

NAVEGANDO EL CONOCIMIENTO EN LA WEB

En los últimos años la Web ha evolucionado desde un repositorio inicialmente sólo de documentos a una Web donde además hay datos, y donde los objetos descritos por datos estructurados se hallan interconectados. Existen diversos esfuerzos por transformar las partes relevantes de la Web en una base de datos enorme. Esto significa esencialmente publicar, describir, organizar e interconectar los datos existentes en la Web.





VALERIA FIONDA

Investigadora Fellow en el Departamento de Matemáticas y Computación de la Universidad de Calabria, Italia. Hizo su MSc en Computing Engineering en 2006 y su Ph.D. en Matemáticas y Computación en 2010 en la Universidad de Calabria. Sus intereses de investigación incluyen Bioinformática, E-commerce, Algoritmos para Manipulación de Grafos, Minería de Procesos y Web Semántica. Ha publicado artículos en conferencias y journals de primer nivel como WWW, IJCAI, CIKM, AIJ.

fionda@mat.unical.it



CLAUDIO GUTIÉRREZ

Profesor Asociado Departamento de Ciencias de la Computación, Universidad de Chile. Ph.D. Computer Science, Wesleyan University; Magister en Lógica Matemática, Pontificia Universidad Católica de Chile; Licenciatura en Matemáticas, Universidad de Chile. Áreas de investigación: Fundamentos de la Computación, Lógica aplicada a la Computación, Bases de Datos, Semántica de la Web, Máquinas Sociales.

cgutierrez@dcc.uchile.cl



GIUSEPPE PIRRO

Investigador postdoctoral senior en el Instituto WeST, Universidad de Koblenz, Alemania. Antes fue Profesor Asistente en la Universidad Libre de Bolzano, Italia y postdoctorado en Inria Grenoble, Francia. Obtuvo su Ph.D. en Computer Engineering en el Departamento DEIS, Universidad de Calabria, Italia, en 2009. Sus intereses de investigación se enfocan en la Web Semántica, Lenguajes de Consulta para Grafos, Inteligencia Artificial y Sistemas Distribuidos. Ha escrito más de 40 papers publicados en conferencias y journals de primer nivel como WWW, ISWC, AAAI, CIKM, CCGRID, TWEB y FGCS.

pirro@uni-koblenz.de

Estas nuevas tendencias pavimentan el camino para servicios que –aprovechando las técnicas de bases de datos sobre datos estructurados e interconectados a escala planetaria– puedan darle más valor a la Web descubriendo nuevo conocimiento. Nuevos desafíos de investigación emergen en este nuevo mundo debido a nuevas realidades: la creación y manipulación de los datos intrínsecamente descentralizada; la carencia de esquemas superimpuestos; el desconocimiento de la topología de los recursos y sus conexiones; la existencia de enormes volúmenes de información cubriendo áreas y dominios diferentes. Este grafo de datos estructurados –ilimitado y distribuido– es conocido hoy como **la Web de los Datos**.

Entre las iniciativas de esta nueva Web, podemos mencionar la extracción de información desde datos Web textuales o semiestructurados [23] y comunidades que comparten conocimiento (e.g., Wikipedia); proyectos como las *Web tables* de Google que aspiran a recobrar pequeñas bases de datos relacionales de las tablas HTML [10]; las iniciativas YAGO [22] y DBpedia

[7] para construir bases de conocimiento que recolecten datos sobre entidades y sus relaciones desde fuentes Web; y desde el mundo de los datos abiertos y la Web Semántica, la iniciativa que intenta la organización e interconexión sistemática de datos semánticos estructurados en la Web, llamada *Linked Open Data* [9].

Una herramienta esencial para descubrir fuentes de datos, información y conocimiento en este grafo gigante es la navegación, el procedimiento de ir, guiado por un mapa, desde lo conocido a lo desconocido en algún espacio. Por ejemplo, considere cómo la navegación desde el nodo *Stanley Kubrick* en DBpedia hasta el nodo *A Clowckwork Orange* nos permite descubrir conocimiento adicional tal como los actores que actuaron en la película.

Este artículo discute cómo realizar navegación de conocimiento en la Web de los datos y los desafíos que acarrea para el manejo de los datos. A modo de ilustración, nos concentraremos en el escenario de *Linked Open Data* (LOD).

NAVEGAR VERSUS CONSULTAR

La herramienta tradicional para consultar datos estructurados es un lenguaje de consultas. Consultar implica acceder a la fuente de datos para satisfacer una necesidad expresada en algún lenguaje formal. La topología de la fuente de los datos está fija a priori. Por otro lado, la navegación consiste en la especificación de partes de las fuentes de datos que tienen que ser "exploradas" o descubiertas para encontrar datos relevantes. Por lo tanto, cuando la topología del espacio de fuentes de datos a ser consultados es (desde un punto de vista práctico) ilimitada, desconocida en su identidad, y dinámica, como en el caso de la Web de Datos, la navegación se vuelve necesaria. En el mundo del manejo de datos, la noción de navegación se volvió popular con el lenguaje de consultas XQuery, cuya filosofía subyacente es permitir la especifica-

