UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

ESTÁNDARES DE PUBLICACIÓN DE DATOS PARA LA INFORMACIÓN PÚBLICA EN CHILE

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS,
MENCIÓN COMPUTACIÓN

DANIEL RICARDO HERNÁNDEZ HERNÁNDEZ

PROFESOR GUÍA:
CLAUDIO GUTIÉRREZ GALLARDO

MIEMBROS DE LA COMISIÓN:
AGUSTÍN VILLENA MOYA
PABLO BARCELÓ BAEZA
RENZO ANGLES ROJAS

Este trabajo ha sido parcialmente financiado por CONICYT
mediante el programa FONDECYT, proyecto Nº 1110287.

SANTIAGO DE CHILE
ABRIL DE 2013

Resumen

En las últimas décadas presenciamos un aumento exponencial en la generación y en el almacenamiento de datos en el mundo. En paralelo, han surgido movimientos promoviendo el acceso abierto a los datos. En Chile, por ejemplo, se dictó la Ley N°20285 sobre el acceso a la información pública, que obliga a las instituciones públicas a publicar cierto conjunto de información y define la forma en que el resto puede ser requerida. En esta dirección han surgido iniciativas que promueven además la publicación de datos en forma de datos enlazados. Este cambio de paradigma en el manejo de la información y datos requiere de nuevas políticas y estándares.

La presente tesis aborda los problemas observados en la publicación de datos, los conceptualiza y propone buenas prácticas para su publicación. Las prácticas propuestas se organizan en tres niveles, que van desde lo general a lo particular.

En el primer nivel se presentan prácticas generales, que son independientes de las tecnologías usadas y que tienen por objetivo satisfacer los principios de los datos abiertos.

En el segundo, se proponen prácticas para la publicación usando el lenguaje RDF. RDF es un estándar del Consorcio de la Web, y su modelo de grafos facilita la integración de datos, uno de los desafíos principales de la publicación de datos. Es de particular interés en este segundo nivel la discusión sobre cómo modelar usando el lenguaje RDF. En esta tesis se propone un conjunto de prácticas y se plantea la necesidad de desarrollar metodologías.

Finalmente, en el tercer nivel, se presenta un modelo (o vocabulario RDF) que ejemplifica las prácticas propuestas sobre un caso de estudio particular, centrado en la publicación de datos de transparencia gubernamental y en datos de historia política chilena.

La tesis concluye con una discusión sobre aquellos aspectos de RDF que lo hacen complejo y que se vislumbran como una barrera para su amplia adopción. Es decir, se plantea la necesidad de revisar la pregunta sobre si las bases de RDF necesitan ser cambiadas.

Tabla de contenido

Índice de cuadros

Índice de figuras

Capítulo 1

Introducción

Durante la última década hemos sido testigos de un aumento en casi un orden de magnitud en la capacidad de generar datos. Este aumento ha cambiado radicalmente el paradigma para comprender el mundo y exige nuevas estrategias para manejar grandes volúmenes de datos [1]. Junto al creciente volumen, también entra al foco de la atención la valorización de los datos. Así, grandes empresas como Google, Facebook o Amazon, hacen de los datos y su capacidad de procesarlos el centro de su negocio. Quienes posean los datos y sean capaces de retener y hacer crecer el caudal entrante de datos (por ejemplo, manteniendo a sus usuarios) tendrán ventajas sobre los demás y "reinarán" durante esta era en la que los datos juegan el rol principal.

La valoración de los datos no sólo ha sido percibida en el ámbito privado, es decir, donde el dato es valioso porque me entrega una ventaja poseerlo sobre quienes no lo poseen, sino que los datos también son valorizados como un bien común, que se manifiesta cuando lo compartimos y cuando la comunidad en general puede sacar provecho de ellos. Esta segunda valorización del dato, vista como un bien común, se manifiesta en las voces que provienen del movimiento de "datos abiertos". Los ciudadanos comienzan a exigir transparencia y algunos gobiernos han levantado portales en los que hacen públicos datos.

El valor público de los datos se manifiesta también en la iniciativa de los datos abiertos enlazados. Si bien los datos enlazados pueden entenderse como una técnica ortogonal a la de los datos abiertos, pues puede ser aplicada también sobre datos cerrados, la combinación con el concepto de datos abiertos es la que le da mayor fuerza. La integración entre distintas fuentes de datos es problablemente una de las características más deseables de los datos

abiertos, aquello que diferencia el bien individual en el dato cerrado del bien colectivo en el dato abierto.

La tecnología de los datos enlazados está construída sobre la Web. Así como la Web surgió en sus inicios para convertirse en un espacio donde todos podrían publicar información, hoy se presenta como el primer candidato a constituirse en el espacio usado por todos para compartir sus datos. Dado el rol central de la Web, es necesario hacer un breve recorrido por sus características principales antes de adentrarse en el modelamiento de datos sobre ella.

La Web de los datos surge como una combinación entre técnicas de modelamiento del conocimiento y los conceptos de la Web. De este modo, los objetos y predicados que conforman los modelos, encuentran una manera de expresarse a través de los identificadores (las URIs), lenguajes (RDF) y protocolos (HTTP) de la Web.

En el caso de Chile, existen varias iniciativas de publicación de datos abiertos y enlazados, tanto dentro de los organismos públicos, como la Biblioteca del Congreso Nacional, como desde la sociedad Civil, como por ejemplo, la Poderopedia.

Junto con estas inciativas, en Chile la Ley Nº 20.285 regula la publicación de la información pública, estableciendo datos que deben ser publicados de manera obligatoria por los organismos públicos (transparencia activa) y un procedimiento por el cual los ciudadanos pueden reclamar información que los organismos no se ven obligados a publicar de manera activa (transparencia pasiva). Además, para resolver aquellos casos en los que un organismo pudiese rehusarse a entregar la información solicitada, se creó el Consejo para la Transparencia.

La obligación de la publicación de datos en el contexto de la transparencia activa ha obligado a la definición de estándares para la publicación. En el caso del gobierno central, éstos se presentan en forma de decretos que tienen como resultado la elaboración de sistemas para la publicación de los datos que, si bien permiten la lectura de la información, aún no hacen posible el procesamiento automatizado ni la integración con otras fuentes de datos.

1.1. Objetivo General

Esta tesis busca facilitar la publicación de datos mediante una propuesta de prácticas que van desde cómo publicar datos en general hasta cómo hacerlo usando las técnicas particulares de los datos enlazados.

1.2. Objetivos específicos

1. Presentar prácticas generales para la publicación de datos abiertos.

2. Presentar prácticas para la publicación de datos enlazados abiertos.

3. Definir un vocabulario para la transparencia activa y otros conjuntos de datos que tienen en común el describir personas, organizaciones y relaciones.

1.3. Metodología

La primera parte de este trabajo consiste en el estudio del estado del arte en la publicación de datos, incluyendo la exploración de los datasets que están siendo publicados, las iniciativas no gubernamentales, los aspectos técnicos de la publicación y las problemáticas a las que ésta se enfrente.

Una segunda etapa consiste en la elaboración de un modelo conceptual para los objetos y los procesos de publicación y consumo de los datos.

En una tercera etapa se describen casos de estudio en la publicación de datos, que servirán de guía para la norma que será desarrollada a continuación. Entre estos casos de estudio se encuentra el de la Biblioteca del Congreso Nacional, la publicación de datos por concepto de transparencia y la publicación de datos por inciativas de provenientes de la sociedad civil tales como la Poderopedia[1].

[1]Poderopedia http://poderopedia.org

En base a los casos de estudio, en una cuarta etapa y final, se desarrollarán las prácticas que formarán parte de la propuesta de la tesis. Estas prácticas estarán divididas en tres grupos. El primer grupo estará orientado a la publicación de datos en general; el segundo, se centrará en la publicación de datos usando las técnicas de datos enlazados; mientras que, el último, presentará vocabularios para cubrir las necesidades centrales de los casos de estudio presentados.

1.4. Resultados obtenidos

Los resultados de esta tesis se pueden agrupar en tres niveles de prácticas para la publicación de datos abiertos. El primero de ellos se conforma por un conjunto de prácticas para la publicación de datos abiertos, que podrían ser consideradas por cualquiera que requiera publicar. El segundo nivel se aborda el problema de cómo publicar datos enlazados y organiza sus prácticas en dos grupos: prácticas de identificación y prácticas de modelamiento. Tal como se presenta con las observaciones hechas al vocabulario de la Biblioteca del Congreso Nacional, definir vocabularios y publicar datos enlazados puede no resultar natural en un principio, por lo que estas buscan guiar a quienes tendrán que realizar esa labor. Por último, se recomienda un vocabulario para su uso en la publicación de datos en ámbitos donde se requiera describir personas, organizaciones y relaciones entre ellas, sin perder información histórica. Este vocabulario es relevante para facilitar la integración de fuentes de datos distintas, como son los datos ublicados por la Transparencia Activa, los datos de la Historia Política y los datos que publicará la Poderopedia.

1.5. Estructura de este documento

El documento de esta tesis se estructura en ocho capítulos:

1. *Introducción:* (este mismo capítulo) describe el problema a resolver, la metodología y la estructura de este documento.

2. *Antecedentes preliminares:* describe el contexto en el que se desarrolla esta tesis, tanto en los aspectos técnicos como en los políticos e históricos.

3. *Modelo:* presenta un modelo conceptual para describir los objetos y procesos en la publicación y consumo de datos.

4. *Casos de estudio:* plantea problemas concretos para los cuales más adelante se elaborarán las prácticas que esta tesis propone.

5. *Prácticas generales:* presenta prácticas para ser aplicadas independientemente de las tecnologías usadas.

6. *Prácticas para datos enlazados:* identifica prácticas específicas para publicar datos en forma de datos enlazados.

7. *Vocabulario propuesto:* presenta vocabulario para ser usado en los casos de estudio.

8. *Conclusiones.*

Capítulo 2

Antecedentes preliminares

Durante la última década hemos sido testigos de un aumento de un orden de magnitud en la capacidad de generar información. Este aumento es crítico además, al considerar que, si bien las capacidades de almacenamiento también han aumentado exponencialmente, parecieran hacerlo a tasas inferiores a aquellas con que se generan los datos. Con la intención de cuantificar ambas capacidades, un estudio realizado por la *International Data Corporation* (IDC) en 2007 [2] estimó que en ese mismo año la capacidad de generación sobrepasaría la de almacenamiento y se iría alejando cada vez más a un ritmo exponencial (ver Figura 2.1). Esta proyección ha sido sobrepasada por las nuevas estimaciones de la IDC elaboradas el 2011, en la que se estimó que en 2010 se crearon 1.227 exabytes, mientras que se proyectó para el 2015 un volumen de 7.910 exabytes. Considerando que el IDC estimó que el 2005 dicho número era de sólo 130 exabytes, podemos concluir que los incrementos han sido muy cercanos al orden de magnitud por cada cinco años.

Estos incrementos exponenciales en la producción de datos han despertado gran interés y generado numerosas opiniones. En el influyente artículo donde introdujo la noción de Web 2.0 [3], O'Reilly recalcó la importancia que cobrarían los datos, afirmando que éstos eran el siguiente "Intel Inside". En el Claremont Report on Database Research [4], algunos de los investigadores internacionales más influyentes en el área de manejo de información centraron su análisis en este fenómeno, mencionando incluso, que la presencia de estos grandes volúmenes de información va a remecer la forma en que se realiza investigación en el área de bases de datos. A ésto se suma el impacto que tiene el uso de grandes volúmenes de datos para la ciencia en general, surgiendo la hipótesis de que nos enfrentamos a "un cuarto paradigma"

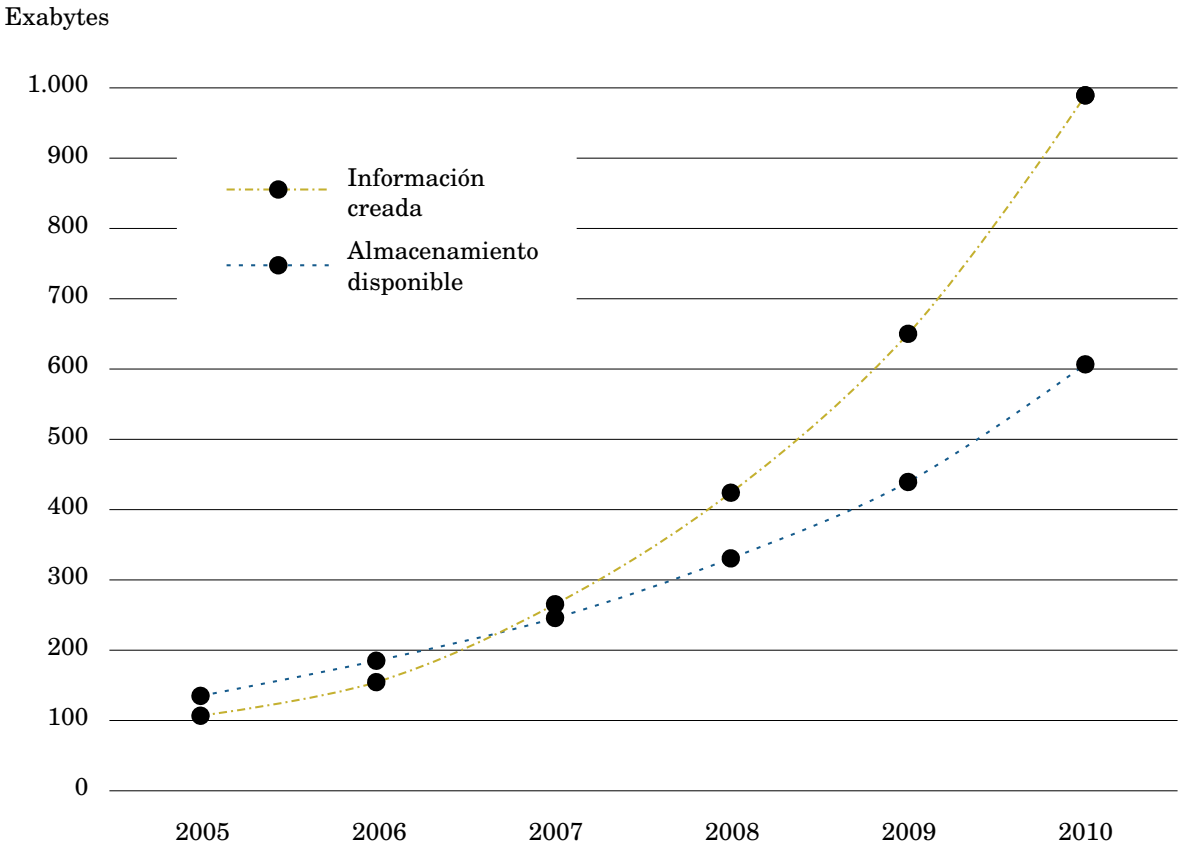


Figura 2.1: Proyección de la IDC hecha el 2007 sobre la creación de información y el almacenamiento disponible. Gráfico de elaboración propia en base a las fuentes [2].

para la investigación científica [1]. Szalay y Gray en su artículo [5] indican que hoy podemos hablar de un "mundo exponencial" por el hecho de que los datos generados y almacenados estén siendo duplicados cada año. Bell, Hey y Szalay hablan en su artículo [6] de un verdadero "diluvio de información". De hecho, ellos indican que algunas áreas científicas almacenan hoy en día, volúmenes de datos miles de veces mayores a los almacenados una década atrás.

Este fenómeno no es exclusivo de las áreas científicas. La misma tendencia ha sido observada en otras áreas, tales como las redes sociales y el manejo de información gubernamental. En el caso particular de las redes sociales, hoy en día pueden encontrarse no sólo enormes volúmenes de información, sino también complejas redes de datos que actualmente sólo son accesibles a investigadores al interior de grandes compañías tales como Google, Yahoo y Facebook y organismos gubernamentales como la *National Security Agency* (NSI) de los Estados Unidos. Aquellos privilegiados investigadores que cuentan con acceso a esos datos producen papers que no pueden ser criticados ni replicados. Esta falta de acceso a los datos, junto con el

conflicto entre la necesidad de acceder a esta información y la necesidad de garantizar la privacidad de las personas, son los mayores impedimentos para desarrollar una ciencia social computacional [7].

2.1. La Web

Desde sus inicios la Web[1] se ha presentado como un espacio de información en el cual todos pueden escribir y leer; la costrucción de un bien común que no estaría bajo el control de ningún grupo reducido. En función de ello, la arquitectura de la Web fue creada con una filosofía que puede resumirse en tres principios básicos: *a*) todos pueden publicar, *b*) todos pueden leer y *c*) nadie debe restringir. Para implementar esta filosofía, la arquitectura de la Web descansa en tres pilares básicos: el protocolo *Hypertext Transfer Protocol* (HTTP) que nos permite acceder a los recursos, el lenguaje *Hiper Text Markup Language* (HTML) con el que codificamos documentos y las *Uniform Resource Identifiers* (URIs) que identifican los documentos globalmente. A las URIs, en su extensión para las diferentes formas de escritura del mundo, se las denomina *Internationalized Resource Identifiers* (IRIs).

2.1.1. Identificadores de la Web

Los identificadores de la Web son una extensión de los identificadores de los nodos de Internet (ver Figura 2.2) y, por ende, el control sobre ellos se define siguiendo la misma estrategia de delegación jerárquica que existe, tanto para la asignación de números como para la creación de dominios. Con la *Internet Corporation for Assigned Names and Numbers* (ICANN) como la raiz del árbol de los dominios, cada sub dominio puede ser administrado por una organización que a su vez puede entregar subdominios a otros. De este modo, en cada rama del árbol habrá un responsable de la administración de los dominios que puede operar de manera independiente a las otras ramas, es decir, gestionando los identificadores de manera distribuida.

En sus inicios los identificadores de la Web eran los *Uniform Document Locators* (UDL), pero como la Web comenzó a almacenar otros objetos aparte de los documentos, estos iden-

[1] La Web fue propuesta incialmente por Tim Berners-Lee y Robert Cailliau mientras trabajaban en la *European Organization for Nuclear Research* (CERN), alrededor de 1989, y hecha pública en 1990 [8].

