad "*elevation*", podrían obtenerse, tras la mencionada proyección, los siguientes ejemplos:

(10)

a)
+(e1:+BE_00 (x1: %TOLEDO_SPAIN_00)Theme (x2:+PLACE_00)Referent)
*(e2:+BE_01 (x1)Theme (x3:$ALTITUDE_00)Attribute
(f1:529+METRE_00)Quantity)

Toledo (España) es un lugar ubicado a 529 m de altitud sobre el nivel del mar

b)
+(e1:+BE_00 (x1: %SAN_FRANCISCO_00)Theme (x2:+PLACE_00)Referent)
*(e2:+BE_01 (x1)Theme (x3:$ALTITUDE_00)Attribute (f1:15.8496+METRE_00)
Quantity)
San Francisco es un lugar ubicado a 15,8496 m de altitud sobre el nivel del mar

c)
+(e1:+BE_00 (x1: %CASTELLON_DE_LA_PLANA_00)Theme
(x2:+PLACE_00)Referent)
*(e2:+BE_01 (x1)Theme (x3:$ALTITUDE_00)Attribute (f1: 30 +METRE_00)
Quantity)

Castellón de la Plana es un lugar ubicado a 30 m de altitud sobre el nivel del mar

Como se desprende de los ejemplos anteriores en (10), la proyección posibilitada por el proceso computacional materializa la sustitución del concepto <C> por un concepto que represente una entidad concreta, así como la del valor <VAL> por otro concepto o por un operador de cuantificación, según corresponda, produciendo de este modo una identificación entre entidades, sus propiedades y el valor real que éstas poseen, según los datos contenidos en los *info-boxes* de Wikipedia, proyectados a través de DBpedia en el Onomasticón de FunGramKB.

3.4. RESULTADOS Y DISCUSIÓN

En el Apéndice I ubicado al final de este trabajo se muestra el listado de las reglas pertenecientes a la clase "*Place*" de DBpedia dentro del Onomasticón de FunGramKB, lo que constituye los resultados de la ejecución del proceso anteriormente descrito para la creación manual de reglas.

Si se observa con detalle cada una de las reglas, en primer lugar, una característica notable es que muchas de ellas presentan una estructura análoga, por lo que pueden agruparse de manera similar. De este modo, se observa que para propiedades geográficas simples que codifican una dimensión (elevación, altura, altitud, anchura, área total, profundidad, longitud, ubicación), las predicaciones que aparecen en segundo lugar (e2)[32] adoptan una estructura de papeles temáticos del tipo "Theme + Attribute +Quantity", además de que todas ellas parecen reflejar propiedades promedio y no valores extremos.

Por otro lado, en aquellas propiedades en las que interviene una especificación de las mismas de tipo superlativo, mediante un premodificador como "*maximum*" o "*minimum*", o un superlativo como "*nearest*", continúa mostrándose una estructura de dos predicaciones, cuya segunda predicación mantiene el esquema "Theme + Attribute + Quantity", si bien aparece un segundo satélite en la segunda predicación, que añade el papel temático de "Scene" o "Position" (con sus respectivos papeles dentro del satélite), con motivo de precisar que esa dimensión tiene lugar en un momento y punto geográfico determinados, en contraposición a lo que ocurre con las propiedades denominadas simples en el párrafo anterior, que abogan por reflejar un valor promedio de la propiedad. Podría argumentarse que una separación entre reglas del tipo "*maximum/minimum*" y aquellas que contienen superlativos como "*nearest*" pudiera ser conveniente; no obstante, sería necesario un estudio más extenso de este tipo de estructuras en el resto de clases para determinar la relevancia y necesidad de dicha separación.

En tercer lugar, se observa otro tipo de propiedades más complejas, que demandan la integración de otros conceptos en el esquema, como muestran las reglas correspondientes a las propiedades área de agua, área de tierra y porcentaje de área cubierto por agua. Los esquemas de estas propiedades precisan la introducción de otros

[32] Es preciso recordar que las predicaciones que aparecen en primer lugar en las reglas, adoptando la posición (e1), cumplen la función de contextualizar la regla en su clase correspondiente, así como conectar una determinada entidad <C> con la clase a la que pertenece.