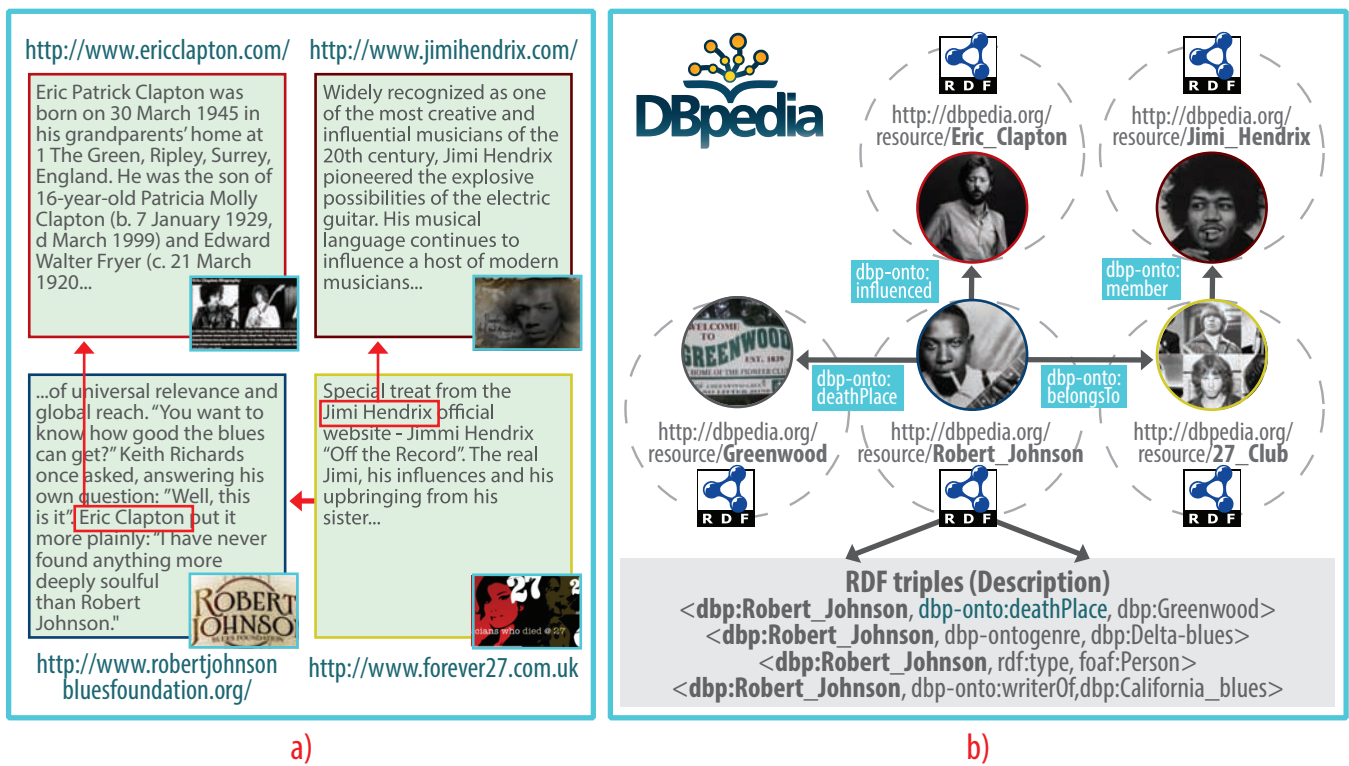


FIGURA 1. EXTRACTO DE WIKIPEDIA Y SU CONTRAPARTE DBPEDIA.

ción de una ruta en un grafo (árbol) de datos, en oposición a los lenguajes de consulta clásicos (e.g., SQL) que representaban afirmaciones lógico/algebraicas. Por lo tanto, la navegación es una función complementaria de consulta, especialmente cuando se trata con grafos. En particular, en la *Web navigare necesse est, vivere non est necesse*.

Trabajar con navegación en la Web crea varios desafíos. Primero, el grafo de la Web está formado por un gigantesco número de fuentes de datos interconectadas que no están disponibles *off-the-shelf*. Por otro lado, la disponibilidad de datos estructurados en cada fuente sugiere que la navegación puede y debe ser productivamente combinada con consultas. De hecho, satisfacer una necesidad de información en este escenario incluye dos facetas que debieran combinarse: el descubrimiento de fuentes relevantes de datos vía navegación; la obtención de respuestas precisas desde esas fuentes a través de consultas.

LA WEB DE LOS DATOS

La Web tradicional estaba basada en tres pilares: identificadores únicos para nombrar recursos (URIs), el protocolo HTTP para intercambiar información, y el lenguaje HTML para describir páginas. La Web de datos extiende este modelo de dos maneras. Primero, se extiende el alcance de las URIs para nombrar nuevos tipos de recursos tales como objetos del mundo real (personas, lugares, equipos de fútbol) y conceptos abstractos (deporte, filosofía, geografía) [14]. Descripciones de tales recursos pueden ser obtenidas de la misma forma que para los documentos tradicionales, esto es, dereferenciando sus URIs asociados a través del protocolo HTTP. Segundo, se introducen *links* semánticos, en vez de los *links* a secas de la Web tradicional, de manera de poder describir y relacionar entidades significativamente.