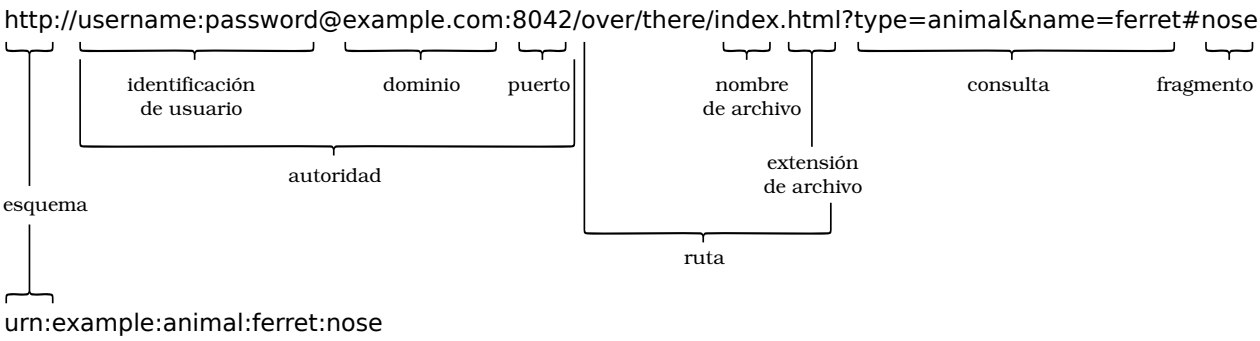


Figura 2.2: La estructura básica de una URL (identificador de arriba) se construye sobre los identificadores de Internet (números IP o nombres de dominio). El resto de la URL especifica el protocolo, las credenciales del usuario, el puerto usado, una cadena que especifica el recurso dentro del dominio. A diferencia de las URL, las URNs (ejemplificadas por el identicador de abajo) no incluyen dominios como parte de su estructura, por lo que no están hechas para ser resueltas directamente mediante los protocolos de Internet. Esta figura es de elaboración propia, usando como base al diagrama presentado en el RFC 3986 [9].

tificadores pasaron a llamarse URLs. Las URIs son una extensión de las URLs, en la que se agregan identificadores que no son dereferenciables, como los *Uniform Resource Names* (URN). Dereferenciar consiste en obtener el recurso referenciado a partir del identificador, tarea que es realizada en la Web mediante el protocolo HTTP.

2.1.2. Protocolos de la Web

HTTP, el protocolo de la Web, fue desarrollado en un trabajo coordinado entre la *Internet Engineering Task Force* (IETF) y el *World Wide Web Consortium* (W3C). La versión actual del protocolo es HTTP/1.1, definido en 1999 en el RFC 2616 [10].

HTTP es un protocolo basado en peticiones y respuestas que se llevan a cabo entre un cliente y un servidor. El cliente envía peticiones de recursos que identifica mediante URIs. Frente a las peticiones, el servidor puede responder, por ejemplo, enviando un documento que se asocia a la URL pedida, enviando una nueva URL para que el navegador repita la petición (redirección) o indicando que tal URL no tiene asociado ningún recurso en el servidor.

Tanto las peticiones como las respuestas se componen de dos secciones: encabezado y cuerpo. El encabezado es una serie de pares clave/valor que contienen los metadatos del mensaje. El cuerpo es usado opcionalmente para enviar un bitstream conteniendo otra información de largo variable, tal como lo son los envíos de documentos o campos de un formulario. En

el caso de las peticiones, el encabezado comienza por una línea que indica el método y la URL del recurso al que se quiere acceder. Los métodos GET, POST, PUT y DELETE, entre otros, sirven para especificar acciones a realizar sobre los recursos asociados a las URI. Por ejemplo, GET se usa para adquirir un recurso, POST para agregarlo al servidor y DELETE para eliminarlo.

Los identificadores y el protocolo de la Web definen un espacio imaginario de recursos que pueden ser publicados y accedidos. Originalmente la Web estaba pensada para el manejo de documentos, es decir, los recursos eran entendidos como los bitstreams codificados como cuerpo de las peticiones y las respuestas del protocolo HTTP. Sin embargo, como hoy es habitual que los servidores respondan con distintos *body* a pesar de recibir la misma petición, la noción de recurso ha terminado por depender de cada aplicación.

2.1.3. Lenguajes de la Web

Con los dos componentes anteriores (URLs y HTTP) hemos creado un espacio de recursos, pero aún nos queda un problema: ¿Cómo podemos acceder a ellos si no conocemos de antemano sus identificadores? Este problema no es menor si consideramos que la publicación es distribuida, lo que impide que podamos conocer todos los recursos a los que podemos acceder en la Web. La tercera componente de la Web, que viene a solucionar este problema, es un lenguaje para codificar recursos, es decir, para codificar los bitstreams que se pueden intercambiar en los cuerpos de los mensajes del protocolo HTTP. La característica principal de este lenguaje, el *Hiper Text Markup Language* (HTML), es la creación de hipervínculos entre recursos usando sus identificadores (URLs). De este modo, la Web adquiere una arquitectura distinta a la de una biblioteca, pues no necesitamos de un catálogo adicional donde se indiquen las posiciones de los libros en sus estanterías, sino que podemos llegar a los recursos navegando a través de los hipervínculos.

Con la posibilidad de establecer hipervínculos entre los recursos, la Web puede imaginarse como un grafo dirigido en la que los nodos corresponden a los recursos y los arcos a las referencias. Este modelo es el usado por el algorítmo *Page Rank* para establecer una métrica de relevancia de los documentos en la Web [11]. Este grafo no resulta "democrático" en el sentido de que ningún contenido ocupe una posición preferente. En efecto, la cantidad de referencias entrantes de los nodos se comporta siguiendo la distribución de Zipf, en la que hay muchas páginas con pocas o ninguna referencia y pocas páginas que acumulan la mayoría

de las referencias [12].

La Web no sólo sería una red desigual en el reparto de enlaces, sino además, se trata de una red en la que las distancias resultan en extremo pequeñas en relación al tamaño total de la red. En 1999 un grupo de investigadores estimó que el diámetro de la Web sería aproximadamente de 19 enlaces [13].

2.1.4. Evolución y principios

Para coordinar la evolución de la Web y garantizar la interoperabilidad (a veces quebrada por algunos navegadores) surgió el Consorcio de la Web (W3C), una organización internacional liderada por Tim Berners-Lee. En una lectura [14] realizada en Japón el año 2004, Tim Berners-Lee desglosa los principios básicos que debería seguir el W3C para dirigir la evolución y la interoperabilidad de la Web:

Independencia de Dispositivo. La misma información debe ser accesible desde diversos dispositivos. Esto significa, por ejemplo, que la visualización debe tener estándares que permitan acceder a la información desde casi cualquier formato de pantalla e interfaces de usuario, tales como podría ser un computador de escritorio o un teléfono celular. Este principio tiene por consecuencia la separación del contenido y la forma, que se ha manifestado en la separación del lenguaje para codificar la información, el HTML, del lenguaje para describir cómo ella debe ser presentada, el CSS.

Independencia de Software. Hay muchas y diversas aplicaciones que forman parte del ecosistema de la Web. En ella, ninguna aplicación debiera ser crítica para su funcionamiento. El desarrollo descentralizado del software es clave para el crecimiento de la Web. Además, tema no menor, la descentralización evita que la Web misma caiga bajo el control de una minoría que controle el desarrollo de dicho software.

Internacionalización. Desde sus inicios, se persiguió que la web fuera neutral frente a las diversas culturas, evitando establecer como estándares prácticas provenientes de algunas culturas en desmedro de las demás. Con la introducción de UNICODE, que se introduce en las nuevas versiones de HTML y en XHTML, se elimina esta la posición preferencial en la que se encontraban los idiomas occidentales. Algo similar ocurre con el otro pilar de la Web, los identificadores, al extender los anteriores Universal Resurce Identifiers

(URIs) a las nuevas IRIs mediante el uso de UNICODE.

Multimedia. Los formatos disponibles para publicar deben estar abiertos a todas las facetas de la creatividad humana. En este sentido, el soportar multimedia no representa sólo un par de avances tecnológicos, sino parte fundamental de su filosofía. De nuevo aquí la independencia del software aparece como un principio para escoger formatos abiertos de multimedia.

Accesibilidad. La gente difiere en múltiples cosas, en particular, en sus capacidades. La universalidad de la Web debe permitir que ella sea usada por la gente independientemente de sus discapacidades. De nuevo aquí la separación de contenido y forma surge como un principio fundamental.

Ritmo y razón. Como dice Tim Berners-Lee, la información varía desde un poema hasta una tabla en una base de datos. El compromiso entre procesamiento automático y humano debe estar presente. Hay que facilitar tanto el acceso a la información a los humanos como a los agentes automáticos, lo que, dado los grandes volúmenes de la información, será fundamental para ayudarnos a procesarla.

Calidad. Muchos sistemas de documentación han sido diseñados para almacenar colecciones de información particulares y se podría asumir que en cada una de ellas se selecciona siguiendo ciertos criterios de calidad. No obstante, estas nociones de calidad son subjetivas y cambian en el tiempo. En la Web la calidad es controlada de manera distribuida, cada cual puede publicar información siguiendo sus propios criterios de calidad. Ello generará la necesidad de herramientas que faciliten el filtrado, combinando opiniones e información desde varias fuentes y bajo el completo control de los usuarios.

Independencia de escala. La Web ha sido diseñada para operar globalmente, no obstante en ella pueden convivir también pequeños grupos. La privacidad de la información de cada grupo debe ser negociada por ellos mismos y permitir que cada grupo se sienta seguro en el control de su espacio. Comenzando en el grupo de una sóla persona la Web debe escalar a grupos de todos los tamaños e intereses, buscando un espacio balanceado de naturaleza fractal que permita a billones de personas convivir pacíficamente.

2.1.5. ¿Ha muerto la Web?

El 17 de agosto de 2010 apareció un artículo escrito por Chris Anderson y Michael Wolff titulado *"The Web is Dead, Long Live the Internet"* [15]. que mostraba un gráfico que ilustraba como los tráficos de red asociados con la Web habían disminuído relativamente (ver figura 2.3). Rob Bechizza respondió con otro artículo titulado *"Is the Web really dead?"* [16] mostrando otro gráfico elaborado usando como fuente los mismos datos (Estimados por CISCO basándose en las publicaciones en CAIDA[2]) en el que se presentaba al tráfico de la Web creciendo exponencialmente (ver figura 2.4). No obstante, el tema central del artículo inicial no era la caída de la Web en cuanto a tráfico, sino en relación a su filosofía.

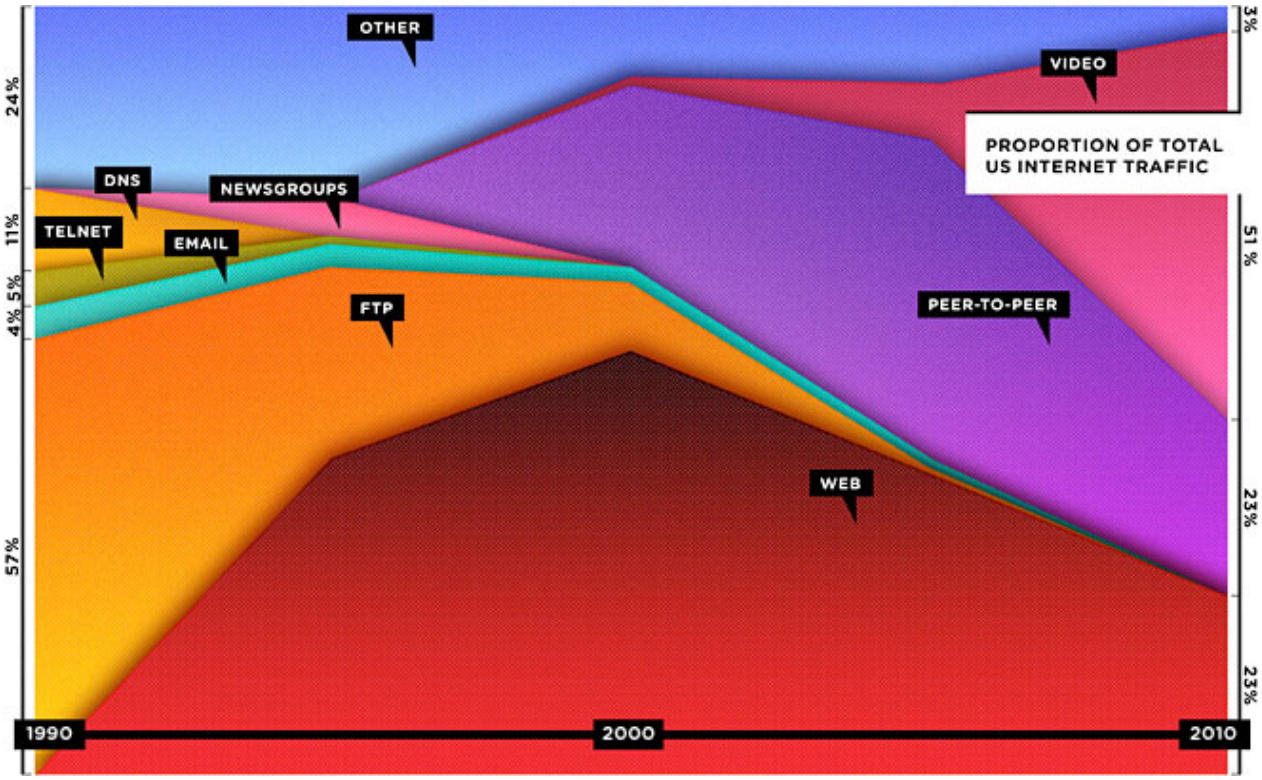


Figura 2.3: Proporción del tráfico a través de Internet presentado en un artículo de Chris Anderson y Michael Wolff [15].

La discusión sobre la muerte de la Web también tuvo su réplica en el debate entre Tim O'Reilly, Chris Anderson y John Battelle [17]. En la visión de Tim O'Reilly la Web sigue el ciclo normal de cualquier industria, que "pasa desde ser abierta, cuando ocurre la innovación, a ser cerrada, cuando el valor es capturado". Esta preocupación por el destino de la Web

[2] Cooperative Association for Internet Data Analysis (CAIDA) es una organización que recoge y analiza datos de Internet. Más información puede encontrarse en su sitio http://www.caida.org

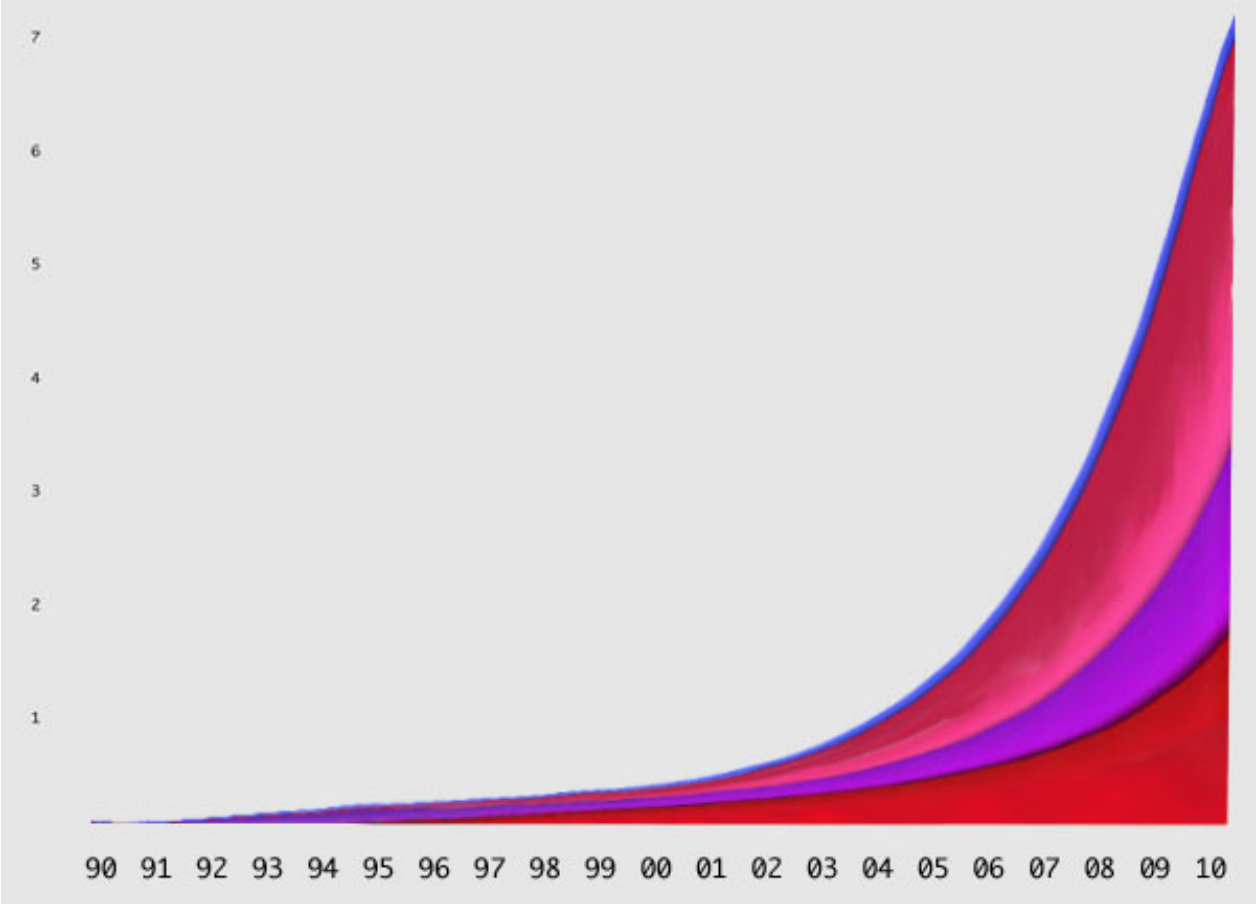


Figura 2.4: Proporción del tráfico a través de Internet presentado en un artículo de Rob Bechizza [16].

es manifestada también por Tim Berners Lee, quien afirma que no ha muerto pero hace un llamado a defenderla [18].