conceptos que reflejen las circunstancias de ubicación espacial (reflejado en los satélites de posición, "Position", que aparecen en las predicaciones), por lo que el número de predicaciones tiende a aumentar como norma general a tres en lugar de dos predicaciones, ya que la predicación ubicada entre aquella que indica la relación con la clase[33] (la primera) y la que contiene el valor de la propiedad, ha de contextualizar espacialmente de manera más precisa la propiedad codificada. Este planteamiento deriva en la aserción, basada en los resultados obtenidos, de que, mientras que la primera predicación siempre indica la clase a la que pertenece la regla y la entidad que posteriormente se proyecte en ella, la última predicación es siempre la que contiene el valor que la propiedad puede adoptar, sin perjuicio de que sea preciso intercalar otras predicaciones entre la primera y la última con objeto de completar la definición de la propiedad que la regla denota.

Finalmente, se pueden contemplar ciertas propiedades para las que no es preciso elaborar una regla por diversas razones, como se ha anticipado en el paso 1 de la sección 3.3. De este modo, para las propiedades referentes a "*otherName*", "*nickname*" y "*nativeName*" no se estima necesaria la creación de una regla por motivos técnicos de la herramienta y con objeto de minimizar la redundancia, ya que aluden a formas alternativas de nombrar la entidad, lo que puede ser resuelto mediante un proceso computacional que permita una redirección a la entidad principal, y que muestre denominaciones alternativas de la entidad en una sección que se designará como "*alternative name box*". Un ejemplo de esto podría observarse en una entidad como la Mona Lisa (%MONA_LISA_00), entidad a la cual se redireccionarían consultas que contuvieran sobrenombres como "La Gioconda" o "La Joconde". Otra propiedad de la que se ha decidido prescindir a la hora de elaborar el inventario de reglas es "*type*" al resultar redundante, puesto que esta propiedad ya queda reflejada en la jerarquía en la que se encuadre la entidad (clase y subclase a la que pertenezca). En última instancia, se observa una propiedad como "*coordinates*", para la cual no se ha elaborado regla, ya que no se estima relevante para el PLN la codificación de propiedades como coordenadas de longitud y latitud. No obstante, no se ha de descartar que, en futuras ediciones y trabajos, esta propiedad u otras que pudieran ser inicialmente descartadas de manera argumentada, adquieran un estatus de relevancia tal que las transforme en dignas de ser incluidas como reglas en el Onomasticón de FunGramKB.

[33] Ídem nota 32.

CAPÍTULO 4. CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO

El presente trabajo ha versado, tras una introducción sobre la materia de ontologías y bases de conocimiento, acerca de las características del Onomasticón de FunGramKB, el proceso y metodología necesarios para llevar a cabo la población semiautomática de este módulo, así como de la descripción de otras herramientas ontológicas (YAGO, DBpedia) y su reutilización con los fines de población mencionados. Si bien la herramienta DBpedia en particular resulta de formidable aprovechamiento para la población del Onomasticón de FunGramKB, existen algunos detalles que precisarían de perfeccionamiento para optimizar la explotación de dicho recurso. DBpedia adolece particularmente de que la documentación e información disponible acerca de cada propiedad es inconsistente e incompleta, una característica que ya se ha reflejado en el anterior capítulo mediante la descripción de la información que DBpedia ofrece sobre cada una de ellas. El objetivo de una mejora de estas características debería suponer que la información proporcionada acerca de cada propiedad sea homogénea, de modo que para cada una se cuente con una descripción de la misma, unidad de medida y clase a la que pertenece, entre otros aspectos relevantes que completen la descripción de la propiedad, con el fin de que estos datos pudieran ser reutilizados y trasladados a otras herramientas (en el caso que nos ocupa, a las reglas del Onomasticón de FunGramKB) de una manera más óptima.

Tras la aplicación del proceso descrito en el capítulo 3 del actual trabajo, se han obtenido los resultados que pueden observarse en el Apéndice I, esto es, las reglas en lenguaje COREL que codifican las propiedades que se encuentran dentro de la clase "*Place*" de la versión 3.4 de DBpedia. Terminado el análisis de dichas reglas en la sección 3.4 del capítulo 3, se observa que es necesario y pertinente realizar posteriores investigaciones acerca de las tipologías de reglas que parecen surgir entre las reglas creadas. Como ya se ha indicado en la mencionada sección, sería preciso realizar un análisis exhaustivo de las reglas que pueblan el resto de clases para observar si, efectivamente, la tipología de reglas observada en la clase "*Place*" es extrapolable al resto de ellas, así como para descubrir ulteriores tipologías en clases de naturaleza tan diversa como "*Person*" o "*Work*", entre otras muchas, algo que en esta fase embrionaria de la investigación se podría vislumbrar. En última instancia, se debería analizar el porqué de estas tipologías de reglas, con objeto de indagar en la hipótesis de que una estructura