La **Figura 1** muestra un extracto de Wikipedia y su contraparte DBpedia. Mientras el primero está basado en documentos y sus *hiperlinks*, el segundo está basado en recursos y descrip-

ciones semánticas. En la **Figura 1**, cada círculo tachado representa una fuente de datos, identificada por una URI, que contiene la descripción del recurso en el lenguaje formal RDF. Por ejemplo, en la fuente de datos asociada con el cantante Robert Johnson, hay un triple RDF que nos dice que Johnson falleció en la ciudad de Greenwood. Observe el *link* semántico entre los recursos correspondientes expresado a través de la propiedad **dbp-onto:deathPlace** (la que es definida en la ontología de DBpedia).

El conocimiento generado distribuidamente puede de esta forma ser interconectado, de tal forma de crear una Web de Datos donde el valor de las piezas individuales de datos aumenta en la medida que más preguntas pueden ser respondidas y más gente pueda acceder a ellos. Los que publican los datos mantienen miles de fuentes de datos que cubren diversos dominios, tales como conocimiento general (e.g., DBpedia, Freebase/Google, YAGO), información geoespacial (e.g., Geonames), información del sector público (e.g., los Gobiernos de EE.UU., UK, Chile, etc.) y así sucesivamente. Iniciativas como la de microformatos y RDFa están haciendo difusa la distinción entre Web de Documentos y Web de Datos al permitir insertar "meta-información" (triples RDF) a nivel de páginas Web.

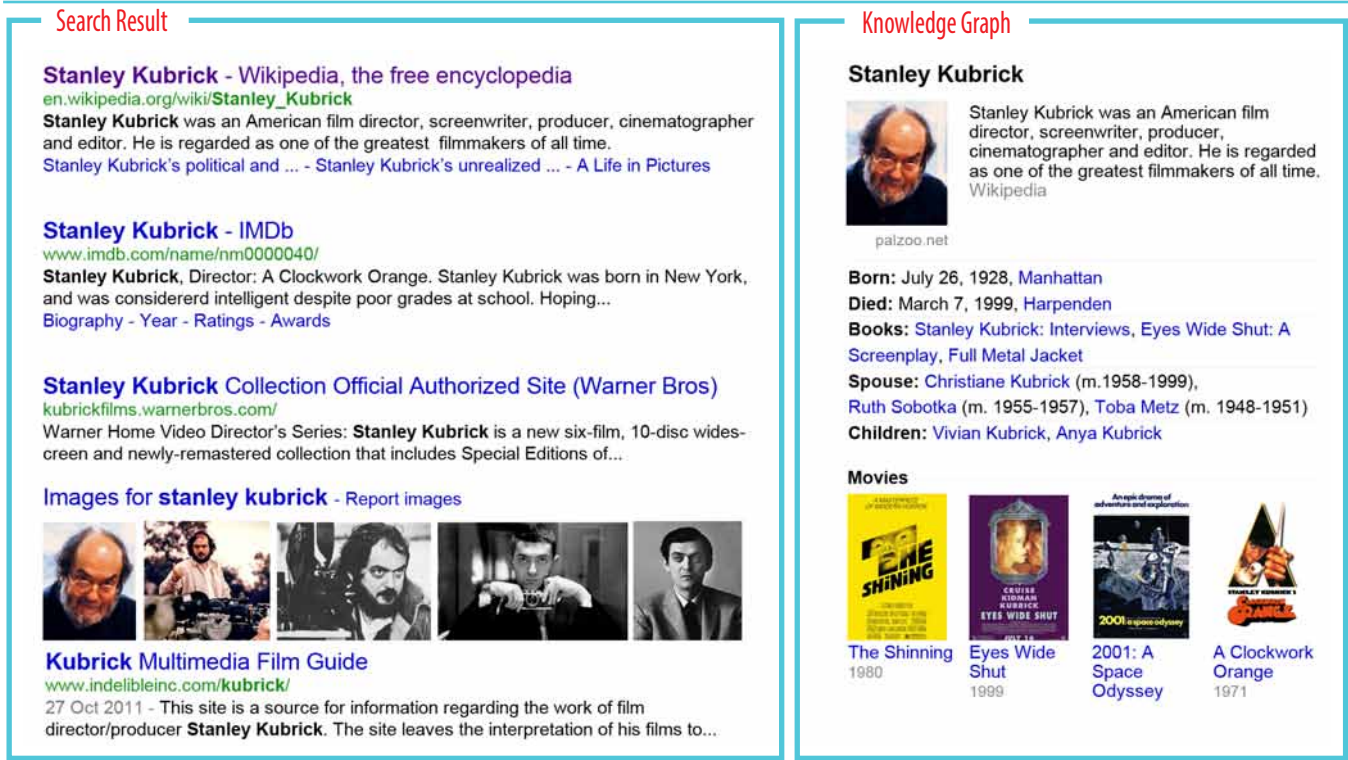


FIGURA 2. RESULTADOS DE BÚSQUEDA DE STANLEY KUBRICK EN GOOGLE.