Antes de preguntarnos si la Web realmente ha muerto es necesario discutir qué entendemos por ella. En esta discusión tomaremos el concepto propuesto por Tim Berners-Lee de la Web como una asociación tanto de los tres pilares que la sustentan (HTTP, HTML e URIs) como de los principios que guían su evolución e interoperabilidad.

Actualmente existen varias aplicaciones que son críticas para los usuarios de la Web y que, peor aún, son de propiedad de grupos reducidos de personas. Ello rompe claramente con el principio de *independencia del software*. Aplicaciones en esta línea son, por ejemplo, los buscadores (de Google, Yahoo, Bing) y los servicios monolíticos de redes sociales (Facebook, Google+, Flicker, etc.). Estos servicios están rompiendo además con el segundo principio de Calidad, pues como lo describió Eli Pariser en su charla *Beware online "filter bubbles"* [19], los

usuarios ya no tendrían el control sobre cómo las máquinas estarían filtrando los contenidos. Por último, estos servicios estarían violando también el principio de independencia de escala al impedir que sean los mismos usuarios quienes controlen la manera en que la información de carácter privada pueda ser accedida por otras personas.

2.2. La Web de los datos

La Web fue diseñada para ser leída por humanos a través de los navegadores, aplicaciones de escritorio que actúan como clientes del protocolo HTTP y que son capaces de interpretar y desplegar las páginas Web. De este modo, el lenguaje de la Web, HTML, está pensado para codificar información. Por ello, la necesidad actual de compartir datos ha tenido como consecuencia el replanteamiento de la Web, definiendo una nueva arquitectura para ello: la Web de los Datos.

Una primera aproximación a la publicación de datos en la Web es la definición de subgramáticas de HTML que serían interpretadas en forma de datos semi-estructurados, conocidos también como microformatos[3]. Por ejemplo, para publicar una ubicación geográfica se puede usar el microformato GEO[4]:

```
1 Las aves anidaron en
2 <span class="geo">
3   <span class="latitud">52.48</span>,
4   <span class="longitud">-1.89</span>
5 </span>
```

Aunque los microformatos permiten publicar datos en la Web, no escapan a las dificultades que presentan los datos semi-estructurados para consultar datos desde distintas fuentes y codificados en distintas gramáticas.

[3] http://microformats.org
[4] Microformato GEO http://microformats.org/wiki/geo

2.2.1. El lenguaje RDF

Uno de los requisitos centrales de la Web de los datos es permitir la publicación de manera distribuida utilizando distintos modelos de datos y que, aún así, los datos puedan ser integrados y consumidos de manera transparente. Para lograr esto el W3C propuso el *Resource Description Framework* (RDF), inicialmente un lenguaje especificado en función de su sintaxis [20] y que luego se separa de la serialización para ser entendido a través del modelo detrás de el: las bases de datos deductivas, en el cual los datos son conjuntos de afirmaciones. De este modo, se resuelve parte del problema de integración gracias a que en muchos casos basta unir los conjuntos de afirmaciones provenientes de dos fuentes de datos.

Las bases de datos deductivas están formadas por aseveraciones que pueden tener distintas cantidades de símbolos. El modelo de RDF escoge un largo uniforme de tres componentes por cada afirmación: un sujeto, un predicado y un objeto. De este modo se asemeja al lenguaje humano, permitiendo expresar la afirmación "las aves anidaron en 52.48, -1.89" mediante el sujeto "las aves", el predicado "anidaron en" y el objeto "52.48, -1.89". A cada una de estas afirmaciones se las conoce también como tripleta y a las bases de datos de tripletas se las llama *triple stores*.

A diferencia del lenguaje natural, en el cual a partir del discurso podemos entender a qué nos referimos por "las aves", en RDF no hay un orden en las afirmaciones y, además, está pensado con el sentido de que las frases sean interpretadas independientemente. Para ello resulta fundamental el uso de identificadores para los recursos, por lo que lo común es que las componentes de las tripletas sean identificadores (URIs), a excepción de la última componente, donde suelen ir también literales (valores codificados como cadenas de caracteres). De este modo, si volvemos al ejemplo anterior, podremos identificar al sujeto "las aves" por un identificador preciso para la pareja de aves, `http://../aves/23`, y además el predicado "anidaron en" por otra URI `http://../anidaronEn`. Así, la oración inicial quedará expresada por la tripleta presentada en el Cuadro 2.1.

Sujeto	*Predicado*	*Objeto*
`<http://../aves/23>`	`<http://../anidaronEn>`	`"52.48, -1.89"`

Cuadro 2.1: Tripleta RDF que expresa la afirmación "La pareja de aves 23 anidó en las coordenadas 52.48, -1.89". Esta afirmación podría por ejemplo pertenecer a un estudio de anidación de aves que estudie un conjunto de parejas.

Notación

El triple anterior introduce una notación para distinguir entre URIs, encerradas por paréntesis triangular (<>), y literales, encerrados entre comillas (""). El cuadro 2.2 muestra varias de las notaciones utilizadas en la sintaxis de N3 para caracterizar los nodos [21, 22].

Notación	Uso
<x>	x es una URI.
<#x>	una URI definida usando como prefijo la URI del documento actual y x como el fragmento.
u:y	es una CURIE, es decir, una URI definida de manera compacta donde u es un string *corto* que representa (y reemplaza en la notación) a una URI x y x:y representa a la URI formada por la concatenación de x con y.
_:y	es una CURIE que define un nodo blanco, es decir, un identificador local que no puede ser referenciado desde otro documento.
"a"	a es un literal.
"a"^^t	a es un literal de tipo t.
"a"@l	a es un literal en idioma l.

Cuadro 2.2: Notaciones de nodos en N3 [21, 22].

De entre estas notaciones, quizá la de los nodos blancos resulte la menos intuitiva. ¿Para qué necesitamos tener identificadores locales si contamos con identificadores globales? Para entender los nodos blancos volvamos a mirar el ejemplo de la ubicación de las aves. Si hubiéramos querido seguir el patrón del microformato GEO tendríamos que haber separado las dos componentes, por ejemplo, usando las tripletas del cuadro 2.3.

Sujeto	*Predicado*	*Objeto*
`aves:23`	`voc:anidaronEnLatitud`	`"52.48"^^xsd:decimal`
`aves:23`	`voc:anidaronEnLongitud`	`"-1.89"^^xsd:decimal`

Cuadro 2.3: Conjunto de dos tripletas que capturan la misma información que de la tripleta del cuadro 2.1. A diferencia del primer conjunto de tripletas, en este segundo se separan las coordenadas de la información geográfica, permitiendo que éstas puedan expresarse dentro de un tipo de datos conocido (`xsd:decimal`). Además, se usan CURIEs para acortar las notaciones.

Para uniformizar la notación de las coordenadas geográficas podemos usar un identificador

para el lugar, de este modo, independizamos la descripción de las coordenadas de la descripción del objeto localizado (ver cuadro 2.4).

Sujeto	*Predicado*	*Objeto*
aves:23	voc:anidaronEn	_:b
_:b	geo:latitud	"52.48"^^xsd:decimal
_:b	geo:longitud	"-1.89"^^xsd:decimal

Cuadro 2.4: Conjunto de dos tripletas que capturan la misma información que los cuadros 2.1 y 2.3. A diferencia de éstos, en este cuadro se crea el nodo blanco `_:b`, que permite agrupar toda la información relativa a la localización.

El uso de un elemento extra como `_:b` no justifica que éste sea un nodo blanco. La razón práctica de los nodos blancos está en gran medida asociada a lenguajes como N3, que permiten definir recursos sin indicar sus URIs. Por ejemplo, los datos del cuadro 2.4 pudieron haber sido codificados con el siguiente código N3:

```
aves:23 voc:anidaronEn [
    geo:latitud  "52.48"^^xsd:decimal ;
    geo:longitud "-1.89"^^xsd:decimal
]
```

De este modo, la creación de nodos blancos tiene sus orígenes en el uso de sintaxis que hacen uso de lenguajes que no exigen la identificación de todos los nodos, como por ejemplo, para compactar la notación de funciones n-arias y listas. Además, nos permiten crear identificadores que pueden ser usados localmente sin comprometernos a mantener su dereferenciabilidad.

2.2.2. Grafos RDF

La elección de la tripleta como unidad básica en la descripción de recursos dentro de RDF, no sólo es por simplicidad, sino también porque los datos pueden ser entendidos como digrafos rotulados en los que cada tripleta es vista como un arco que va desde el sujeto al objeto. Por ejemplo, la figura 2.5 representa los datos del cuadro 2.4.

El uso de modelos con estructura de grafos no exclusivo del lenguaje RDF, también está presente en otros desarrollos, como por ejemplo el *Entity Attribute Value* EAV/CR [23] usado principalmente para registros clínicos y el modelo de datos de Freebase [24]. Una descripción

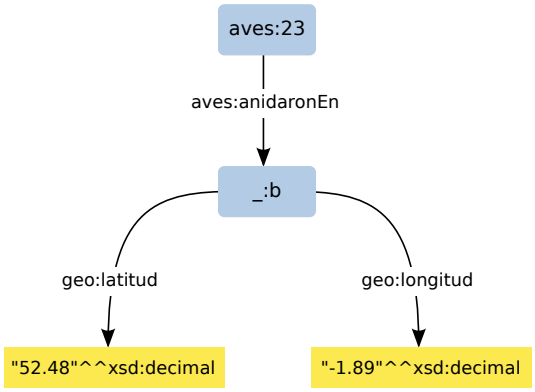


Figura 2.5: Grafo RDF que representa los datos del cuadro 2.4. Cada uno de los tres arcos representa a una de sus tripletas.

de los modelos de bases de datos de grafos es presentado en [25]. Además, el uso de identificadores globales y grafos para codificar datos en la Web no sólo se restringe a las propuestas del W3C. La organización OASIS propuso, ya por 2006, un modelo de grafos llamado *XRI Data Interchange* (XDI) [26], basado en una extensión de las URIs llamada *eXtensible Resource Identifier* (XRI) y un formato de archivos conocido como X3. Junto con XDI se propone la extensión de la Web a la llamada Dataweb, que se basa en la creación de enlaces de confianza entre los actores de la red (o *XDI Trust Information links* (XTI)) para el intercambio de datos [27]. El establecimiento de estos enlaces y la creación de identificadores de manera distribuida hacen que la red basada en XDI sea especialmente aplicable en el contexto de generar una red social global. No obstante, el modelo XDI está lejos de lograr la popularidad de RDF y su desarrollo se encuentra congelado.

2.2.3. Vocabularios

Al igual que con las palabras del lenguaje natural, a las URIs de las afirmaciones hechas en RDF las llamamos vocabulario. En la figura 2.5 hay URIs que están pensadas para ser utilizadas en más de un conjunto de datos. Por ejemplo, es posible que los términos `geo:latitud` y `geo:longitud` sean usados en cualquier conjunto de datos que requiera localizar puntos en el globo terrestre.

La integración es uno de los principales desafíos que impone la publicación de datos en la Web. En general, la integración de datos consiste en proveer a los consumidores (o usuarios) una interfaz común para acceder datos dispersos y de manera heterogénea [28]. Siguiendo el ejemplo de la anidación de aves, podemos suponer que en el mundo se hacen distintos estudios

de ellas, y que además, podrían ser contrastados con información de las poblaciones de otras especies o con información climática. Algunos investigadores podrían querer consultar toda esta información desde la Web, tal como si proveniese de una misma fuente y usase un vocabulario común. Este es el desafío de la integración.

Los vocabularios de la Web de los datos son creados de manera distribuida, y al igual que con las palabras, se convierten en estándares en la medida en que se hacen populares. Sin usar los mismos vocabularios llegamos a una situación similar a la del mito de la Torre de Babel. Por ello una recomendación recurrente es no crear vocabularios nuevos sin antes buscar entre los ya existentes. No obstante, ésto no siempre es posible, pues muchas veces los que hay no se adaptan a nuestros datos. Existe una gran barrera a la integración por medio del uso de vocabularios comunes: la ausencia de un modelo universal de la información. Angelo Taivalsaari aborda este problema en [29]:

> Un ejemplo de un concepto que es difícil de describir en términos de propiedades compartidas es el de "obra de arte". ya que nadie puede definir límites claros para decidir qué es arte y qué no lo es, no hay una clase general "obra de arte" que comparta propiedades comunes. La definición es subjetiva y depende en gran medida de la situación o del punto de vista.
>
> Algunas personas que viven cerca del Ecuador no pueden distinguir entre el hielo y la nieve, mientras que los esquimales poseen numerosas palabras para distinguir entre distintos tipos de nieve. Los Dani, de Nueva Guinea, tienen sólo dos términos de colores básicos: mili (oscuro/frío) y mola (luminoso/cálido) que cubren todo el espectro y tienen gran dificultad para diferenciar colores con mayor detalle.

2.2.4. Deducción

Las bases de datos deductivas son aquellas que proveen reglas que permiten obtener datos a partir de otros. Por ejemplo, la regla

$$\forall x \forall y \forall z : \frac{x \texttt{ padre } y \ \wedge \ y \texttt{ padre } z}{x \texttt{ abuelo } z}$$

nos permite deducir una tripleta a partir de dos tripletas dadas, cada vez que sea posible instanciar las variables x, y, z. De este modo la deducción nos lleva a definir el conocimiento explícito K, compuesto de todas las tripletas que contiene un *triple store* y el conjunto de

datos implicitos K^* compuesto por todas las tripletas que es posible generar aplicando reglas de deducción. Esta diferenciación también se extiende entre los lenguajes de consulta que limitan su espacio de búsqueda a K de aquellos que se extienden a K^*.

2.2.5. Esquemas y ontologías

Los esquemas y las ontologías corresponden a vocabularios en los que se han introducido reglas deductivas para explicitar su semántica. Para introducir estas reglas lo que se hace es definir un lenguaje (generalmente también en la forma de un vocabulario RDF) al que se han incorporado dichas reglas. Enmarcados en este usual compromiso entre expresividad y complejidad de procesamiento se han desarrollado varios lenguajes que pueden ser agrupados, a grueso modo, en tres grupos: *a*) aquellos con una mínima semántica o sin ella (especialmente para definir jerarquías de tipos, clases y predicados) [30], *b*) *RDF Schema* más algunas extensiones menores y *c*), OWL, el lenguaje para definir ontologías en la Web.

Si bien los lenguajes de especificación de vocabularios son útiles como una especificación de los contenidos de los grafos publicados, en la práctica las reglas de deducción actualmente tienen poco uso en el proceso de integración. Esto debido a que el lenguaje de consulta más usado para RDF, SPARQL en su versión de 2008 [31], no soporta el uso de reglas de deducción. Sin embargo existen otros lenguajes de consulta que sí utilizan la semántica como son el caso del proyecto 4sr [32] que utiliza la semántica minimal para RDFS propuesta en [30] para extender SPARQL, el lenguaje de consulta SeRQL implementado dentro del conjunto de herramientas Sesame [33] y el uso de Datalog como herramienta de consulta de RDF [34, 35].

2.2.6. Formatos para RDF

La figura 2.6 presenta algunos de los formatos para codificar grafos RDF y las relaciones de inclusión entre las funcionalidades que proveen. Este diagrama no es exhaustivo; varios formatos han surgido desde entonces para satisfacer distintas necesidades. En esta tesis no ahondaremos en los formatos de codificación de RDF y la gran mayoría de los ejemplos serán expresados a través de diagramas de grafos.

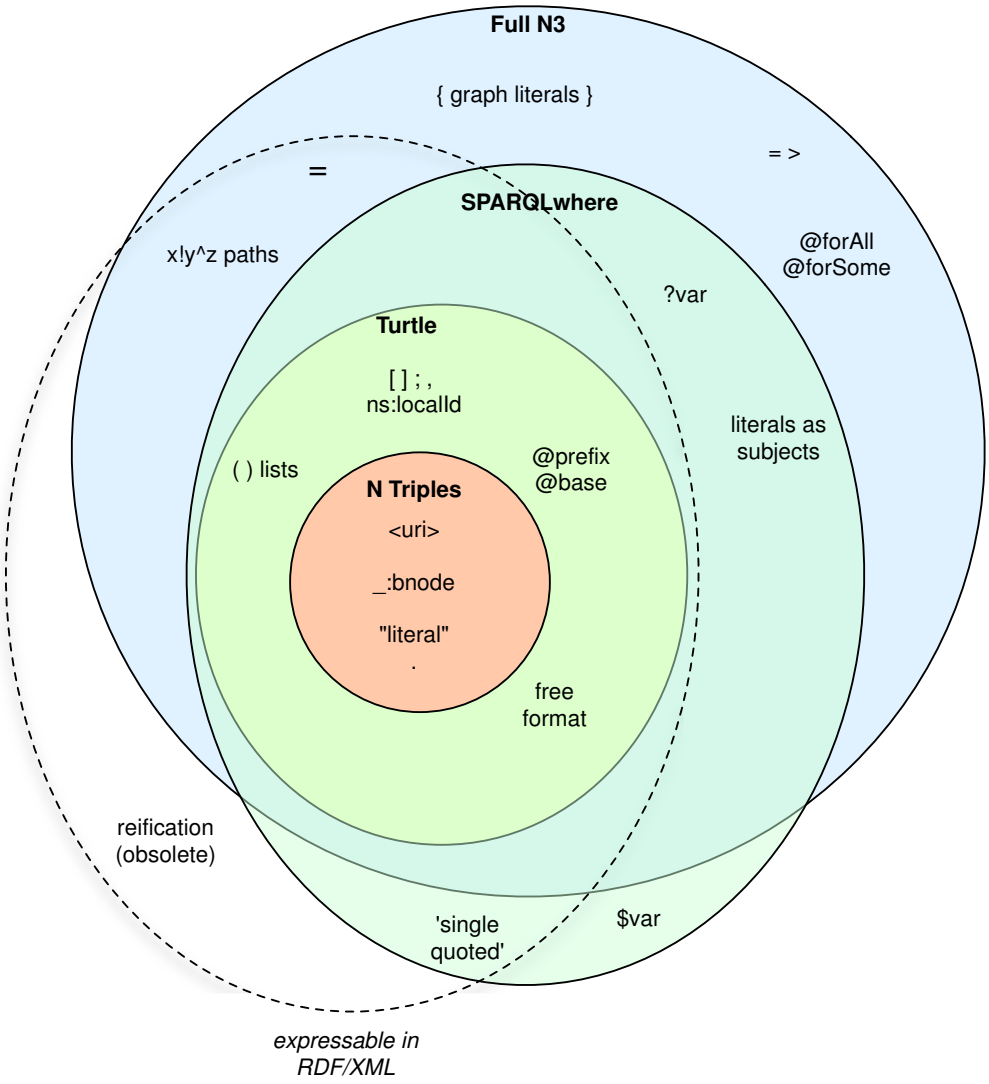


Figura 2.6: Relaciones entre algunas sintaxis para expresar grafos RDF. Este diagrama es parte de la documentación del W3C en Design Issues y puede descargarse de http://www.w3.org/DesignIssues/diagrams/n3/venn.svg

2.2.7. Linked Data

Una de las principales iniciativas de publicación de datos en la Web es el proyecto *Linked Data*, que busca hacer de la Web un repositorio global donde los datos podrían ser publicados y consultados [36]. El proyecto se originó en las ideas de Tim Berners-Lee sobre la arquitectura de la Web [37]. La adopción del modelo RDF fue natural dada la multiplicidad de las fuentes y la importancia de su integración. El movimiento se autodefine de la siguiente manera[5]:

[5] Definición publicada en http://www.linkeddata.org.