compartida pueda estar ligada a una motivación semántica compartida, hipótesis que se acometerá en la tesis doctoral que suceda al presente trabajo. Asimismo, y paralelamente a la elaboración de dicha tesis, es importante hacer notar que se están llevando a cabo en la actualidad investigaciones acerca de la implementación computacional de las reglas en el Onomasticón de FunGramKB, la cual permite la proyección de los datos procedentes de DBpedia en las plantillas de reglas creadas, al igual que se están perfilando futuros y perfeccionados desarrollos del DBpedia Mapper que soporta dichas implementaciones computacionales y proyecciones dentro del Onomasticón de FunGramKB.

REFERENCIAS BIBLIOGRÁFICAS

Allen, J.F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM 26 (11)*, 832-843.

Allen, J.F. y Ferguson, G. (1994). Actions and events in temporal logic. *Journal of Logic and Computation 4 (5)*, 531- 579.

Auer, S., Bizer, C., Lehmann, J., Kobilarov, G., Cyganiak, R., e Ives, Z. (2007). DBpedia: A Nucleus for a Web of Open Data. En Aberer et al. (Eds.), *The Semantic Web, 6th International Semantic Web Conference, 2nd Asian Semantic Web Conference, ISWC 2007 + ASWC 2007, Busan, Korea, November 11–15, 2007*. Berlin / Heidelberg: Springer.

Bateman, J.A. (1991). The theoretical status of ontologies in natural language processing. En Susanne Preuss y Birte Schmitz (Eds.), *Text Representation and Domain Modelling - Ideas from Linguistics and AI, KIT-Report 97* (pp. 50-99). Berlin: Technische Universitaet Berlin.

Bizer C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R. y Hellmann, S. (2009). DBpedia – A Crystallization Point for the Web of Data. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web, Issue 7*, 154–165.

Corcho, O., Fernández López, M. y Gómez Pérez, A. (2001). *Technical Roadmap v. 1.0, IST-OntoWeb Project*. Madrid: Universidad Politécnica de Madrid.

Giles, J. (2008, 31 Mayo). Birth pangs for the 'semantic web'. *New Scientist, Issue 2658*.

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition. Vol. 5 (2)*, 199-220.

Guarino, N. y Giaretta, P. (1995). Ontologies and knowledge bases. Towards a terminological clarification. En N.J.I. Mars (Ed.), *Towards Very Large Knowledge Bases* (pp. 25-32). Amsterdam / Tokio: IOS Press.

Jentzsch, A. (2009). DBpedia – Extracting structured data from Wikipedia. Presentación en *Semantic Web In Bibliotheken (SWIB2009)*, Colonia, Alemania, Noviembre 2009. Extraído de http://www.anjajentzsch.de/slides/SWIB09_DBpedia.pdf

Lakoff, G. (1987). *Women, Fire, and Dangerous Things. What Categories Reveal about the Mind*. Chicago / Londres: The University of Chicago Press.

Langacker, R. (1991). *Foundations of Cognitive Grammar. Vol. 2: Descriptive Application*. Stanford: Stanford University Press.

Langacker, R. W. (1987). *Foundations of Cognitive Grammar. Vol. I: Theoretical Prerequisites*. Londres: Longman.

Lenat, D.B., Guha, R.V., Pittman, K., Pratt, D. y Shepherd, M. (1990). CyC: toward programs with common sense. *Communications of the ACM, vol. 33, nº 8*, 30-49.

Liu, H. y Singh, P. (2004a). Commonsense reasoning in and over natural language. *Proceedings of the 8th International Conference on Knowledge-Based Intelligent Information & Engineering Systems (KES'2004)* (pp. 293-306). Wellington, New Zealand: Springer.

Liu, H. y Singh, P. (2004b). ConceptNet-a practical commonsense reasoning tool-kit. *BT Technology Journal, vol. 22, nº 4*, 211-226.

Mairal Usón, R. y Periñán Pascual, C. (2009a). Role and Reference Grammar and Ontological Engineering. *Volumen Homenaje a Enrique Alcaraz*. Alicante: Universidad de Alicante.