CONECTANDO LOS PUNTOS

Aplicaciones tales como *Facebook Graph* (FG) [2] y *Google's Knowledge Graph* (KG) [3] reconocen la importancia de enriquecer el paradigma de búsqueda tradicional al tratar a los grafos semánticos como ciudadanos de primera clase. Los nodos de FG almacenan información sobre entidades junto con relaciones con otras entidades. Para acceder FG, Facebook provee de una interfaz de búsqueda basada en el lenguaje natural, donde los usuarios pueden preguntar, por ejemplo, "gente que vive en mi ciudad". KG de Google, con el lema "*Things not Strings*", resalta la necesidad de considerar una palabra ya no meramente como un string, sino como identificadores para entidades del mundo real (e.g., personas) en el grafo. Para dar un ejemplo, la **Figura 2** muestra los resultados de una búsqueda de Stanley Kubrick en Google. La parte izquierda reporta el clásico conjunto plano de páginas Web que contienen esa *keyword*. El KG que se muestra en la parte derecha lista las películas de Kubrick, la ciudad donde nació y así

sucesivamente, con *links* a las entidades correspondientes en el KG. Al seguir cada uno de esos *links* es posible "descubrir" nuevo conocimiento. FG y KF ofrecen un atisbo de las potencialidades de la navegación sobre una Web de entidades interconectadas. En particular, estos sistemas intentan tratar una de las limitaciones de la búsqueda tradicional: el descubrimiento del *unknown unknown*, esto es, algo que uno no sabe que no sabe.

EL MÉTODO LOD

Sistemas como FG y KG utilizan modelos propietarios, grafos construidos de forma ad-hoc, y actualmente no pueden procesar peticiones complejas de navegación incluyendo la cobertura de múltiples fuentes de datos distribuidas. LOD adopta una filosofía diferente. Links semánticos son introducidos mediante un lenguaje estándar de descripción semántica llamado Resource Description Framework (RDF). Este es un lenguaje simple y extensional basado en triples de la forma sujeto-predicado-objeto. Por ejemplo, el triple RDF (Kubrick, birthPlace, Manhattan) expresa que Stanley Kubrick nació en Manhattan.

Una colección de triples RDF puede ser pensado como un grafo etiquetado, el cual debido a su uso de taxonomías estándar, vocabularios y ontologías expresadas en lenguajes formales tales como RDF(S), SKOS [8] u OWL [18], permite describir, integrar e intercambiar conocimientos de áreas de dominio específico. A las entidades se les asignan identificadores (en la forma de URIs) y se generan descripciones distribuidas y autónomas que pueden ser parte de cualquier fuente de datos. Por lo tanto, usando las palabras de Tim Berners-Lee, cualquiera puede decir cualquier cosa sobre cualquier cosa y publicar donde quiera. La **Figura 3** muestra un extracto de datos incluyendo información sobre Stanley Kubrick, David Lynch, Quentin Tarantino y las películas *Path of Glory* y *A.I.* Cada rectángulo gris tachado representa una fuente de datos que contiene un conjunto de triples RDF (representados con el mismo color) que pueden ser obtenidos al dereferenciar la URI correspondiente (representada como un círculo tachado). Por ejemplo, al dereferenciar Stanley Kubrick en DBpedia uno obtiene, entre otras cosas, el triple `<dbpo: Stanley Kubrick, dbpo: birthPlace, dbpo: Manhattan>`. Interesantemente, la información ha sido obtenida desde tres fuentes de datos diferentes.

dbpo: <http://dbpedia.org/ontology/>
 dbp: <http://dbpedia.org/>

Imdb: <http://linkedmdb.org/>
 owl: <http://www.w3.org/2002/07/owl/>
 fb: <http://rdf.freebase.com/ns/>

rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
 foaf: <http://xmlns.com/foaf/spec/>