> *Linked Data* consiste en el uso de la Web para conectar datos que no estaban enlazados, o usar la Web para facilitar el proceso de conectar datos que ya estaban enlazados con otros métodos. Más específicamente, la wikipedia define *Linked Data* como "un término usado para describir las mejores prácticas para publicar, compartir y conectar datos, información y conocimiento de la Web Semántica, usando URIs y RDF.

La Web Semántica es aquella en la que el conocimiento estaría expresado mediante sentencias que pudieran ser procesadas por máquinas, usando RDF como base para ello. El movimiento Linked Data usa como base la Web Semántica, pero pone su énfasis en interralcionar los datos publicados, dejando para más adelante los aspectos más semánticos.

Como estrategia motivacional el proyecto Linked Data publica un grafo en el que se presentan los más importantes *datasets* que utilizan las prácticas propuestas (ver figura 2.7).

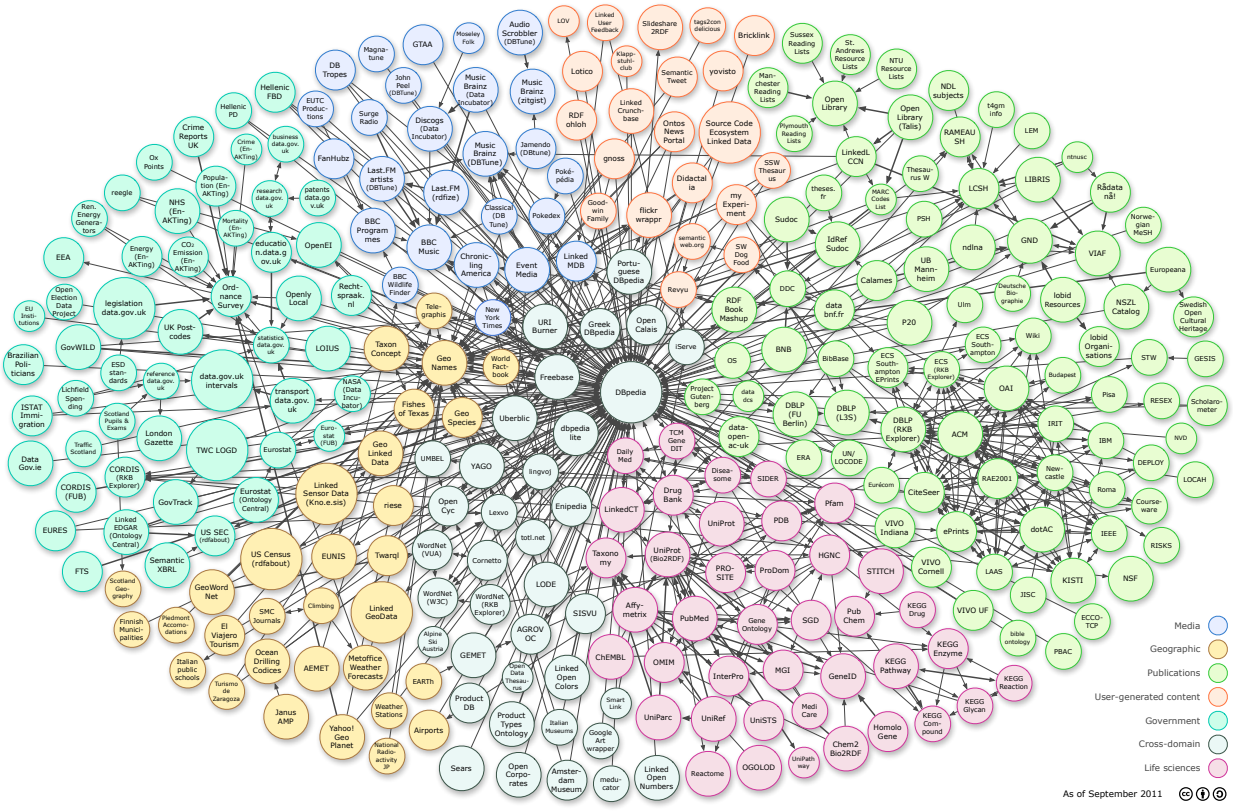


Figura 2.7: Diagrama de la nube de Linked Open Data en su versión de septiembre de 2011. Este diagrama fue elaborado por Richard Cyganiak y Anja Jentzsch y se encuentra disponible en http://lod-cloud.net. Cada nodo representa un *dataset*, es decir, un conjunto de datos, y los arcos representan la existencia de referencias entre ellos.

Para que un dataset sea incluido en el grafo de la figura 2.7 debe satisfacer las siguientes condiciones:

1. Deben haber URIs resolubles en el protocolo HTTP o HTTPS asociados al dataset.

2. Dichas URIs deben ser resueltas, con o sin negociación de contenidos, respondiendo con datos RDF en alguno de sus formatos más populares (RDFa, RDF/XML, Turtle, N-Triples).

3. El dataset debe contener al menos 1.000 tripletas.

4. El dataset debe estar conectado vía enlaces RDF con otro dataset que pertenezca previamente al diagrama. Esto significa que el dataset debe utilizar URIs del otro dataset o viceversa. Arbitrariamente se requieren al menos 50 enlaces.

5. Se debe entregar acceso al dataset completo mediante crawling RDF, un dump de las tripletas RDF o un SPARQL endpoint.

2.3. Open Data

A diferencia del movimiento de datos enlazados, el de los datos abiertos busca la apertura de los datos, independientemente de las tecnologías usadas, mientras éstas sean también abiertas. En esta sección se presentarán los principios de los datos abiertos haciendo un recorrido por las distintas visiones aportadas por diversas personas y agrupaciones surgidas en función de fomentar la apertura de los datos.

2.3.1. Software Libre

El movimiento del Software Libre fue uno de los primeros en desarrollar el concepto de libertad en relación a objetos digitales. Fundado por Richard Stallman, buscaba que los programas de computadora se desarrollen de manera abierta. Richard Stallman distingue el software libre como aquel que posee cuatro libertades [38]:

L_0: La libertad de usar el programa para cualquier propósito.

L_1: La libertad de estudiar cómo el programa funciona y de hacer modificaciones sin restricciones. Esta libertad implica el acceso al código fuente.

L_2: La libertad de redistribuir copias a otras personas.

L_3: La libertad de redistribuir copias que incluyan las modificaciones que se le hayan hecho.

Tanto el movimiento del software libre como el movimiento de los datos abiertos han desafiado el modelo de negocios establecido para la comercialización de los objetos físicos, situación que se ha repetido en otros contextos, como el descrito por Larry Lessig en torno a los derechos sobre la cultura en la era digital [39], y que ha dado origen a movimientos y organizaciones como el de Creative Commons[6], organización que promueve un marco de licencias que permita compartir las obras licenciadas.

2.3.2. Código abierto

Un grupo de integrantes del movimiento del Software Libre se desprendió de éste definiendo un nuevo concepto, el de Código Abierto, y creó una nueva organización, la *Open Source Initiativa* (OSI), fundada por Eric S. Raymond y Bruce Perens. La OSI define el *Open Source* como aquel que satisfaga los siguientes principios[7]:

1. *Libre redistribución:* el software debe poder ser regalado o vendido libremente.

2. *Código fuente:* el código fuente debe estar incluido u obtenerse libremente.

3. *Trabajos derivados:* la redistribución de modificaciones debe estar permitida.

4. *Integridad del código fuente del autor:* las licencias pueden requerir que las modificaciones sean redistribuidas sólo como parches.

5. *Sin discriminación de personas o grupos:* nadie puede dejarse fuera.

6. *Sin discriminación de áreas de iniciativa:* los usuarios comerciales no pueden ser excluidos.

[6]Creative Commons http://creativecommons.org
[7]Principios del Código Abierto http://opensource.org/docs/osd

7. *Distribución de la licencia:* deben aplicarse los mismos derechos a todo el que reciba el programa

8. *La licencia no debe ser específica de un producto:* el programa no puede licenciarse solo como parte de una distribución mayor.

9. *La licencia no debe restringir otro software:* la licencia no puede obligar a que algún otro software que sea distribuido con el software abierto deba también ser de código abierto.

10. *La licencia debe ser tecnológicamente neutral:* no debe requerirse la aceptación de la licencia por medio de un acceso por clic de ratón o de otra forma específica del medio de soporte del software.

No obstante la cercanía entre los principios del Software Libre y del Open Source, la *Free Software Foundation* (FSF) se aleja de la OSI por considerar que ésta ve sólo el beneficio cualitativo para la producción del software y no el fin ético de que el software sea libre. Este mismo tipo de discusiones aparece en torno a la apertura de los datos, entre quienes lo ven como una necesidad para la democracia y el empoderamiento de la ciudadanía y quienes lo ven como la creación de un ecosistema sobre el cual las empresas podrán montar nuevos servicios.

2.3.3. Datos abiertos de gobiernos

Una parte fundamental de los datos abiertos consiste en aquellos que son producidos por organismos públicos, tambien llamados *Open Goverment Data* (OGD). Luego de la creación de los repositorios de datos de los gobiernos de Estados Unidos[8] y Reino Unido [9], varios gobiernos de otros países siguieron el ejemplo. La fundación CTIC mantiene un catálogo[10] en el que ya se registran más de 100 catálogos de datos. Chile no se ha quedado atrás y también cuenta con sus catálogos de datos publicados por el Gobierno [11] y por la Biblioteca del Congreso Nacional [12].

[8] http://data.gob
[9] data.gov.uk
[10] http://datos.fundacionctic.org/sandbox/catalog/faceted
[11] http://datos.gob.cl
[12] http://datos.bcn.cl

Una de las principales definiciones de qué son los datos abiertos de gobierno fue desarrollada por treinta defensores del gobierno abierto para redactar sus ocho principios [40]:

1. *Completos:* Todos los datos públicos deben estar disponibles. Los datos públicos son aquellos que no están sujetos a restricciones de privacidad, seguridad o privilegios.

2. *Primarios:* Los datos deben ser entregados tal como fueron recolectados de su fuente, con la mayor granularidad posible, sin ninguna agregación o modificación en su forma.

3. *Oportunos:* Los datos deben estar disponibles tan pronto como sea posible con el fin de preservar su valor.

4. *Accesibles:* Los datos deben estar disponibles para el más amplio rango de usuarios y para el más amplio rango de propósitos.

5. *Procesables automáticamente:* Los datos deben estar razonablemente estructurados para permitir el procesamiento automático.

6. *No discriminatorios:* Los datos deben estar disponibles para todos, sin requerir registro.

7. *No propietarios:* Los datos deben estar disponibles en un formato sobre el cual ninguna entidad tenga un control exclusivo.

8. *Libres de licencias:* Los datos no deben estar sujetos a ninguna restricción producto de derechos de autor, patentes, derechos de marca o secreto industrial. Restricciones razonables sobre privacidad, seguridad y privilegios deberían ser consideradas.

Con estos principios no se pretende que todos los datos sean abiertos, sino definir cuáles datos pueden ser considerados datos abiertos de gobierno.

La definición de estos principios se hizo en el marco de una iniciativa organizada por Tim O'Reilly y Carl Malamud, una de las varias iniciativas entorno al OGD[13].

[13]Ver su documentación en en el sitio http://www.opengovdata.org/

2.3.4. Metadata en datos estadísticos

La Open Data Foundation (ODF)[14] es una organización dedicada a la adopción de estándares de metadatos y al desarrollo de soluciones open-source para promover el acceso y uso de datos estadísticos provenientes de áreas diversas. Para ello la ODF crea la siguiente desiderata sobre la publicación:

La publicación debe ayudar:

1. a descubrir la existencia de fuentes de datos;
2. a accesar los datos para análisis e investigación;
3. a encontrar información detallada que describa los datos y su proceso de producción;
4. a accesar las fuentes de datos e instrumentos de recolección, de las cuales y con los cuales los datos fueron recolectados, compilados y agregados;
5. a comunicarse efectivamente con los organismos involucrados en la producción, almacenamiento y distribución de los datos;
6. a compartir conocimiento con otros usuarios.

2.4. Privacidad

Existe un compromiso entre la apertura de datos y la privacidad que es resumido por David Lazer y otros investigadores mediante la siguiente frase: *Probablemente el mayor desafío en el campo de los datos es aquel que surge en la relación entre acceso y privacidad* [7]. Hoy resultan bastante aceptadas las necesidades de apertura de la información y de privacidad de las personas. Sin embargo, el problema surge cuando ambas necesidades entran en pugna, es decir, cuando ciertos datos podrían no querer publicarse por motivos de privacidad.

Para describir este conflicto primero es necesario detallar qué se entiende por privacidad. La privacidad es probablemente uno de los conceptos más complicados, malentendidos y deba-

[14] Open Data Foundation www.opendatafoundation.org

tidos en ciencias sociales, en ámbitos legales, filosóficos y tecnológicos, durante las últimas décadas en el mundo [41]. Por ello existen múltiples definiciones del concepto de privacidad [42]:

1. El derecho a ser dejado tranquilo o solo [43].

2. La habilidad para limitar físicamente, vía interacción, sicológicamente e informacionalmente el acceso a mi individualidad o la individualidad de un grupo (citado en [44]).

3. La libertad o carencia de limitaciones irrazonables sobre la construcción de mi propia identidad [45].

4. La necesidad de los individuos, grupos o instituciones de determinar por ellos mismos, cuándo, cómo y hasta qué nivel de información acerca de ellos es comunicada a otros [46].

5. La capacidad de una persona de compartamentizar su vida social, de manera que la información sobre ella que pudiera ser dañina o vergonzosa se mantenga protegida, circunscrita a las reglas del contexto donde se generó [47, 48].

Es necesario recalcar que la última de las definiciones de privacidad, aquella que la define como la preservación de la integridad del contexto, elimina la línea que divide los que pueden considerarse datos sensibles de datos no sensibles. Sin embargo, esta línea aparece en las legislación chilena, en la actual Ley 19.628 sobre la protección de la vida privada. Allí, en las definiciones se diferencia entre datos personales y datos sensibles, siendo los últimos definidos como:

> Aquellos datos personales que se refieren a las características físicas o morales de las personas o a hechos o circunstancias de su vida privada o intimidad, tales como los hábitos personales, el origen racial, las ideologías y opiniones políticas, las creencias o convicciones religiosas, los estados de salud físicos o psíquicos y la vida sexual.

La Ley 19.628 define, además, los datos personales, como aquellos "relativos a cualquier información concerniente a personas naturales, identificadas o identificables". Sin embargo, ello nos lleva a dos difíciles dilemas, el de distinguir cuándo un dato refiere a una persona y el de entender cuando una persona está identificada o resulta identificable.

La dificultad de entender cuándo un dato es de carácter personal surge frente a las posibilidades de deducción que ofrecen los datos. Por ejemplo, los datos estadísticos son típicamente considerados como no personales. Decir que el 10 % de una población de 10 millones de habitantes padece una determinada enfermedad suele ser considerado información no personal, en cambio, la lista de 10 personas que padecen una enfermedad es considerado información personal. El problema de distinguir si ciertos datos son personales o no, se da en aquellos casos límite, como por ejemplo, decir que el 90 % de una población de 100 habitantes padece una enfermedad. Resulta que la ley no considera ni graduaciones de certeza ni la posibilidad de deducir información de carácter personal a partir de información catalogada como no personal.

Otro gran problema de la Ley es el concepto de identificabilidad de las personas, pues si bien ciertos sistemas pudieran no llegar a conocer nuestros nombres reales o nuestro número de la célula de identidad (RUN), pueden identificarnos dentro de sus sistemas utilizando las cookies que insertan en nuestro navegador o según nuestros patrones de comportamiento.

La Ley define a la anonimización como un proceso mediante el cual se evitaría la posibilidad de identificar a las personas mencionadas en un conjunto de datos. Sin embargo, a pesar de que la anonimización en la práctica es entendida como el ocultamiento de los RUN y los nombres de las personas, varios casos han demostrado que tal mecanismo está lejos de ser eficaz [49, 50, 51]. En consecuencia, existen estrategias de anonimización más complejas como la K-anonimización [52, 53, 54] y la disminución de certeza sobre patrones considerados como sensibles [55].