Mairal Usón, R. y Periñán Pascual, J.C. (2009b). The anatomy of the lexicon within the framework of an NLP knowledge base. *Revista española de lingüística aplicada, Vol. 22*, 217-244.

Mairal Usón, R. y Periñán Pascual, J.C. (2010). Teoría lingüística y representación del conocimiento: una discusión preliminar. En Dolores García Padrón y María del Carmen Fumero Pérez (Eds.), *Tendencias en lingüística general y aplicada* (pp. 155-168). Berlín: Peter Lang.

Mairal Usón, R. y Ruiz de Mendoza Ibáñez, F. (2009). Levels of description and explanation in meaning construction. En Christopher Butler y Javier Martín Arista (Eds.), *Deconstructing Constructions*. Amsterdam/Philadelphia: John Benjamins.

Minsky, M. (1974). A Framework for Representing Knowledge. *MIT-AI Laboratory Memo 306, June, 1974.*

Minsky, M. (2000). Commonsense-based interfaces. *Communications of the ACM 43, nº 8*, 67-73.

Noy, N. F. y McGuinness, D.L. (2001). *Ontology Development 101: A Guide to Creating Your First Ontology, Technical Report KSL-01-05.* Stanford Knowledge Systems Laboratory, Stanford: Stanford University.

Panton, K., Matuszek, C., Lenat, D., Schneider, D., Witbrock, M., Siegel, N. y Shepard, B. (2006). Common sense reasoning - from Cyc to intelligent assistant. En Yang Cai y Julio Abascal (Eds.), *Ambient Intelligence in Everyday Life* (pp. 1-31). Berlin: Springer.

Periñán Pascual, C. y Arcas Túnez, F. (2004). Meaning postulates in a lexico-conceptual knowledge base, *15th International Workshop on Databases and Expert Systems Applications, IEE, Los Alamitos (California),* 38-42.

Periñán Pascual, C. y Arcas Túnez, F. (2006). Reusing Computer-oriented Lexica as Foreign-Language Electronic Dictionaries. *Anglogermanica online: Revista electrónica periódica de filología alemana e inglesa*, *nº 4,* 69-93.

Periñán Pascual, C. y Arcas Túnez, F. (2007a). Cognitive modules of an NLP knowledge base for language understanding. *Procesamiento del Lenguaje Natural, nº 39,* 197-204.

Periñán Pascual, C. y Arcas Túnez, F. (2007b). Deep semantics in an NLP knowledge base. En Daniel Borrajo, Luis Castillo y Juan Manuel Corchado (Eds.), *12th Conference of the Spanish Association for Artificial Intelligence (*pp. 279-288). Salamanca: Universidad de Salamanca.

Periñán Pascual, C. y Arcas Túnez, F. (2010a). Ontological commitments in FunGramKB. *Procesamiento del Lenguaje Natural, nº 44,* 27-34.

Periñán Pascual, C. y Arcas Túnez, F. (2010b). The Architecture of FunGramKB. En Nicoletta Calzolari et al. (Eds.), *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA).

Periñán Pascual, C. y Mairal Usón, R. (2009). Bringing Role and Reference Grammar to natural language understanding. *Procesamiento del lenguaje natural, nº 43*, 265-273.

Periñán Pascual, C. y Mairal Usón, R. (2010). La gramática de COREL: un lenguaje de representación conceptual. *Onomázein 21*.

Rosch, E.H. (1973). Natural categories. *Cognitive Psychology 4*, 328-50.

Ruiz de Mendoza Ibáñez, F.J. y Mairal Usón, R. (2008). Levels of description and constraining factors in meaning construction: an introduction to the Lexical Constructional Model. *Folia Linguistica. Volume 42, Issue 3-4,* 355–400.

Suchanek, F., Kasneci, G. y Weikum, G. (2007). YAGO: A Core of Semantic Knowledge - Unifying WordNet and Wikipedia. *16th International World Wide Web Conference (WWW 2007)*, 697-706.

Uschold, M. y Gruninger, M. (1996). Ontologies: principles, methods and applications. *Knowledge Engineering Review 11, nº 2*, 93-136.

APÉNDICE I.