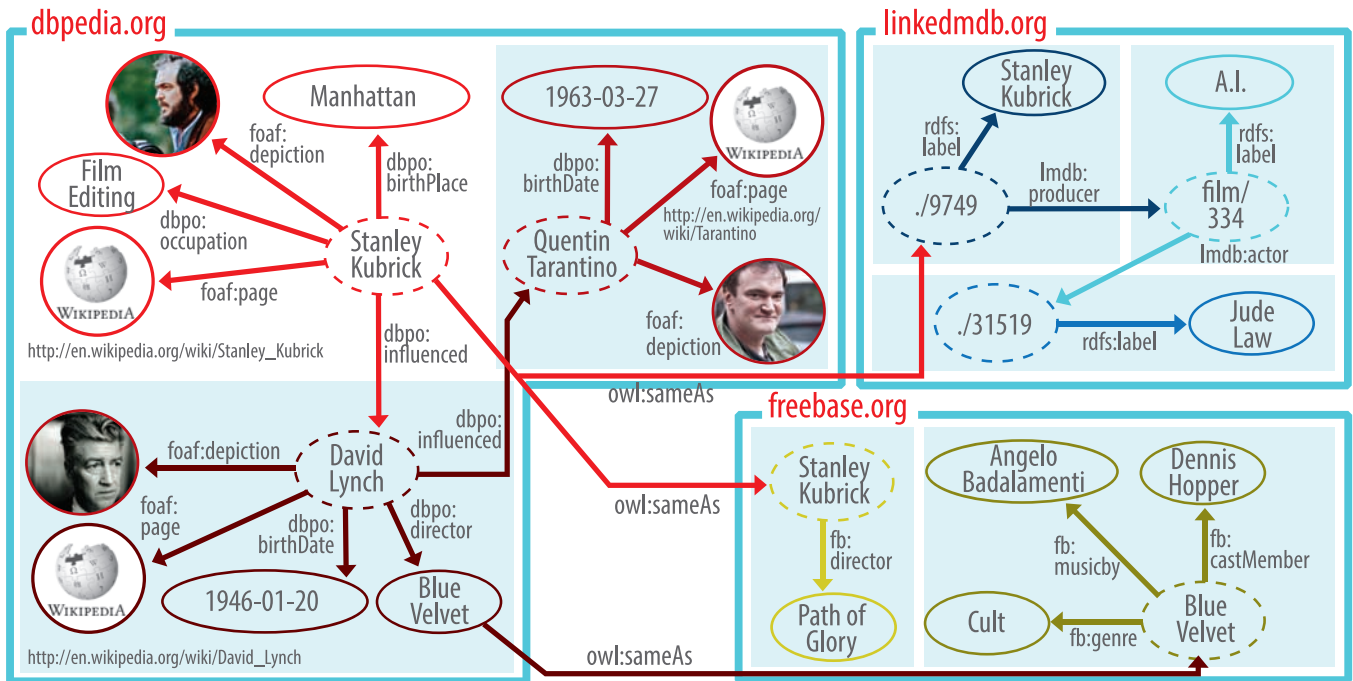


FIGURA 3. EXTRACTOS DE DATOS DE DBPEDIA.

La interconexión entre los *datasets* permite juntar datos de múltiples fuentes. En la Web de Datos, el predicado `owl:sameAs` es usado para unir dos URLs que identifican la misma "cosa". Por ejemplo, el recurso Stanley Kubrick en DBpedia, que provee conocimiento general, está unido con aquel en LinkedMDB, el cual es una fuente especializada de conocimiento sobre películas, directores y actores. Esto permite descubrir conocimiento adicional, como el hecho que Jude Law actuó en la película A.I. de la cual Stanley Kubrick fue el productor.

directa o indirectamente por Stanley Kubrick" o "encuentre los actores de menos de cincuenta años que hayan actuado en películas de Kubrick". Para consultar grafos RDF, existe un lenguaje de consulta estándar, llamado SPARQL [21]. Se parece a SQL enriquecido con un protocolo para consultar datos estructurados en la Web. Las debilidades de SPARQL son las mismas que las de cualquier lenguaje cuando enfrenta una topología que no está fija y de creación distribuida. Por tanto, no es ninguna sorpresa que varios lenguajes de consulta y navegación de grafos hayan sido propuestos para solucionar parcialmente la navegación semántica en la Web.

para especificar navegación sobre las fuentes de datos en la Web. Otras iniciativas optaron por enriquecer el lenguaje de consulta SPARQL con operadores navegacionales [20,6,12] mediante la consideración de un conjunto fijo de fuentes de datos (típicamente un único grafo RDF) pero con navegación a escala Web. Finalmente, otros métodos extienden el alcance de las consultas SPARQL mediante operadores de navegación [13] que permiten descubrir fuentes de datos relevantes para la consulta, aunque carecen de maneras de especificar la navegación. Es importante recalcar que ninguno de los métodos mencionados arriba incorpora acciones, una funcionalidad que, como veremos, se vuelve importante al momento de navegar.

CONSULTANDO LA WEB DE DATOS

Para acceder la información en la Web de Datos no es suficiente contar con un lenguaje de consultas. Como ya mencionamos, aunque las iniciativas como KG enriquecen los resultados de las búsquedas, sus capacidades de responder consultas son aún limitadas. Por ejemplo, no es posible expresar peticiones simples como "encuentre los directores italianos influenciados

La especificación (y recuperación) de colecciones de sitios fue tempranamente tratada por herramientas como `wget`, que permite recolectar recursivamente información de sitios Web ("crawling"). El problema es que, aparte de no ser declarativa, esta herramienta se halla restringida solo a funcionalidades sintácticas. También existen *crawlers* semánticos como LDSpider [17] y técnicas de indexamiento como SWSE [16] y Sindice [19]. Estos métodos se enfocan en el *crawling* eficiente y masivo para construir repositorios centralizados y, por tanto, no se concentran en los lenguajes declarativos

NAVEGACIÓN EN LA WEB DE DATOS

Generalmente hablando, un lenguaje de navegación de grafos (ver [24] para un reciente survey) permite encontrar pares de nodos que están conectados por una secuencia de etiquetas

de arcos que satisfacen cierto patrón. La forma más común de especificar un patrón (o expresión navegacional) es vía una expresión regular sobre el alfabeto de etiquetas, comenzando la navegación en un nodo semilla. En el ejemplo previo sobre KG, el nodo semilla es Stanley Kubrick (o, más precisamente, su identificador interno en el KG). Sin embargo, el mecanismo de navegación estaba limitado: no era posible especificar patrones para seleccionar información relevante. KG sólo provee un conjunto de nodos relacionados (e.g., la película *The Shining*) desde los cuales es posible continuar la navegación manualmente. La Web de Datos, gracias al lenguaje de descripción RDF, permite la navegación semántica de grafos a escala Web.

En lo que sigue, presentamos un lenguaje navegacional para la Web de Datos llamado NautiLOD [11]. Este lenguaje está basado en expresiones regulares sobre predicados RDF entremezclados con tests del tipo ASK en SPARQL realizados sobre la descripción RDF de los recursos. Las expresiones regulares permiten expresar necesidades de información complejas que requieren de navegación a través de los nodos del grafo, mientras los tests permiten la selección de fuentes de datos relevantes desde las cuales continuar la navegación. NautiLOD también presenta un mecanismo que gatilla acciones (e.g., enviar mensaje de notificación) usando los datos encontrados durante la navegación. Discutiremos ejemplos del mundo real y presentaremos una implementación en la herramienta *swget*, la cual está disponible online¹.