Si aceptamos la privacidad definida como integridad del contexto, nos encontraremos con que todas las empresas que registran nuestras actividades en internet están violando nuestra integridad, pues nosotros entregamos nuestros datos en un contexto distinto al de sus posteriores usos. Lo mismo ocurre con aquellos estudios que utilizan datos tomados desde twitter u otras redes sociales. “No porque estos datos sean públicamente asequibles significa que fueron pensados para ser consumidos por cualquiera” [56]. Al respecto, Boyd [57] sugiere los siguientes principios éticos:

1. Seguridad a través de la oscuridad es una estrategia razonable. Esto implica que quienes publican sus datos aún sin protección no esperan que éstos estén públicos y que puedan tener cualquier uso.

2. No todos los datos fueron hechos públicos para que sean publicitados.

3. Quienes publican información públicamente identificable no necesariamente rechazan su privacidad.

4. Agregar y distribuir datos fuera de contexto es una violación de privacidad.

5. La privacidad no es control de acceso.

En general los principios de privacidad son planteados en el sentido de que toda persona tiene igual derecho a ella. Sin embargo, hay quienes adoptan una postura diferente. Por ejemplo, Wikileaks se guía por cinco principios que hacen una diferencia entre los derechos a privacidad y transparencia entre débiles y poderosos[15]:

1. El derecho a comunicarse, incluyendo el derecho a hablar y el derecho a ser escuchados, la libertad de pensamiento, el derecho a comunicarse con privacidad y el derecho a comunicarse de forma anónima.

2. La inviolabilidad de la historia.

3. Privacidad para los vulnerables.

4. Transparencia para los poderosos.

5. La búsqueda de la verdad como pre-requisito primario para una civilización más justa.

Principios de este mismo tipo son los que motivan también a proyectos como el de la Poderopedia.

2.5. Repositorios de archivos y catálogos

En la actualidad la visión frente al problema de acceso a los datos, sostenida por los principios de datos abiertos presentados en la sección 2.3, se encuentra centrada en los documentos. Es decir, la gestión de datos sería una extensión de la gestión de información, en la cual los documentos de archivo pasarían a contener datos accesibles mediante procesos automatizados. La modalidad de acceso a los datos sería simplemente la descarga de dichos documentos de

[15]Los ideales de Wikileaks se pueden encontrar en https://wlfriends.org/about

archivo. Un caso que se sale de esta regla es el de los datos enlazados donde se promueve una estrategia distinta, la de brindar servicios de consulta a SPARQL endpoints (ver sección 2.2.7). El ámbito de los datos científicos es otro ámbito que se sale de la regla, pues para entregar acceso a grandes volúmenes de datos muchas veces también es necesario entregar facilidades de cómputo de manera distribuida. Este paradigma es descrito por Jim Gray quien recomienda el desarrollo de arquitecturas *scale out* que realizen los cómputos de la manera más cercana posible a los datos (ver más detalles en la sección 2.6).

La norma ISO 15489-1 define "sistemas de gestión documental" o "sistemas de documentos de archivo" como un "sistema que incorpora, gestiona y facilita el acceso a los documentos de archivo a lo largo del tiempo". A su vez, define los "documentos de archivo" como información creada o recibida, conservada como la información y prueba, por una organización o individuo con el desarrollo de sus actividades o en virtud de sus obligaciones legales" y al "documento" como la "información u objeto registrado que puede ser tratado como una unidad".

Llevando las definiciones anteriores al ámbito de los documentos digitales, un documento de archivo corresponde a una secuencia de bytes y, el sistema de gestión de documentos de archivo como un sistema que permite almacenar y recuperar documentos de archivo íntegramente. La integridad implica que la secuencia de bytes del archivo recuperado sea la misma que la del archivo que se había almacenado.

Junto con la capacidad de almacenar y recuperar documentos, los sistemas de gestión de archivos poseen otras funcionalidades tales como la gestión de los permisos de acceso de los usuarios, la gestión de metadatos para los documentos almacenados y la posibilidad de buscar documentos usando la metadata o los contenidos de ellos. En muchos casos estos sistemas también permiten realizar búsquedas sobre documentos que no se encuentran en ellos, sino en sistemas que se han federado y sobre los que se intercambia la metadata mediante procesos conocidos como *metadata harvesting*, entre los cuales destaca el protocolo Open Archives Initiative (OAI) [58].

R.W. Moore, A. Rajasekar y M. Wan proponen la siguiente clasificación de los sistemas de manejo de datos científicos [59]:

- **Colección distribuida de datos.** En élla los datos se encuentran distribuidos físicamente, pero descritos usando un único espacio de nombres.

- **Grid de datos.** Corresponde a la integración de múltiples colecciones de datos, cada

una utilizando un espacio de nombres separado.

- **Biblioteca digital federada.** Es una colección de datos distribuida o un grid de datos que provee servicios para la manipulación, presentación y búsqueda de objetos digitales.

- **Archivos persistentes.** Son las bibliotecas digitales que se preocupan de la curación de los datos y de enfrentar el problema de la evolución de las tecnologías de almacenamiento.

La distinción que se hace entre las colecciones distribuidas de datos y los grids de datos guarda relación con la forma en la que se identifican los recursos, usuarios, archivos, colecciones y servicios. Estas clases de sistemas de manejo de datos difieren en el uso de espacios de nombres. Si bien ambas pueden estar construidas sobre sistemas distribuidos, en la segunda clase, el uso de espacios de nombres permite la construcción de los identificadores de manera independiente. Cabe recalcar que uno de los requisitos básicos para la construcción de identificadores (o nombres) es que para cada identificador haya un único recurso asociado. Juntar dos colecciones es siempre un riesgo de que hayan dos objetos distintos en ellos que sean identificados utilizando el mismo nombre. El uso de espacios de nombre, como componente del identificador resuelve este problema, permitiendo construir sistemas de gestión de datos mayores, donde la gestión de los identificadores puede realizarse de manera distribuida sin riesgo de generar colisiones de nombres. Resulta necesario agregar que la noción de espacios de nombre se encuentra presente también en la Web, dado que los dominios y los prefijos de las URIs permiten gestionar la creación de URIs de manera distribuida.

Para Moore "el manejo de datos ha sido tradicionalmente gestionado con sistemas de software que asumen explícitamente el control sobre los sistemas de almacenamiento locales (sistemas de archivos) o que organizan la información en registros (bases de datos)" [59]. Estas dos formas en las que los sistemas manejan la información difieren en la unidad y la granularidad de los objetos que manejan. Para una, la unidad básica son los documentos, mientras que para la otra lo son los datos. Cuando trabajamos con RDF, nos abstraemos de las serializaciones para operar con conjuntos de tripletas, es decir, pasamos de trabajar con documentos a trabajar con datos.

Entre los sistemas de gestión documental es necesario también destacar los repositorios de documentos. Estos constituyen una segunda capa sobre el sistema de archivos que proveen los sistemas operativos y que pueden construirse sobre multiples sistemas operativos y sobre múltiples dispositivos de almacenamiento. Los repositorios se caracterizan por, junto con

permitir el almacenamiento de documentos, incorporar un identificador para cada documento ingresado y proveer de una interfaz que permite gestionar la metadata de cada uno de los documentos. Los repositorios sirven de base para construir bibliotecas federadas, en la medida en que los repositorios que las conforman implementan protocolos de *metadata harvesting*. Entre los repositorios de documentos destaca DSpace[16], desarrollado por el MIT, y que soporta el protocolo AOI para la federación.

2.5.1. Metadata para catálogos de datos

Los sistemas de gestión documental (y en particular los *data grids*) utilizan la metadata para proveer de contexto a los objetos digitales y para facilitar las búsquedas. Detrás de cualquier sistema de metadatos existe una ontología que describe las clases de objetos y los tipos de metadatos. Típicamente, en los repositorios de documentos esta metadata es usada para describir los documentos y sus relaciones. En algunos casos, la metadata también es usada para describir colecciones de documentos o autores.

En el caso de DSpace los metadatos utilizados corresponden al estándar Dublin Core, aunque puede ser extendido con otros metadatos. El cuadro 2.5 muestra cómo es representada la metadata en un repositorio DSpace (cuando se selecciona mostrar vista detallada). La metadata corresponde a una tabla que asocia términos con valores. Además, en algunos casos se indica el lenguaje de dichos valores.

Los términos de los metadatos son codificados usando los nombres de los términos de Dublin Core. Éstos son ingresados remarcando la jerarquía generalidad/especificidad de los conceptos. Por ejemplo, cuando se usa el término "dc.date.available" se está indicando que el término "available" es un atributo más específico que el término "date" y que éste, a su vez, es un término de la jerarquía "dc". Estas notaciones aplican al uso de RDF, puesto que los predicados deben expresarse mediante URIs y la jerarquía se modela mediante predicados como rdfs:subPropertyOf [60].

[16] DSpace http://www.dspace.org.

DC	Field Value	Language
dc.contributor.author	Meza, F.	-
dc.contributor.author	Perez, J.	-
dc.contributor.author	Eterovic, Y.	-
dc.date.accessioned	2007-11-20T20:49:03Z	-
dc.date.available	2007-11-20T20:49:03Z	-
dc.date.issued	2005	-
dc.identifier.citation	Advanced Distributed Systems 3563: 51- 62	en
dc.identifier.issn	0302-9743	-
dc.identifier.uri	http://dspace.utalca.cl/handle/1950/4054	-
dc.description	Meza, F. Depto. de Ingeniería de Sistemas, Universidad de Talca, Camino Los Niches Km. 1, Curicó, Chile	en
dc.description.abstract	We present a simple implementation of a token-based distributed mutual exclusion algorithm for multithreaded systems. Several per-node requests could be issued by threads running at each node. Our algorithm relies on special-purpose alien threads running at host processors on behalf of threads running at other processors. The algorithm uses a tree to route requests for the token. We present a performance simulation study comparing two versions of our algorithm with a known algorithm based on path reversal on trees. Results show that our algorithm performs very well under a high load of requests while obtaining acceptable performance under a light load.	en
dc.format.extent	2511 bytes	-
dc.format.mimetype	text/html	-
dc.language.iso	es	-
dc.publisher	Springer Berlin / Heidelberg	en
dc.subject	Distributed mutual exclusion, multithreading, parallel programming, concurrent programming, distributed shared memory.	en
dc.title	Implementing distributed mutual exclusion on multithreaded environments: The alien-threads approach	en
dc.type	Article	en
Appears in Collections:	Artículos en publicaciones ISI - Universidad de Talca	

Cuadro 2.5: Metadatos asociados a un recurso en un repositorio DSpace. En este caso se trata del documento identificado por el handle 1950/4054 en el repositorio de la biblioteca de la Universidad de Talca (http://dspace.utalca.cl).

2.6. Datos científicos

Es indudable la importancia del trabajo científico para la comprensión del mundo y para nuestra supervivencia en él. Varios investigadores concuerdan en que la ciencia se está moviendo hacia un cuarto paradigma: el uso intensivo de grandes volúmenes de datos como método de investigación [1]. Este paradigma implica desafíos tecnológicos que incluyen mejoras en los procesos de captura, análisis, modelamiento y visualización de la información científica. Junto con adquirir conocimiento, la ciencia nos ayuda a tomar decisiones para responder a necesidades prácticas. Por ejemplo, el conocimiento del clima nos permite entender cómo éste es afectado por nuestros actos y puede aportar a las decisiones que se tomarán

para enfrentarlo. Se trata en muchos casos de problemas activos y en continuo cambio, que requieren ser abordados con información aún incompleta, pues los eventos no esperarán a que la información pueda ser completada.

Tal como se comentó al inicio de este capítulo, los datos tienen la tendencia a duplicar su volumen cada cinco años. En el campo de las ciencias el crecimiento también ha sido explosivo [61]. En el área de la astronomía se cita que desde 2004 se esperaban incrementos de volumen de 500 TB anuales en el observatorio virtual NVO. Mientras, en un artículo más reciente [62], se indica que el telescopio ALMA producirá desde 2012, 200 TB anuales y el telescopio E-LTC, que se agregará al complejo de telescopios Paranal, producirá 17,4 PB anuales cuando entre en operación el 2017.

El manejo de grandes volúmenes de datos ha abierto nuevos desafíos; no disponemos de herramientas apropiadas para el análisis de datasets del orden de Terabytes o Petabytes [5]. Actualmente no existe una solución única para abordar los heterogéneos datos científicos. Si embargo, se han propuesto estrategias, como las que Jim Gray resumió de manera informal mediante las siguientes leyes [1]:

1. Los cómputos científicos usarán datos de modo cada vez más intensivo.

2. La solución se encuentra en una arquitectura *scale out*.

3. Llevar los cómputos a los datos en vez de los datos a los cómputos.

4. Comenzar diseñando con las 20 preguntas más relevantes en mente.

5. Desarrollar mediante iteraciones funcionales.

Las leyes de Gray constituyen un modelo que se contrapone al de los repositorios de archivos de datos, que fue presentado en la sección 2.5. En los repositorios se supone que los archivos serán lo suficientemente pequeños para que los usuarios puedan descargarlos y analizarlos en sus propias estaciones de trabajo. En cambio, Jim Gray supone que los usuarios deberán consultar los datos a través de servicios que le facilitarán realizar las 20 consultas más relevantes del área.

Para Gray el crecimiento explosivo en el volumen de los datos hace conveniente utilizar arquitecturas que escalen horizontalmente (arquitecturas *scale out*) por sobre a sistemas que

escalen verticalmente (arquitecturas *scale up*). Mientras las arquitecturas scale out enfrentan el aumento en la demanda de recursos agregando más nodos (computadores) al sistema, las arquitecturas scale up agregan más recursos (CPUS o memoria) a cada nodo. Sin embargo, no basta con tener un gran número de nodos accediendo a un mismo arreglo de disco, pues ello generará un cuello de botella en la red. Por consiguiente, Gray propone soluciones en las que cada nodo posea localmente los datos sobre los que va a computar. Este cambio de paradigma requerirá un rediseño de los algoritmos utilizados en cada área, para operar de manera distribuída. No obstante, el dividir y conquistar tiene sus limitaciones cuando las consultas requieren un gran número de joins. Por ello, los particionamientos hechos a los datos deberán tener en cuenta las 20 preguntas más relevantes del área.

La creación de servicios que lleven la consulta a los datos expande la definición de lo que son los datos abiertos. Como se presentó en la sección 2.3.3, la noción definida por las 8 leyes de OGD planteaba que bastaba con permitir el acceso en forma de descarga de archivos procesables automáticamente para que sus datos puedan ser considerados abiertos. Sin embargo, los costosos recursos requeridos para almacenar y procesar grandes volúmenes de datos hacen que, en la práctica, si no se entregan servicios para computar sobre dichos datos, ellos serán inaccesibles al público.

La complejidad de entregar servicios eficientes para el procesamiento de grandes volúmenes de datos implica escoger las consultas más relevantes. Jim Gray propone considerar las primeras 20. Esta selección implica un enfoque contrapuesto con el de los documentos de archivo, donde se buscan los modelos más generales posibles, que permitan responder cualquier tipo de pregunta sobre los datos. Los repositorios de documentos de archivo están diseñados para perdurar inmutables en el tiempo, por el contrario, los servicios de consulta a grandes bases de datos distribuidas estarán sujetas a un cambio constante. Las 20 preguntas podrán cambiar, al igual como cambia la ciencia. Para enfrentar este mundo cambiante Jim Gray propone desarrollar mediante iteraciones funcionales, en vez de embarcarse en un desarrollo demasiado complejo que generará herramientas demasiado tarde, cuando las necesidades ya no sean las mismas.

Capítulo 3

Modelo

Hasta este momento, a lo largo de esta tesis se ha hablado muchas veces de los datos, los documentos y la información, sin entrar en mayor detalle sobre qué son y cuáles son sus diferencias. Siendo el objetivo de este trabajo el proponer una normativa o guía para la publicación de la información y los datos públicos, resulta crucial aclarar qué son dichos objetos. El que no exista un acuerdo sobre qué son los datos y qué son los documentos digitales justifica que hagamos este esfuerzo de conceptualización.

En el modelo propuesto, en vez de intentar definir documentos y datos por sus propiedades, las definiciones se formularán a través de los usos. Es decir, si llegaran a nuestras manos dos objetos digitales, nuestro modelo no nos servirá para distinguir si se trata de datos o documentos. En cambio, para hacer dicha distinción, necesitaremos ver a alguien usarlos.

Para especificar estos usos, a lo largo de esta sección se presentará una serie de modelos de sistemas de gestión documental y sistemas de manejo de datos. Cada sistema definirá un uso dentro del cual hablaremos de datos o documentos. Además, estos modelos matizarán esta distinción dato-documento, presentando una serie de variantes entre la gestión documental y la de datos.

3.1. Documentos

El diccionario de la Real Academia Española [63] entrega dos definiciones para el concepto de documento: "diploma, carta, relación u otro escrito que ilustra acerca de algún hecho, principalmente de los históricos" y "escrito en que constan datos fidedignos o susceptibles de ser empleados como tales para probar algo". Además, una tercera definición, indicada como en desuso, es la de "instrucción que se da a alguien en cualquier materia, y particularmente aviso y consejo para apartarle de obrar mal". Estas definiciones nos acercan a usos primitivos de la palabra documento, como un caso particular de los "escritos", usados para "documentar" algo, es decir, para dejar una constancia de hechos. En cambio, en nuestro uso cotidiano, los documentos han perdido su condición de documentar hechos, extendiéndose a ser considerados como cualquier "escrito", e incluso, imágenes o cualquier cosa que pueda archivarse digitalmente.

La norma ISO 15489-1 [64], sobre la gestión documental indica que un documento es "información u objeto registrado que puede ser tratado como una unidad". De este modo, se pone de manifiesto que los documentos son "información", es decir, un objeto que en la literatura suele caracterizarse en un nivel intermedio de complejidad, entre los datos y el conocimiento.