REGLAS DE LA CLASE "*PLACE*" DE DBPEDIA EN EL ONOMASTICÓN DE FUNGRAMKB

1	TYPE: snapshot
CLASE	Place
PROPIEDAD	/areaLand
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2:+BE_02 (x3:+LAND_00)Theme (x1)Location (f1:+IN_00)Position) *(e3:+BE_01 (x3)Theme (x4:+BIG_00)Attribute (f2: <VAL> $SQUARE_METRE_00)Quantity)

2	TYPE: snapshot
CLASE	Place
PROPIEDAD	/areaTotal
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+BIG_00)Attribute (f1: <VAL> $SQUARE_METRE_00)Quantity)

3	TYPE: snapshot
CLASE	Place
PROPIEDAD	/areaWater
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2:+BE_02 (x3:+AREA_OF_WATER_00)Theme (x1)Location (f1:+IN_00)Position) *(e3:+BE_01 (x3)Theme (x4:+BIG_00)Attribute (f2: <VAL> $SQUARE_METRE_00)Quantity)

4	TYPE: snapshot
CLASE	Place
PROPIEDAD	/coordinates
ESQUEMA COREL	Se prescinde de esta propiedad por no resultar relevante para el PLN.

5	TYPE: snapshot
CLASE	Place
PROPIEDAD	/depth
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+DEEP_00)Attribute (f1: <VAL> +METRE_00)Quantity)

6	TYPE: snapshot
CLASE	Place
PROPIEDAD	/elevation
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:$ALTITUDE_00)Attribute (f1: <VAL> +METRE_00)Quantity)

7	TYPE: snapshot
CLASE	Place
PROPIEDAD	/height
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+HIGH_00)Attribute (f1: <VAL> +METRE_00)Quantity)

8	TYPE: snapshot
CLASE	Place
PROPIEDAD	/length
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+LONG_00)Attribute (f1: <VAL> +METRE_00)Quantity)

9	TYPE: snapshot
CLASE	Place
PROPIEDAD	/location
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_02 (x1)Theme (x3:<VAL>)Location (f1:+IN_00)Position)

10	TYPE: snapshot
CLASE	Place
PROPIEDAD	/maximumDepth
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3: +DEEP_00)Attribute (f1: <VAL> +METRE_00)Quantity (f2: (e3:+BE_01 (x1)Theme (x4: m +DEEP_00)Attribute))Scene)

11	TYPE: snapshot
CLASE	Place
PROPIEDAD	/maximumElevation
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3: $ALTITUDE _00)Attribute (f1: <VAL> +METRE_00)Quantity (f2: (e3:+BE_01 (x1)Theme (x4: m $ALTITUDE _00)Attribute))Scene)

12	TYPE: snapshot
CLASE	Place
PROPIEDAD	/minimumElevation
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3: $ALTITUDE _00)Attribute (f1: <VAL> +METRE_00)Quantity (f2: (e3:+BE_01 (x1)Theme (x4: p $ALTITUDE _00)Attribute))Scene)

13	TYPE: snapshot
CLASE	Place
PROPIEDAD	/nativeName /nickname /otherName
ESQUEMA COREL	No se precisa creación de regla, ya que es una redirección a la que se accederá a través del "*alternative name box*" que se creará a tal fin en el Onomasticón de FunGramKB.

14	TYPE: snapshot
CLASE	Place
PROPIEDAD	/nearestCity
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_02 (x3:<VAL>)Theme (x1)Location (f1: m +NEAR_00)Position)

15	TYPE: snapshot
CLASE	Place
PROPIEDAD	/percentage Of AreaWater
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) +(e2:+BE_02 (x3:+AREA_OF_WATER_00)Theme (x1)Location (f1:+IN_00)Position) *(e3:+BE_01 (x3)Theme (x4:+BIG_00)Attribute (f2: <VAL> $PERCENTAGE_00)Quantity)

16	TYPE: snapshot
CLASE	Place
PROPIEDAD	/type
ESQUEMA COREL	Se prescinde de regla para esta propiedad, ya que resulta redundante enumerar de nuevo el tipo, puesto que aparecerá en la jerarquía de la Ontología de DBpedia (clase y subclase) presente en el Onomasticón de FunGramKB a través del DBpedia Mapper.

17	TYPE: snapshot
CLASE	Place
PROPIEDAD	/width
ESQUEMA COREL	+(e1:+BE_00 (x1: <C>)Theme (x2:+PLACE_00)Referent) *(e2:+BE_01 (x1)Theme (x3:+WIDE_00)Attribute (f1: <VAL> +METRE_00)Quantity)