NAUILOD

Para dar un esbozo de las potencialidades del lenguaje NautiLOD, mostraremos algunos ejemplos considerando el extracto de datos mostrado en la **Figura 3**. Una presentación detallada de la sintaxis será provista en la siguiente sección.

Imagine que queremos descubrir todo lo que se predica sobre Stanley Kubrick en distintas fuentes de datos. La idea es considerar los arcos

`<owl:sameAs>` que comienzan en el identificador de Kubrick en DBpedia. Se chequean entonces todos los triples de la forma `<dbpo: Stanley Kubrick, owl:sameAs, v>`, y se seleccionan todos los `v`'s encontrados. Finalmente, por cada uno de esos `v`'s, se retornan todas las URIs `w` de los triples de la forma `<v, p, w>` que se encuentren en la fuente de datos de `v`.

Lo anterior puede ser especificado utilizando NautiLOD mediante la expresión:

```
<dbpo:Stanley Kubrick>
<owl:sameAs>/ <_>
```

Aquí, la expresión `<_>` denota un wildcard para los predicados RDF. Al evaluar esta expresión comenzando desde la URI `dbpo: Stanley Kubrick` obtenemos todas las distintas representaciones de Stanley Kubrick provistas por `dbpedia.org`, `freebase.org` y `linkedmdb.org`. Desde estos nodos, la expresión `<_>` se puede instanciar en cualquier predicado. El resultado de la evaluación es `{lmdb:/film/334, fb:Path of Glory}`. Esto remarca cómo NautiLOD es capaz de tratar con fuentes de datos distribuidos y dinámicas. Un ejemplo más complejo que realiza acciones sobre los datos se presenta a continuación. En él intentaremos encontrar aquellas películas (y sus alias) cuyo director tiene más de cincuenta años y ha sido influenciado, ya sea directa o indirectamente, por Stanley Kubrick. Cada vez que uno de esos directores es detectado enviaremos su sitio Wiki por email.

Esta especificación trata sobre caminos de *influencia* y alias como en el ejemplo anterior; tests (expresados en NautiLOD usando consultas del tipo ASK en SPARQL) sobre la fuente de datos asociada con la URI dada (si alguien influenciado por Kubrick es encontrado, verifica si él/ella es de la edad apropiada); y acciones a realizarse usando datos de la fuente. La especificación en NautiLOD que expresa el ejemplo mencionado en el párrafo anterior es:

```
<dbpo:Stanley Kubrick>
(<dbpo:influenced>)+[Test]/Act/
<dbpo:director>/<owl:sameAs>
```

En la expresión, el símbolo `+` denota que uno o más niveles de influencia son aceptables, e.g., obtenemos directores como David Lynch o Quentin Tarantino. Desde este conjunto de recursos, la restricción en la edad forzada por la consulta ASK es evaluada en la fuente de datos asociada con cada uno de los recursos ya matcheados. Este filtro deja en este caso solo `dbpo: DavidLynch`. En este momento, sobre los elementos del conjunto (un elemento en este caso), la acción enviará vía email la página Wiki (obtenida de la consulta SELECT).

La acción **sendEmail**, implementada por un procedimiento de programación ad-hoc, no influencia el proceso de navegación. Por tanto, la evaluación continuará desde la URI `u = dbpo: DavidLynch`, al navegar la propiedad `dbpo:director` (encontrada en el conjunto de datos al dereferenciar `u`). Es posible hacer esto, por ejemplo, al seguir el triple `<u, dbpo:director, dbpo: BlueVelvet>`. Luego, desde `dbpo: blueVelvet`, la parte final de la expresión es evaluada. El resultado de la evaluación es: (1) el conjunto `{dbpo: BlueVelvet, fb: BlueVelvet}`, esto es, los datos acerca de la película *BlueVelvet* que se obtienen de `dbpedia.org` y `freebase.org`; (2) el conjunto de acciones realizadas, en este caso un email enviado.

LA SINTAXIS DE NAUILOD

NautiLOD provee un mecanismo para declarativamente: (1) definir expresiones de navegación; (2) permitir el control semántico de la navegación; (3) realizar acciones como efectos colaterales a partir de los caminos de navegación. El reducto navegacional del lenguaje se basa en expresiones regulares de caminos, casi de la misma forma que algunos lenguajes de consulta para la Web y XPath. El control semántico es realizado a través de tests existenciales usando consultas SPARQL del tipo ASK sobre fuentes de datos RDF. Este mecanismo permite orientar la navegación basado en la información presente en cada nodo del camino. Finalmente, se gati-

¹ <http://swget.wordpress.com>

llan acciones de acuerdo a decisiones basadas en la especificación original y a la información local encontrada en cada fuente de datos.

La sintaxis del lenguaje está definida con respecto a la gramática mostrada en la Tabla XX. El lenguaje está basado en expresiones de camino (i.e., **path**), esto es, concatenación de expresiones de caso base construidas sobre predicados (i.e., **pred**), tests (i.e., **tests**) y acciones (i.e., **action**). Las expresiones complejas son disyunciones de expresiones: (1) expresiones que utilizan un número de repeticiones utilizando las funcionalidades de las expresiones regulares, y (2) expresiones con tests.