Con la aparición de los computadores, la gestión documental pasó de trabajar con objetos físicos, como el papel o microfilms almacenados en archivadores y muebles, a trabajar con secuencias de bits, almacenadas digitalmente en dispositivos electrónicos. Este cambio de paradigma remeció la concepción del manejo de la información y generó una distinción entre tres niveles de ella: los datos, la información y el conocimiento. Tradicionalmente se jerarquizó a estos tres elementos definiendo a los datos como los elementos más simples y básicos, a la información como datos acompañados de una descripción de su estructura y contexto, mientras que el conocimiento, como información más la capacidad de hacer algo con ella. No obstante esta jerarquización no es aceptada por todos, para Ilkka Tuomi los datos son un concepto más elaborado que el conocimiento [65]. El conocimiento es aquello que tenemos en nuestra mente y que nos permite actuar. Podría interpretarse que el conocimiento es el estado de nuestra red de neuronas, que cambia a medida que interactuamos con nosostros mismos o con nuestro medio. Cuando andamos en bicicleta y ésta se inclina hacia la izquierda, sabemos que tenemos que girar el manubrio también hacia la izquierda para no caernos. Muchos ciclistas lo hacen, a pesar de no haberlo razonado ni verbalizado. La información, en cambio, sería el producto de un esfuerzo de materializar nuestro conocimiento, por ejemplo, a través de un escrito o de un dibujo. De este modo, la información se convierte en un paso más

allá del conocimiento, en el que lo convertimos en algo que podemos transmitir y comunicar a otros. Por último, los datos son una formalización de la información que permite realizar operaciones automáticas sobre ella. Este último paso requiere modelar conceptualmente la información, definiendo estructuras que nos permitan representar físicamente los datos para escribirlos y luego acceder a ellos. De este modo la jerarquía de Ilkka describe el esfuerzo de llevar el complejo sistema de nuestras mentes a una formalización plasmable en una secuencia de bits que podremos almacenar, transmitir y procesar.

Con esta interpretación de Ilkka del concepto de información, podemos volver a la definición propuesta en la norma ISO 15489-1 de documento, "información u objeto registrado que puede ser tratado como una unidad". El término "objeto registrado" indica que es necesario que los documentos sean registrados, es decir, grabados en algún medio. Por ejemplo, nuestras palabras pueden ser información, pero no serán documento hasta que se las registre en una grabación o en un texto. Nótese que la noción de documento ha dejado de ser usada sólo en el ámbito de los escritos; ahora documento tiene relación con toda información registrada. En particular, cuando hablamos de documento digital, nos referimos a información que ha sido registrada como una secuencia de bits.

Otro elemento de la noción de documento propuesta en la norma ISO 15489-1 es el de unidad. Esto fija el mínimo nivel de granularidad con el que los sistemas documentales tratan a la información: el documento. Tratándose de sistemas documentales digitales, los documentos serán secuencias de bits.

Esta definición de documentos como secuencias de bits choca con dos conceptualizaciones que a menudo hacemos de los documentos. La del documento que está en proceso de edición y la del documento que se representa en más de un formato. Ambas conceptualizaciones exigen que el documento sea visto como un concepto que englobe un conjunto de secuencias de bits. En el caso del documento en proceso de edición, el documento es el conjuto de todas las secuencias de bits que representan a estados por los que ha pasado el documento mientras se lo editaba, mientras que en el caso de los distintos formatos, a todas las secuencias de bits que representan al documento ideal. Sin embargo, en la práctica resulta complejo, o imposible decidir cuales son los límites del documento ideal. Es decir, si tenemos un conjunto S de secuencias de bits, resulta imposible definir un procedimiento genérico para determinar un conjunto S_d que determinen a un determinado documento ideal d. En la práctica, cuando editamos un documento en nuestros computadores y su secuencia de bits es modificada, lo que nos hace mantener la noción de que se trata del mismo documento es su ruta en el sistema de archivos. En consecuencia, los documentos ideales son conceptos que nos encontramos

definidos a través de los metadatos.

Llamaremos "documentos vivos" a aquellos documentos ideales que por medio de los metadatos quede establecido que pertenecen a "versiones" de un mismo documento a lo largo de su edición. De igual manera, llamaremos "documentos lógicos a" los documentos ideales que están asociados a un conjunto de documentos digitales que los instancian como "representaciones físicas" de ellos en formatos específicos.

En base a esta definición de documentos como secuencias de bits, se presentará una serie de modelos de gestión documental, que establecen una clasificación de ellos. Todos ellos tendrán en común el permitir a un agente recuperar íntegramente un documento a partir de una consulta. La figura 3.1 nos presenta la estructura de todos estos modelos, compuesta por un agente y una máquina capaz de responder las consultas del agente.

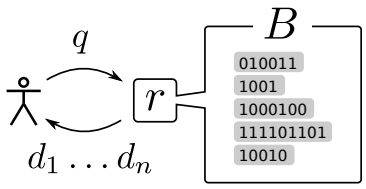


Figura 3.1: Esquema de los modelos de gestión documental. A la izquierda se presenta un agente, que es capaz de enviar una consulta q a un sistema, representado por una máquina r que retorna uno o más documentos d que satisfacen o son relevantes para la consulta recibida. A la derecha de la función de recuperación se encuentra la biblioteca, es decir, los datos que usa el sistema para responder a las consultas.

La máquina (representada en la Figura 3.1 por r) posee dos cintas infinitas, que usará respectivamente para recibir la consulta y para responderla. Además posee tres espacios de almacenamiento infinitos donde podrá almacenar la biblioteca (representada por B), el programa para responder las consultas y sus cálculos temporales. Si bien estos espacios de almacenamiento podrán ser infinitos, el espacio usado en ellos será siempre finito.

No es necesario entrar en detalle de cómo la máquina es capaz de ejecutar el programa, nos basta con suponer que posee cualidades equivalentes a las de los procesadores modernos en términos de capacidad de cómputo. Dicho de otro modo, lo que necesitamos es que nuestro sistema sea equivalente a una máquina de Turing.

Antes de entrar al desarrollo de los modelos de sistemas documentales, es necesario remarcar que la clasificación que se hará de ellos no se hará en función de la complejidad de cómputo

de ellos, sino en relación a la funcionalidad o problema que buscan resolver. Así por ejemplo, dentro del modelo de recuperación de información no binario puede haber varios submodelos instanciándolo, cada uno de los cuales definirá lo que son las consultas y los documentos. El invariante de todos ellos será el poder recibir una consulta y procesar el conjunto de documentos para identificar la relevancia de cada documento respecto de la consulta.

3.1.1. Sistemas documentales muertos

Para iniciar la caracterización de los sistemas documentales nos abstraeremos de sus posibles cambios en el tiempo, definiendo los sistemas documentales muertos, como aquellos en los que prestamos atención sólo a la resolución de una única consulta.

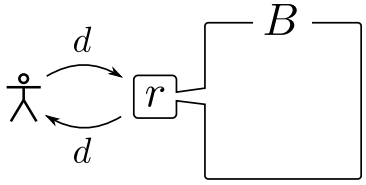
Cada sistema responde con un conjunto de documentos D que es un subconjunto del conjunto de todos los documentos posibles $\mathcal{D}$. Es importante destacar que $\mathcal{D}$ corresponde a todas las combinaciones de bits (aunque en la práctica de bytes) de largo finito, por lo que se trata de un conjunto infinito numerable. En cambio, D podría ser finito o infinito, según lo cual hablaremos de sistemas documentales finitos o infinitos, respectivamente.

En base a esta definición de documentos como secuencias de bits, se planteará un primer tipo de sistema de gestión documental: la Biblioteca de Babel[1].

M1 – Biblioteca de Babel. Este sistema corresponde a un espacio imaginario en el que están almacenados todos los documentos posibles, es decir $D = \mathcal{D}$. Para obtener un libro de esta biblioteca contamos con una función $r : \mathcal{D} \to \mathcal{D}$ definida simplemente como la función identidad. Si bien, en el mundo físico, la Biblioteca de Babel resulta imposible de construir, en el mundo digital nos basta con un servicio eco (ver figura 3.2).

Contradictoriamente, aunque completa, la Biblioteca de Babel es completamente inútil porque no agrega información a lo que el agente ya tenía. Una variante de este modelo es definir un conjunto de documentos finitos $D \subset \mathcal{D}$ donde el sistema buscará dentro de D los documentos más parecidos a un documento q que entregamos como referencia. Esta funcionalidad es la misma que realizan los sistemas de recuperación de información y que serán estudiados más adelante.

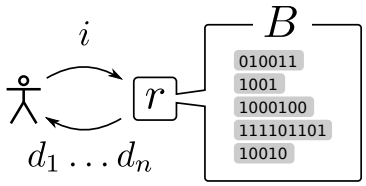
[1] Nombre tomado del título de un cuento de Jorge Luis Borges.



$$r(d) = d$$

Figura 3.2: Modelo de la Biblioteca de Babel. Para este modelo no es necesario almacenar información. Lo único que requiere tal biblioteca es la codificación de la función eco en el espacio de programa de la máquina.

M2 – Biblioteca de hashing. Este modelo es una variante de la Biblioteca de Babel. En él se supone que hay un conjunto finito de identificadores $\mathcal{I}$ y que la biblioteca almacena un conjunto de documentos D que es también finito y menor en tamaño a $\mathcal{I}$. Además existe una función de hashing $h : \mathcal{D} \to \mathcal{I}$ que nos permite recuperar sólo aquellos documentos cuyo resultado de aplicar h sea igual a lo solicitado (ver figura 3.3).



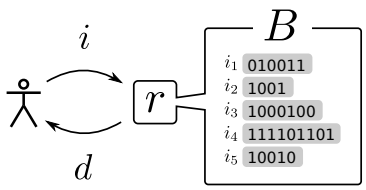
$$r(i) = \{d \in D \mid h(d) = i\}$$

Figura 3.3: Modelo de la biblioteca de hashing. El argumento i funcionará como identificador sólo cuando h es inyectiva sobre D.

La ventaja de este sistema de información sobre la Biblioteca de Babel es que no se necesita conocer el documento completo, en cambio, basta un identificador para poder recuperarlo. Resulta necesario recalcar la relevancia que se le confiere al uso de identificadores para los documentos desde el momento del ingreso al sistema. Esta relevancia queda de manifiesto también en la norma ISO 15489-1 donde el registro es definido como el "acto por el que se atribuye a un documento de archivo un identificador único al introducirlo al sistema".

En el caso de la biblioteca de hashing el identificador es calculado a partir del documento. Una estrategia distinta es la que sigue el sistema de registro, donde el identificador es escogido externamente, siguiendo algún protocolo convenido para ello.

M3 – Sistema de registro. Al igual que la biblioteca de hashing este sistema utiliza una función que que asocia identificadores con documentos, pero se diferencia en que no existe una función genérica $h : \mathcal{D} \to \mathcal{I}$ que los asocie, sino que esta información es codificada dentro de los datos de la biblioteca mediante una función inyectiva $f : D \to \mathcal{I}$, donde D es finito e $\mathcal{I}$ podría ser infinito (ver figura 3.4).



$$r(i) = \{d \in D \mid f(d) = i\}$$

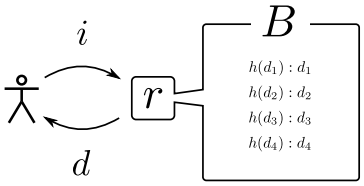
Figura 3.4: Modelo de la biblioteca de registro. Cada identificador i está asociado a un único documento $d \in D$. Como D es finito, la función f puede ser codificada de manera finita en el espacio de la biblioteca.

Tanto en los sistemas de hashing como en los de registro, la única posibilidad de obtener un documento es conocer previamente su identificador. Si bien puede parecer que el requisito de conocer el identificador de un documento para poder buscarlo hace a estos sistemas impracticables, en la práctica son comunes. Por ejemplo, los registros de las escrituras públicas en Archivos Nacionales son recuperables mediante la notaría, el número de repertorio y las fojas. La razón es que las notarías juntan las escrituras públicas en libros llamados repertorios, que una vez cerrados son entregados a Archivos Nacionales para su preservación. Estos repertorios son ordenados internamente por número y por notaría. Dado que existen varias notarías y repertorios, buscar una escritura sin conocer dichos parámetros puede requerir de tanto esfuerzo que en la práctica, para los usuarios, Archivos Nacionales se comporta como tal modelo.

Una de las cualidades principales del sistema de registro es que los documentos y los identificadores son opacos para el código del programa que define su máquina, es decir, el programa de la máquina sólo necesita verificar la igualdad entre identificadores para encontrar el identificador recibido como parámetro y no aplica ninguna función sobre el documento, salvo leerlo para escribirlo en la salida. En cambio, para la biblioteca de hashing los documentos no son opacos, pues requieren calcular sobre ellos la función h.

El sistema de registro sirve de base para la implementación de otros sistemas. De este modo, varios de los sistemas que se mostrarán a continuación pueden ser descompuestos de manera tal que una de sus componentes es un sistema de registro. En particular, un sistema de

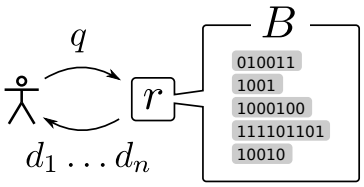
registro puede ser utilizado como componente trivial de un sistema de hashing tal como lo muestra la figura 3.5.



$$r(i) = \{d \in D \mid f(d) = i\}$$

Figura 3.5: Biblioteca de hashing descompuesta utilizando un sistema de registro. El sistema de registro es implementado expresando por extensión la función h sobre el conjunto D.

M4 – Sistema de recuperación de información binario. Estos sistemas se definen mediante una función f cuyo dominio es $Q \times \mathcal{D} \times \wp\mathcal{D}$, donde Q es un conjunto de consultas posibles (o lenguaje de consulta) y $\mathcal{D}$ es el conjunto de todos los documentos posibles. Al igual que con el sistema de registro y con la biblioteca de hashing, el sistema de recuperación de información binario almacena un conjunto D finito. La respuesta de este sistema corresponde al conjunto de elementos dentro del conjunto para el cual la función f retorna 1 (ver figura 3.6).

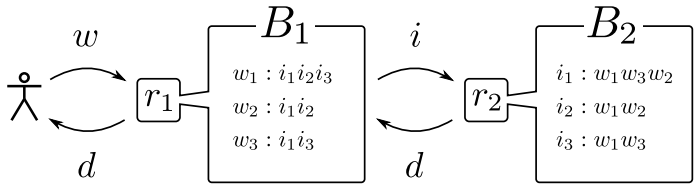


$$r(q) = \{d \in D \mid f(q, d, D) = 1\}$$

Figura 3.6: Modelo de recuperación de información binario.

Es común que los sistemas de recuperación de información no trabajen sobre los documentos directamente, sino con un modelo de ellos. Por ejemplo, cuando las consultas son palabras y los documentos son modelados como listas de palabras, la función $f(q, d, D)$ entrega 1 si el documento d posee al término q y 0 si no. Tal modelo es descomponible en un modelo que usa como información índices de ocurrencia de palabras en documentos y un modelo de registro (ver figura 3.7). Esa descomposición permite que la función evalúe sin los datos del modelo, sin necesidad de procesar los documentos.

La figura 3.7 presenta una variante al modelo original, pues la descomposición agrega la capacidad de que la máquina actúe también como agente, escribiendo y leyendo en las cintas



$$r_1(w) = \{r_2(i) \mid i \in f(w)\}$$

Figura 3.7: Modelo de recuperación de información binario de búsqueda de documentos que incluyen una determinada palabra. La función f calcula el conjunto de identificadores de los documentos en los cuales se encuentra una palabra, a partir de la información en B_1.

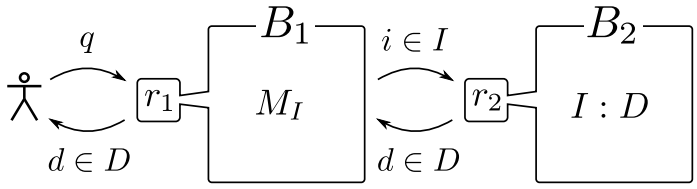
de entrada y salida de una segunda máquina. Otra variante de esta descomposición consiste en suponer que el agente es capaz de tomar la respuesta de una máquina y hacer una consulta a una segunda máquina.

M5 – Sistema de recuperación de información no binario. Es una variante del sistema de recuperación binario en el cual la función f no entrega valores binarios, sino valores reales que permiten establecer un ranking entre los documentos según su relevancia frente a la consulta. De este modo, el resultante de este tipo de modelo es una lista ordenada de documentos o un conjunto de pares ordenados de la forma $(d, f(q, d, D))$.

Este tipo de modelos es una componente principal de la mayoría de los sistemas de recuperación de información, entre los que se encuentran, por ejemplo aquellos en los que la función f es una variación del modelo TF∗IDF [66].

M6 – Sistema con metadatos. El modelo de metadatos es el complemento de los modelos de recuperación de información (binarios o no). Mientras en el modelo de recuperación de la información la función que seleccionaba o calculaba la relevancia de los documentos respecto a una consulta usaba como parámetro los documentos, en el modelo de metadatos la función usará como argumento sólo los metadatos. De este modo, el sistema de metadatos puede ser comprendido como la concatenación de dos sistemas. El primero recibe una consulta y nos devuelve un identificador $i \in I$ y el segundo es un sistema de registro que recibe i y nos devuelve el documento (ver la figura 3.8).

Este tipo de sistemas es común en los repositorios documentales, como por ejemplo DSpace, donde los documentos son recuperados mediante sus metadatos. Además, por razones de



$$r_1(q) = \{r_2(i) \mid i \in f(q, M)\}$$

Figura 3.8: Sistema con metadatos. Este sistema se construye en base a una función $f(q, M)$ que entrega un conjunto de identificadores que permiten recuperar los documentos usando un sistema de registro. Es decir, el sistema de metadatos se construye como la concatenación de un sistema documental cualquiera y un sistema de registro.

eficiencia, los sistemas de recuperación de información son construidos como sistemas de metadatos, en los que M es construido como un índice a partir de los documentos ($I : D$). Esta última observación nos lleva a diferenciar entre metadatos que no pueden ser deducidos desde un documento, como por ejemplo, la fecha en la que un documento fue escrito, de aquellos que son deducibles de los documentos, como por ejemplo, el histograma de color de las imágenes o las frecuencias de términos en documentos de texto.

M7 – Sistema documental generalizado. Por último tenemos una generalización de todos los modelos anteriores que combina los parámetros de f, siendo el nuevo dominio el conjunto $Q \times D \times \wp D \times M$.