Los bloques sobre los que se construye una expresión NautiLOD son:

1. Predicados: el caso base de **pred** puede ser un predicado RDF o el wildcard **<_>** que denota cualquier predicado.
2. Expresiones Test: un **test** denota una ex-

presión de consulta. Su caso base es una consulta ASK en SPARQL.

3. Expresiones Action: un **action** es una especificación procedural de un comando (e.g., enviar un mensaje de notificación, un comando GET, etc.), el cual obtiene sus parámetros de la fuente de datos alcanzada durante la navegación. Es un efecto colateral, esto es, no influencia el proceso subsiguiente de navegación.

Si se restringe a (1) y (2), NautiLOD puede ser visto como un lenguaje declarativo para describir porciones de la Web de Datos, i.e., un conjunto de URIs que se ajusta a cierta especificación semántica. Las expresiones NautiLOD son evaluadas contra la Web de Datos comenzando en una segunda URI **u**. El significado de una expresión NautiLOD es un conjunto de URIs más un conjunto de acciones gatilladas por la evaluación. Para una discusión más comprensiva de la semántica y complejidad del lenguaje el lector puede referirse a Fionda et al. 2012 [11].

SWGET

El comando `swget` es una implementación Java de NautiLOD. La herramienta está libremente disponible en el sitio web `swget` (<http://swget.wordpress.com>) donde también se discuten otros ejemplos y una descripción detallada de sus principios de funcionamiento. La aplicación se halla disponible tanto como una herramienta de línea de comando y como una GUI. Un ejemplo de la herramienta `swget` GUI se muestra en la **Figura 4**. La evaluación de una expresión puede ser visualizada de distintas maneras (e.g., como conjunto de URIs, como grafo RDF). En la Figura 4 se muestra la visualización del grafo, donde las descripciones de los diferentes recursos visitados durante la navegación están interconectados. Es posible seleccionar partes específicas del grafo al filtrar predicados, buscar predicados o nodos específicos, y expandir los nodos proveyendo un valor del radio a partir desde un nodo objetivo.

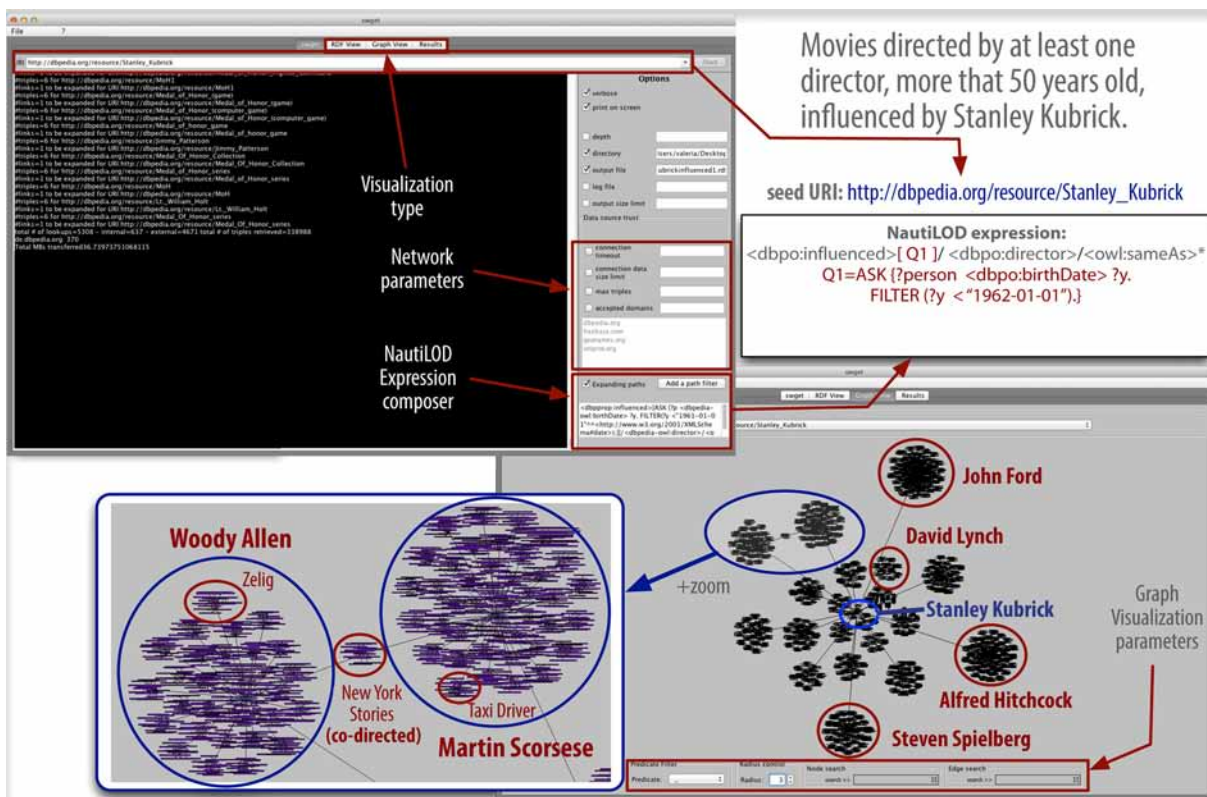


FIGURA 4. EJEMPLO DE LA HERRAMIENTA SWGET ([HTTP://SWGET.WORDPRESS.COM](http://swget.wordpress.com)).

CONCLUSIÓN

EL CONOCIMIENTO ESTRUCTURADO COMO GRAFO IMPREGNA NUESTRA VIDA COTIDIANA. INICIATIVAS COMO FACEBOOK OPEN GRAPH Y GOOGLE KNOWLEDGE GRAPH SON BUENOS EJEMPLOS DE ESTA TENDENCIA. SIN EMBARGO, ELLOS FALLAN A LA HORA DE CAPTURAR UN ASPECTO DEL CONOCIMIENTO ESTRUCTURADO COMO GRAFO: LA POSIBILIDAD DE LA NAVEGACIÓN DECLARATIVA. EN ESTE ARTÍCULO HEMOS DEMOSTRADO LA IMPORTANCIA DE TENER UN LENGUAJE DECLARATIVO PARA LA NAVEGACIÓN AUTOMÁTICA EN LA WEB. DADO QUE MILES DE FUENTES DE DATOS SEMÁNTICAS, DISTRIBUIDAS E INTERCONECTADAS ESTÁN HOY EN DÍA DISPONIBLES, LA NAVEGACIÓN SE VUELVE NAVEGACIÓN SEMÁNTICA. EL LENGUAJE NAUJOD Y SU IMPLEMENTACIÓN EN SWGET DEMUESTRAN LA POSIBILIDAD DE LA NAVEGACIÓN SEMÁNTICA Y AUTOMÁTICA A ESCALA WEB. NUESTRA EXPERIENCIA INDICA LA NECESIDAD DE CONTAR CON METADATOS SEMÁNTICOS ESTANDARIZADOS, LENGUAJES DE NAVEGACIÓN EXPRESIVOS PARA ESPECIFICAR CONOCIMIENTO EN LA WEB, ASÍ COMO LENGUAJES PARA ESPECIFICAR ACCIONES. POR ELLO PODEMOS DECIR: ¡UNA NUEVA ERA PARA LA WEB ESTÁ EMERGIENDO! ■

BIBLIOGRAFÍA

- [1] DBLP Bibliography Database: <http://dblp.l3s.de/d2r/>
- [2] Facebook Graph: <https://www.facebook.com/about/graphsearch>.
- [3] Google Knowledge Graph: <http://www.google.com/insidesearch/features/search/knowledge.html>
- [4] Google Scholar: <http://scholar.google.com>.
- [5] Microsoft Academic Research: <http://academic.research.microsoft.com>
- [6] F. Alkhateeb, J.-F. Baget, and J. Euzenat. Extending SPARQL with Regular Expression Patterns (for querying RDF). *Journal of Web Semantics*, 7(2):57-73, 2009.
- [7] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. DBpedia: A Nucleus for a Web of Open Data. *The Semantic Web*, pages 722-735, 2007.
- [8] S. L. Bechhofer and A. Miles. SKOS Simple Knowledge Organization System, 2009.
- [9] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *IJSWIS*, 5(3):1-22, 2009.
- [10] M.J. Cafarella, A. Halevy, and J. Madhavan. Structured data on the web. *Communications of the ACM*, 54(2):72-79, 2011.
- [11] V. Fionda, C. Gutiérrez, and G. Pirrò. Semantic Navigation on the Web of Data: Specification of Routes, Web Fragments and Actions. In *World Wide Web Conference*, pages 281-290. ACM, 2012.
- [12] S. Harris and A. Seaborne. SPARQL 1.1 Query Language, 2010.
- [13] O. Hartig, C. Bizer, and J.C. Freytag. Executing SPARQL Queries over the Web of Linked Data. In *International Semantic Web Conference*, pages 293-309, 2009.
- [14] T. Heath and C. Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Clay-pool, 2011.
- [15] J. E. Hirsch. An Index to Quantify an Individual's Scientific Research Output that Takes into Account the Effect of Multiple Coauthorship. *Scientometrics*, 85(3):741-754, 2010.
- [16] A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, and S. Decker. Searching and browsing linked data with swse: The semantic web search engine. *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(4):365-401, 2011.
- [17] R. Isele, A. Harth, J. Umbrich, and C. Bizer. LDspider: An OpenSource Crawling Framework for the Web of Linked Data. In *Poster- International Semantic Web Conference*, 2010.
- [18] D. L. McGuinness and F. van Harmelen. *OWL Web Ontology Language*, 2004.
- [19] E. Oren, R. Delbru, M. Catasta, R. Cyganiak, H. Stenzhorn, and G. Tummarello. Sindice.com: A document-oriented lookup index for open linked data. *Int. J. of Metad., Semant. and Ontolog.*, 3(1), 2008.
- [20] J. Pérez, M. Arenas and C. Gutiérrez. nSPARQL: A Navigational Language for RDF. *Journal of Web Semantics*, pages 255 - 270.
- [21] Eric Prud'hommeaux and Andy Seaborne. *SPARQL 1.1 Query Language*, 2008.
- [22] F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO - A Core of Semantic Knowledge. In *World Wide Web Conference*, pages 697-706. ACM, 2007.
- [23] G. Weikum, G. Kasneci, M. Ramanath, and F. M. Suchanek. Database and information-retrieval methods for knowledge discovery. *Communications of the ACM*, 52(4):56-64, April 2009.
- [24] P. T. Wood. Query Languages for Graph Databases. *SIGMOD Record*, 41(1):50-60, 2012.