3.1.2. Comparación entre sistemas documentales muertos

El cuadro 3.1 presenta los modelos descritos en relación a la información que es usada para la recuperación de documentos. La propiedad común entre todos ellos es que permiten recuperar íntegramente documentos. Esta recuperación íntegra del bitstream es una de las diferencias fundamentales entre la gestión documental y la gestión de datos, que será presentada en la siguiente sección, y por ende, una de las características que nos permitirá diferenciar entre datos y documentos.

Sistema	D	I	Q	M
Biblioteca de Babel	X			
Biblioteca de hashing	X	X		
Sistema de registro		X		
Sistema de recuperación de la información	X		X	X
Sistema con metadatos		X	X	X
Sistema generalizado	X	X	X	X

Cuadro 3.1: Las características comparadas son el uso de: datos como base para responder las consultas (D), identificadores para acceder a los documentos (I), lengujes de consulta (Q) y metadatos (M).

3.1.3. Sistemas documentales vivos

En los modelos previos, los documentos codificados en los sistemas documentales no cambiaban. En esta subsección se introducirá una serie de eventos que pueden modificar el estado del sistema y por los cuales los agentes podrán obtener distintos resultados según el momento en el cual realicen sus consultas. De este modo la interacción podrá ser vista como una secuencia de eventos que ocurren asíncronos y que modifican el estado del sistema (ver figura 3.9), más una secuencia de consultas.

Eventos (editor)		e_0		e_1			$\cdots$
		$\downarrow$		$\downarrow$			$\cdots$
Estados (sistema)	s_0	s_0	s_1	s_1	s_2	s_2	$\cdots$
	$\uparrow$		$\uparrow$		$\uparrow$		$\cdots$
Consultas (agentes)	q_0		q_1		q_1	q_2	$\cdots$
Tiempo	0	1	2	3	4	5	$\cdots$

Figura 3.9: Secuencia de eventos y consultas en un sistema documental vivo. Dado que los eventos modifican los estados, podría ocurrir que la consulta q_1 no de los mismos resultados en los tiempos 2 y 4.

Para los sistemas documentales basta con definir tres eventos posibles: la agregación y eliminación de documentos y la edición de los metadatos.

Agregación de documento [$\oplus(d)$] Consiste en agregar un documento al conjunto de documentos, es decir:

$$D_t \xrightarrow{\oplus(d)} D_{t+1} = D_t \cup \{d\}$$

Eliminación de documento [$\ominus(d)$] Consiste en quitar los documentos que responden a una consulta, es decir:

$$D_t \xrightarrow{\ominus(q)} D_{t+1} = D_t \setminus r_t(q)$$

Edición metadatos [$\otimes(M)$] En el caso de los metadatos, como no se han definido en relación a su estructura, bastará decir que el cambio consiste en reemplazar el estado actual de ellos por un nuevo estado, es decir:

$$M_t \xrightarrow{\otimes(M)} M_{t+1} = M$$

3.1.4. Sistemas documentales con seguridad

Una extensión de los modelos anteriores consiste en entregar una componente de autorización e identificación, que permitiría a los sistemas decidir si responder según las credenciales de los agentes. Este tipo de sistemas son una extensión de los sistemas de metadatos, pues mediante los metadatos puede codificarse a quién se le permite el acceso a cada documento y en las consultas pueden agregarse los parámetros para la identificación del agente. Si no se hace en la consulta, se requeriría que los sistemas documentales tengan memoria para manejar las sesiones, tal como lo hacen gran parte de las aplicaciones en la Web. Sin embargo, dicha memoria sólo agrega comodidad al usuario, no aporta funcionalidades al sistema de identificación.

3.1.5. Sistemas documentales con memoria

Es común que hoy los sistemas de recuperación de información usen los logs de los usuarios para mejorar la asociación entre los documentos y las consultas. De este modo la respuesta a una consulta puede cambiar dependiendo de la cantidad de veces que se hace y de la alternativa que el agente escoge entre ellas. Incluso estas huellas de los agentes son usadas no sólo para afinar las repuestas de la máquina al conjunto de los agentes, sino también con el fin de generar respuestas que puedan ser relevantes a la vista de las necesidades del agente en cuestión.

3.1.6. La Web como un sistema de registro

Sin considerar su lenguaje, la Web puede ser comprendida como un sistema de registro donde las URLs son los identificadores y HTTP define la recuperación de los documentos. En la práctica se trataría de un sistema de registro vivo, pues no se garantiza que las peticiones sobre las mismas URIs entreguen las mismas respuestas, con seguridad, pues no todos pueden acceder a los mismos recursos, y con memoria, porque muchas veces las respuestas dependen de lo que hemos consultado previamente. No obstante, preferiremos omitir estas características y nos quedaremos con que se trata de un sistema de registro.

Tal como vimos en la sección 2.1.3, al considerar los hipervínculos, los documentos de la Web pueden ser modelados como un grafo dirigido [11]. Sin embargo, este grafo nunca es accedido completamente por el agente, sino que siempre tiene una visión parcial de lo que va recorriendo. Aunque el grafo completo esté codificado al interior del sistema de registro, este sistema se limita sólo a entregar secuencias de bits asociadas a identificadores. De este modo, el trabajo de interpretación de la Web queda en manos de un agente que nunca verá la red por completo.

3.2. Datos

La sección anterior, donde nos referimos a los documentos como secuencias de bits, nos servirá de base para abordar la conceptualización de qué son los datos. Frente a esto distinguimos dos posturas principales. La primera de ellas considera a los datos como secuencias de bits, al igual que los documentos. Mientras, la segunda los define como los elementos atómicos, definidos a nivel lógico, cuya reunión constituye una base de datos. Esta segunda definición es similar a la del documento lógico que vimos en la sección anterior, pues los datos y la base de datos no corresponden a las instanciaciones como secuencias de bits, sino a un conjunto de datos "ideal" que permite definir clases de instanciaciones equivalentes. Esta segunda conceptualización se presenta en la ontología para la publicación de datos DCat, donde el conjunto de datos ideal corresponde al dataset y sus instanciaciones a las distribuciones del dataset [67].

3.2.1. Datos como bitstreams

En varias comunidades los datos se definen simplemente como cadenas de bits o bitstreams, por ejemplo, Moore y Rajasekar afirman que: "la comunidad de *data grid* define a 'datos' como cadenas de bits que componen una entidad digital. Una entidad digital puede representar, por ejemplo, un archivo digital, un objeto en un *ring buffer*, un registro de una base de datos, una URL o un objeto binario en una base de datos" [59]. De este modo, al igual que en el modelo de documentos, los data grids requieren de repositorios en los cuales los datos sean almacenados. Ya sean sistemas de archivos o registros en sistemas de bases de datos.

Al igual que en los sistemas documentales de registro y de metadatos, la identificación de los datos en el mundo de los data grids es fundamental. En sus inicios los data grids fueron propuestos como un complemento a los grids, es decir, como un sistema de almacenamiento distribuido que podría servir a las redes de cómputo distribuidas [68]. Los datos eran inicialmente almacenados como archivos, en la carpeta de cada investigador. Ello permitía un acceso restringido a un sólo usuario y la identificación de los datos podía realizarse mediante los nombres de archivo. Esta forma de identificación y acceso cambió con la generación de repositorios compartidos de datos, que pasaron de ser compartidos por equipos de investigadores a ser compartidos dentro de redes federadas de instituciones. Para poder implementar la identificación en dicho ambiente distribuido y multi institucional, se adoptó la convención de usar espacios de nombres para crear identificadores globales para los datos. Un resumen de la evolución de los data grids puede encontrarse en [59].

La interfaz básica que proveen los data grids consiste en dos servicios: acceso a datos y acceso a metadatos. Dado que los datos son vistos como secuencias de bits, esta interfaz resulta equivalente a la de los sistemas documentales con metadatos presentados previamente en la sección 3.1. Según Ann Chervenak y su equipo [68], las componentes básicas de los data grids son:

1. *Acceso y almacenamiento de datos.* Es importante recalcar que ellos entienden por datos a las instancias de archivo. El almacenamiento se lograría, por ejemplo, con sistemas de archivos Unix, servidores HTTP, sistemas de almacenamiento jerárquico como *High Performance Storage System* (HPSS), sistemas de almacenamiento distribuidos como el *Distributed Parallel Storage System* (DPSS) o sistemas que mapean múltiples sistemas de almacenamiento, como el *Storage Resource Broker* (SBR). Mientras, para el acceso se plantea la necesidad de elaborar una API común.

2. *Acceso y almacenamiento sobre metadatos.* Dado que no existe una convención sobre el formato de los metadatos (ver sección 3.2.4), en [68] se propone usar estructuras jerárquicas y distribuidas como las definidas en el *Lightweight Directory Access Protocol* (LDAP).

3. *Control de autorización y autentificación.* La *Grid Security Infraestructure* (GSI) [69] sería suficiente.

4. *Gestionar la reserva de recursos.* El objetivo es garantizar que las transferencias entre dos nodos tengan un comportamiento predecible (por ejemplo [70]).

5. *Medir y estimar rendimientos.* Esto incluye mediciones sobre el almacenamiento, las redes y computadores (por ejemplo el Network Weather Service [71]).

6. *Instrumentar las operaciones entre nodos.* Ello implica medir los rendimientos de las transferencias y otras operaciones (por ejemplo NetLogger [72], Pablo [73] y Paradyn [74]).

De entre estas seis funcionalidades básicas que se enumeraron, las tres primeras son parte de aquellas que modelamos en los sistemas documentales. Las otras, son funcionalidades que no están al nivel de entregar formas de acceder a la información, sino al de las cualidades de gestión del rendimiento del sistema. Por esta similitud entre las funcionalidades de los sistemas documentales y los datagrids es que no es raro encontrar propuestas para su integración [59].

En un nivel superior de la arquitectura de datagrid aparecen otras funcionalidades como: *a*) la gestión de Réplicas y caché y *b*) la creación de réplicas y selección de réplicas [68].

Gestión de Réplicas y caché. La gestión de réplicas consiste en crear o borrar instancias de archivos en nodos de almacenamiento específicos. Típicamente una réplica es creada para ofrecer un mejor rendimiento para el acceso a los datos, por ejemplo, cuando desde una localización A se dispone una conexión más veloz a un servidor B que a un C es conveniente mover archivos desde C a B para que A los pueda acceder directamente (y en menor tiempo) desde B. Mientras, la gestión de caché consiste en crear una copia del archivo, pero sin agregarlo al catálogo, de modo que quede disponible para ser usada sólo localmente.

La gestión de réplicas y caché exige que los archivos replicados sean sólo de lectura. De este modo, se evitan los problemas relacionados con la actualización de los archivos y la

coherencia. Tal como se había discutido previamente con respecto al presentar la distinción entre documentos vivos y muertos, el requisito de que los archivos sean de sólo lectura permite crear versiones en una capa más arriba, a través de los metadatos.

Creación y selección de réplicas. Una funcionalidad de la selección de réplicas es interesante porque no es construida sobre las funcionalidades básicas de los datagrids ni las que habíamos presentado previamente como componentes de los sistemas documentales. A diferencia de la selección de conjuntos de documentos relevantes para una consulta en los sistemas documentales, la selección de réplicas permite seleccionar partes de un documento, con el fin de minimizar los tiempos de la transmisión y el uso de recursos de almacenamiento.

Un ejemplo de sistemas que permiten esta funcionalidad es el *Storage Access Coordination System* (STACS) [75], que es capaz de satisfacer las necesidades de las aplicaciones de la física de alta energía al permitir la extracción de un subconjunto de los datos contenidos en un archivo. Para ello STACS usa un complejo sistema de indexación que representa los metadatos para los eventos contenidos en un archivo. Una estrategia distinta para proveer la misma funcionalidad es la que desarrollan aplicaciones que incrustan los metadatos en formatos auto descriptivos como el *Network Common Data Format* (NetCDF) [76] o el *Hierachical Data Format* (HDF) [77].

La funcionalidad de seleccionar datos implica la habilidad de aplicar funciones de filtrado sobre la estructura de los archivos y entregar una secuencia de bits distinta a la almacenada. De este modo se rompe el concepto de recuperación íntegra de la información, que fue presentado en los sistemas documentales y hace una primera aproximación hacia el concepto de datos. Los documentos dejan de ser oscuros para ser entendidos como serializaciones de estructuras de datos. De este modo, un documento puede ser interpretado como un arbol para seleccionar una rama dentro de él y replicarla en otro nodo del datagrid como un documento más.

Romper con la recuperación íntegra a través de la selección de partes conceptuales de un documento es un primer paso de los documentos a los datos. Un segundo paso es la agregación de información contenida en más de un documento o en partes distintas de uno de ellos. Es decir, luego de seleccionar partes de una estructura mayor, un segundo paso es poder juntar estas partes para obtener nuevos resultados. Nuevamente, para que esto sea posible es necesario contar con un modelo conceptual sobre el formato de los bitstreams.

Algunos datagrids son capaces de transformar el formato de los datos, extraer subconjuntos, convertir tipos y realizar una transimisión directa de datos entre nodos que usen sistemas de almacenamiento distinto. Estas funcionalidades han sido demostradas como parte del *Active Data Repository* [78]. Además, sistemas como Globus, ofrecen la habilidad de extraer datos arbitrariamente y ejecutar operaciones de procesamiento sobre los archivos, como parte de las actividades de manejo de datos.

La evolución de los datagrids incorporando funciones que permiten computar consultas localmente valida la afirmación que hace Jim Gray frente a la pregunta de ¿dónde realizar los cómputos? (ver sección 2.6). Los altos costos de transferir datos de un nodo a otro, como se hace en el modelo original de los datagrids, se evitan si conseguimos construir modelos de consulta de los datos que minimicen dichas transferencias. De este modo, el modelo propuesto por Jim Gray consiste en dividir los cómputos en pequeños nodos, cada uno con sus propios datos para procesar.

3.2.2. Bases de datos

A diferencia de los datagrids, las bases de datos son desarrolladas en el sentido opuesto. Mientras los datagrids son sistemas que siguen el camino desde ser sólo repositorios de bitstreams hacia ser sistemas donde se definen ciertas operaciones que requieren la interpretación de estructuras codificadas en los bitstreams, los sistemas de bases de datos suelen presentarse como modelos conceptuales antes de ser implementados. Esto hace que cada sistema de base de datos tenga un modelo de datos y un lenguaje de consulta bien definidos, independientemente de su posterior implementación. Es así como, por ejemplo, el modelo que Codd definió en [79] es lo que tenemos en mente cuando accedemos a datos relacionales. Dicho de otro modo, cuando nos enfrentamos a bases de datos nos abstraemos de los bitstreams que son almacenados en disco, pues se trata de sólo serializaciones particulares en disco que en la práctica no nos interesan. Esta diferencia hace una distinción entre los documentos digitales y los datos, mientras los primeros son secuencias de bits los segundos son componentes de modelos a nivel lógico. De este modo un dataset pasa a ser un conjunto de datos definidos a nivel lógico y una distribución a una instanciación de ella como un bitstream que es almacenable y transmitible por la red.

Existen diversos tipos de bases de datos, entre los que podríamos citar, las relacionales, las deductivas y las semi-estructuradas. En los dos primeros tipos, los datos son los registros, es

decir, las tuplas en el caso del modelo relacional y los hechos en el caso de las bases de datos deductivas. Resulta sencillo mapear una base relacional como una base de datos deductiva en la que cada registro r de una relación R se entienda como un hecho $R(r)$. De esta manera, podemos entender estos dos primeros tipos de bases de datos como un conjuntos de hechos. En consecuencia resulta natural llamar datos a los hechos, lo que además concuerda con el uso que le damos al término *dataset* y con el uso común, que encontramos en el diccionario Webster (1913) de dato:

1. Algo dado o admitido; un hecho o principio probado; algo sobre lo que se puede inferir o fundamentar argumentos.

2. Una pieza de información; un hecho; especialmente una pieza de información obtenida de la observación o experimentación.

3. Las cantidades o relaciones que son asumidas como dadas en un problema.

Por su parte la RAE propone también varias definiciones para dato e indica que en el campo de la informática datos son: "Información dispuesta de manera adecuada para su tratamiento por un ordenador". Sin embargo, esta definición tiene el problema de no poder escapar a la ambigüedad de lo que significa el "tratamiento por un ordenador".

Entre los modelos de datos semi-estructurados nos encontramos: XML, YAML, JSON, NetCDF o HDF. En este tipo de estructuras no resulta tan claro cuáles son los datos que forman los datasets. Algunas alternativas son: *a*) mapear los datos semi-estructurados a datos estructurados [80, 81], *b*) distinguir tuplas dentro de árboles [82], y *c*) en vez de definir la unión como la forma de componer datasets dentro de los datos, usar operaciones de composición estructural.

3.2.3. Datos en la Web

En el modelo RDF los datos corresponden a las tripletas y los datasets a los grafos. Cuando consideramos los datos en la Web, podemos entender que cada recurso que es accesible por una URI y que en algunos de ellos nos encontraremos con definición de tripletas cuyas URIs serán dereferenciables, permitiéndonos llegar a más datos. Este modelo de la Web de los datos es el propuesto en [83].

3.2.4. Metadatos

En la sección 3.1.1 definimos los metadatos como otro sistema documental anexo que usábamos para recuperar los identificadores de documentos. Estos metadatos podrían describir características de los documentos o constituir un índice para su búsqueda. Ahora que ya contamos con una noción de datos a nivel lógico, podemos definir a los metadatos, como datos en los cuales aparecen los identificadores de los recursos que están siendo descritos. En el caso de los sistemas documentales, los metadatos conforman un paquete distinto al de los documentos, que es intercambiado mediante protocolos de *harvesting* como el *OAI Protocol Metadata Harvesting* (OAI-PMH) desarrollado por la *Open Archives Initiative* (OAI). En cambio en el caso de los datos, los metadatos suelen fundirse con los mismos datos, como con los datos semi-estructurados, donde los datos son autodescriptivos, o con RDF, donde comparten el mismo grafo.

Capítulo 4

Casos de estudio

Durante el desarrollo de esta tesis se trabajó en cuatro proyectos que sirven de casos de estudio para las propuestas que siguen el desarrollo de esta tesis en los capítulos 5, 6 y 7. De entre estos proyectos, los datos publicados en el contexto de la transparencia activa y en las biografías conforman el caso de estudio para los vocabularios que se propondrán en el capítulo 7.

4.1. Datos de gobierno

Enmarcado en el extendido impulso por publicar datos (ver sección 2.3.3), el gobierno chileno lanzó un portal de datos públicos[1]. En el preproyecto se trabajó en la definición y documentación de prácticas para la publicación de datos y en la investigación del estado del arte de los catálogos de datos de otros países. El trabajo realizado decantó en parte en las recomendaciones generales presentadas en el capítulo 5.

[1] http://datos.gob.cl

4.2. Datos científicos

En el marco de un proyecto realizado con la ONG Derechos Digitales[2] para la definición de una política para los datos generados con fondos entregados por CONICYT, se hizo un estudio sobre los datos científicos y una propuesta para su publicación.

A nivel mundial no existen políticas claras y muchas de ellas están diseñadas para un contexto distinto del actual. Por ejemplo, la *Deutsche Forshungemeinshaft* (DFG), la organización que financia la mayor parte de la investigación científica en Alemania, recomienda que "los datos originales sobre los que se sustentan las publicaciones deben ser almacenados de manera segura por diez años en un formato durable en la institución donde fueron generados" [84]. El hecho de que estas políticas hayan sido desarrolladas en 1998 deja ver que en aquellos tiempos el problema de enfrentarse a grandes volúmenes de datos aún no se había interiorizado en la comunidad científica. En la recomendación de la DFG los datos que deben almacenarse son aquellos tomados por los investigadores en el laboratorio, su cuadernillo o datos que pudiera almacenar de manera digital en un CD, es decir, la recomendación apunta al tratamiento de datos pequeños.

En este contexto de pequeños datasets, con formatos y esquemas que dependerán de cada investigación, resulta natural tratar de garantizar la preservación y acceso posteriores a los datos mediante la publicación de ellos como archivos descargables de un catálogo, al igual que se plantea para datos de gobierno. Además, resulta crucial el uso de identificadores para los datasets publicados en el catálogo, pues ello permitirá que sean referenciados desde los estudios que se basan en ellos.

En cambio, para grandes proyectos, lo común es que los datos se almacenen en sistemas o formatos que son estándares del área. Por ejemplo, el uso de HDF en geofísica permite que grandes volúmenes de datos puedan ser compartidos como un esfuerzo común de varios centros de investigación y consultados mediante protocolos que permiten seleccionar rangos de datos para su procesamiento. De este modo, dada su heterogeneidad en tamaños y formatos, la publicación de datos científicos no puede ser abordada con una única estrategia.

[2] http://www.derechosdigitales.cl

4.3. Transparencia activa

La Ley Nº 20.258, sobre el acceso a la información pública, define dos formas de acceso. La primera de ellas es la transparencia activa, que consiste en la publicación obligatoria de los organismos públicos de un conjunto específico de datos. La segunda es la transparencia pasiva, que define un procedimiento por el cual solicitar información no publicada a los organismos públicos.

Los temas sobre los que trata la transparencia activa incluyen: *a*) marco normativo, *b*) actos y resoluciones, *c*) estructura orgánica, *d*) dotación de personal, *e*) compras y adquisiciones, *f*) presupuesto, *g*) transferencias, *h*) auditorías del ejercicio presupuestario, *i*) trámites, *j*) subsidios y beneficios, *k*) participación ciudadana y *l*) vínculos institucionales. En resumen, la transparencia activa trata sobre la estructura y las contrataciones dentro de los organismos públicos, cómo se usan los prespuestos y qué beneficios otorgan los organismos.

Para la implementación de la transparencia activa en los organismos del gobierno se publicaron una serie de instructivos y decretos[3], que se suceden unos a otros complementando o introduciendo modificaciones en los previos.

En esta tesis se han escogido dos áreas de la transparencia activa: la definición de la estructura orgánica de los organismos y las contrataciones de personal[4].

4.4. Biografías parlamentarias

El trabajo realizado en la Biblioteca del Congreso Nacional de Chile (BCN) se desarrolló en el contexto de una serie de proyectos que es documentada en [85] como un esfuerzo de la Biblioteca para incorporar las tecnologías de la Web Semántica y abrir datos a la comunidad.

La Biblioteca tiene un sistema de gestión de la información parlamentaria, que reúne fichas de parlamentarios y partidos políticos. Dichas fichas contenían tanto una biografía escrita como datos semi-estructurados que eran marcados mediante el sistema de plantillas de la

[3] Una recopilación de ellos se encuentra en http://www.gobiernotransparente.cl/asistente/documentos.php.

[4] Para comprender la estructura de las contrataciones lo más conveniente es leer el Instructivo Nº 9 del Consejo para la Transparencia, publicado en el Diario Oficial el 20 de agosto de 2010.

plataforma usada: MediaWiki 1.8[5], la plataforma utilizada por la Wikipedia[6] para la edición y publicación de sus contenidos.

Al igual que la mayoría de las demás plataformas wiki, MediaWiki está basada en la edición de páginas a través de un código, diseñado para ser fácil de entender y escribir, que es convertido en HTML al momento de visualizar las páginas. La capacidad de mantener las múltiples versiones de las páginas hace de MediaWiki una plataforma ideal para la edición de los contenidos por un grupo numeroso de personas.

4.4.1. De información a datos

El uso de MediaWiki en las Biografías Parlamentarias sigue el modelo de generación de datos presentado en el capítulo 3 y que fue propuesto por Ilkka Tuomi en [65]. En este modelo, los datos son una consecuencia de contar con un conocimiento, que se ha plasmado como información. Es decir, el proceso de generación de datos requiere pasar por las etapas 1: conocimiento, 2: información y 3: datos. Estas etapas son consecuentes con el desarrollo del Portal de Historia Parlamentaria, en el cual las personas encargadas de recoger la historia política ya habían traspasado gran parte de su conocimiento en textos biográficos y habían llegado a la tercera etapa de marcar los datos existentes en las biografías mediante la sintaxis de plantillas de MediaWiki. De este modo, al ingresar a este proyecto, me encontré con que ya existían datos semi-estructurados codificados en las páginas del wiki que usaba la BCN como una de las fuentes de su Portal Parlamentario.

Es importante recalcar los beneficios de la estrategia seguida por la Biblioteca. Si se hubiera querido partir por almacenar datos sobre los parlamentarios con los cuales luego podría añadirse información a modo de textos, se habría corrido el riesgo de no contar con los modelos apropiados para la incorporación de datos y, los consecuentes costos que implica la creación y modificación de interfaces de captura de datos. De este modo, comenzar desde la información, era una estrategia de bajo costo en la implementación de software y en la capacitación de las personas que iban a subir la información.

El uso de las plantillas de MediaWiki estaba inicialmente orientado a la presentación de los datos mediante fichas al interior de las plantillas, por lo que el aporte al proyecto consistió en

[5]Sistema MediaWiki http://www.mediawiki.org
[6]Enciclopedia Colaborativa http://www.wikipedia.org

reconocer que dichas plantillas podrían ser modificadas para agregar RDFa al marcado, y de esta forma, lograr que los datos estén disponibles mediante un estándar de publicación y en el modelo de datos RDF.

Posteriormente a la presentación de la solución que aquí se va a describir, el proyecto quedó supendido debido al anuncio de cambiar el sistema de plantillas actual de MediaWiki, que terminó en convertirse en un lenguaje demasiado complejo, por un sistema de plantillas basado en el lenguaje Lua. Se pronostica que este nuevo sistema de plantillas estará disponible para la versión 1.19 de MediaWiki [86]. Junto a la propuesta de usar las plantillas de MediaWiki para generar el RDFa, se identificaron los datos y vocabularios que podrían usarse, modelamiento que fue continuado por Francisco Cifuentes Silva, quien ha creado las versiones 1.1 y 1.2 que se encuentran publicadas en el portal de la datos de la BCN [87].

4.4.2. MediaWiki y la Web Semántica

La publicación de datos en la plataforma MediaWiki ha despertado el interés de la comunidad de datos enlazados por la gran cantidad de información contenida en la Wikipedia y por la gran cantidad de wikipedistas tras ella. Así, han surgido los proyectos de DBpedia[7] y SemantiMediaWiki[8]. El primero es un proyecto que extrae información de la Wikipedia en inglés para publicarlo en el modelo RDF a través de un SPARQL endpoint (base de datos para el modelo RDF). El segundo, es un módulo de MediaWiki que permite establecer relaciones entre los artículos de una wiki y provee de un lenguaje de consulta para usar sobre la información codificada en dichas relaciones. Además, otro módulo de MediaWiki mapea y publica la información de SemanticMediaWiki usando RDFa.

En el proyecto de la BCN se exploró una tercera alternativa, diseñada originalmente para su uso en la publicación de datos de historia parlamentaria de la Biblioteca del Congreso Nacional, pero que podría ser aplicable a otras publicaciones que utilicen MediaWiki. Esta tercera alternativa se basa en el uso del sistema de plantillas que provee MediaWiki [88][9].

El sistema de MediaWiki permite definir visualizaciones de datos usando plantillas. Una plantilla es simplemente otra página de un wiki cuyo nombre (o path-info) comienza por "/Tem-

[7]DBpedia http://dbpedia.org
[8]SemanticMediaWiki http://semantic-mediawiki.org/
[9]La razón de por qué se decidió utilizar las plantillas en vez del módulo de SemanticMediaWiki para el caso estudiado es detallado en la sección 4.4.4.

plate:", por ejemplo, una plantilla puede ser un documento llamado "/Template:saludos" y que contiene el texto: "¡Saludos a todos!".

Para llamar esta plantilla desde otra página del wiki basta con usar el sufijo de su nombre encerrado en un par de llaves: "{{saludos}}, bienvenidos a este wiki". Con ello, esta página será visualizada finalmente como: "¡Saludos a todos!, bienvenidos a este wiki".

La codificación y visualización de datos a través de las plantillas es posible gracias a que las plantillas se pueden invocar con parámetros.

```
{{Persona
 | apellido paterno = Muñoz
 | apellido materno = D'Albora
 | nombres = Adriana
}}
```

En el código anterior se llama a la plantilla "Persona" con los parámetros de apellido paterno, apellido materno y nombres. El ejemplo es un fragmento de lo que se publica actualmente en el portal de Biografías Parlamentarias. Para poder procesar los parámetros con los que se llama la plantilla, éstos pueden ser identificados en ella encerrándolos con tríos de llaves.

```
{| class="infobox"
 |-
 | <b>Nombre</b>
 | {{{nombres}}} {{{apellido paterno}}} {{{apellido materno}}}
 |-
|}
```

El primer paso de la generación de las vistas es la evaluación de los parámetros de las plantillas y, recursivamente, las plantillas que sean llamadas dentro de estos parámetros. Así, el resultado de la evaluación de las plantillas de este ejemplo será:

```
{| class="infobox"
 |-
 | <b>Nombre</b>
 | Adriana Muñoz D'Albora
 |-
|}
```

Este es un código de MediaWiki, que será evaluado para generar el fragmento de HTML final:

```
<table class="infobox">
 <tr>
  <td><b>Nombre</b></td>
  <td>Adriana Muñoz D'Albora</td>
 </tr>
</table>
```

Para introducir semántica en la publicación anterior basta con redefinir la plantilla de la siguiente manera:

```
{| class="infobox" about="foaf:Person"
 |-
 | <b>Nombre</b>
 | <span property="foaf:givenName">Adriana</span>
   <span property="foaf:familyName">Muñoz D'Albora</span>
 |-
|}
```

El llamado de plantillas permite definir datos semi-estructurados mediante la anidación de plantillas, es decir, definiendo el valor de unos parámetros como función de resultado de otras. En el siguiente ejemplo se muestra cómo se puede definir el valor del parámetro parlamentaria (correspondiente a la historia parlamentaria de una persona) a través del resultante de la invocación de otras plantillas.

```
{{Persona
 | apellido paterno = Muñoz
 | apellido materno = D'Albora
 | nombres = Adriana
 | parlamentaria =
   {{parlamentario
    | final = 2014
    | división = Distrito N° 9
    | enlace = distrito09
    | tipo = Diputado
    | inicio = 2010
    | partido = Partido por la Democracia
   }}
   {{parlamentario
    | final = 2010
    | división = Distrito N° 9
    | enlace = distrito09
    | tipo = Diputado
    | inicio = 2006
    | partido = Partido por la Democracia
   }}
}}
```

El lenguaje de plantillas de MediaWiki es aún más flexible dado que permite la evaluación de funciones y posee estructuras de control como el "`#if`", que posee la sintaxis "`{{\#if:=c|x|y}}`". Los parámetros de la sintaxis de "`#if`" son: una cláusula a evaluar (`c`), lo que imprime `x` cuando `c` es verdadera e `y` en el caso contrario. Por otra parte la notación "`{{{x|y}}}`" imprime el valor de `x` si está definido e `y` en el caso contrario. Así, las estructuras de control pueden ser usadas, por ejemplo, como en la siguiente plantilla para distinguir entre personas con y sin valor en el campo de carrera parlamentaria:

```
{| class="infobox"
 |-
 | <b>Nombre</b>
 | {{{nombres|}}} {{{apellido paterno|}}} {{{apellido materno|}}}
 |-
   {{#if:{{{parlamentaria|}}}
    | {{!}} colspan=''2'' class=''cabecera''
      {{!}} '''Trayectoria Parlamentaria'''
      {{!}}-
```

```
10        {{{parlamentaria|}}}
11        {{!}}-
12     }}
13  |}
```

4.4.3. Identificación de recursos

Un error común en la publicación de datos mediante RDF es confundir entre recursos ideales y las páginas donde se los describe[10]. En el caso del portal parlamentario se habría cometido un error si a cada parlamentario se lo hubiera identificado mediante el nombre de la página de la wiki que lo describía. Para evitar este problema se utilizó la estrategia de utilizar fragmentos identificadores dentro del documento. De este modo, si se quería identificar al parlamentario Manuel De Salas Corbalán, la URI sería `/Manuel_De_Salas_Corbalán#Persona`[11].

Siguiendo este mismo patrón otros identificadores fueron creados como secciones dentro de la página. Así, la instancia donde Manuel De Salas asumió como diputado fue identificada como `#Diputado-1831`. Agregar el año es importante porque una misma persona puede asumir en más de una ocasión el cargo de diputado y cada una de ellas es identificable, dentro de una persona, por el año. Con este esquema se siguen varias de las prácticas tratadas en la sección 6: se usan nombres de clases o colecciones a las que pertenecen los recursos identificados (práctica 6.1.2), el período de ejercicio parlamentario como un sufijo de la página que habla del parlamentario sigue la práctica de generar identificadores jerárquicos (sección 6.1.3) y el año como parámetro para distinguir el período sigue la práctica de usar llaves que distingan los recursos como parámetros de las URIs (práctica 6.1.4).

4.4.4. ¿Por qué no se usó Semantic Media Wiki?

SemanticMediaWiki es una extensión de MediaWiki que hace uso de la sintaxis de categorías provista por MediaWiki para incorporar información semántica. No se trata de marcado utilizando RDF, sino que se utiliza un modelo propio para codificar relaciones entre los artículos de un wiki y permitir consultas semánticas. Dado que la información semántica que

[10] En la sección 6.1.1 se describirá con más detalle cómo evitar este error.

[11] Omitimos la base de esta URI porque las biografías sólo son usadas como un backend más para el actual portal de Historia Política. Actualmente cada página del wiki es tomada y procesada para ser mostrada con el prefijo `http://historiapolitica.bcn.cl/resenas_parlamentarias/wiki`.

se marca con SemanticMediaWiki sólo es accesible mediante las consultas provistas por dicha extensión, existe otra extensión de WikiMedia que expone la información utilizando RDFa[12].

La notación utilizada por SemanticMediaWiki extiende la notación de categorías. Por ejemplo, el siguiente código describe la ciudad de Berlin[13]:

```
[[capital of::Germany]]
[[locate in::Europe]]
[[has LatLong::52.5166667,13.4]]
[[Category:Location]]
```

Al encontrarse este código en la página que describe a Berlín se genera el siguiente código RDFa:

```
<div id="RDFa"
     about="http://localhost/mediawiki/index.php/Berlin"
     xmlns:wiki_1="http://localhost/mediawiki/index.php/"
     xmlns:wiki_1_property="http://localhost/mediawiki/index.php/Property:"
     xmlns:wiki_1_category="http://localhost/mediawiki/index.php/Category:"
     typeof="wiki_1_category:Location">
  <a href="wiki_1:Germany" rel="wiki_1_property:Capital_of"></a>
  <div property="wiki_1_property:Has_LatLong"
       content="52.5166667,13.4"></div>
  <a href="wiki_1:Europe" rel="wiki_1_property:Locate_in"></a>
  <div property="wiki_1_property:Modification_date"
       content="3 September 2009 02:19:03"></div>
</div>
```

Podemos observar que en este código hay expresadas tripletas que describen tanto propiedades del recurso (literales) como relaciones con otros recursos. De este modo, si ponemos también `[[Is capital of::Chile]]` en la página de Santiago introduciremos la tripleta (*Santiago*, *Is capital of*, *Chile*) que relaciona la página de "Santiago" con la de "Chile" a través del predicado "Is capital of".

Lo anterior tiene un problema: SemanticMediaWiki fue desarrollado para codificar información semántica entre los artículos de un wiki y no para su uso con el modelo RDF. La

[12]RDFa es una extensión de MediaWiki desarrollada por Jin Guang Zheng y Jie Bao que dejó de ser mantenida en 2010 que sirve como herramienta para generar marcado RDFa sobre SemanticMediaWiki (http://www.mediawiki.org/wiki/Extension:RDFa).

[13]Este ejemplo fue tomado textualmente de la documentación del complemento RDFa para MediaWiki.

extensión que se hace con el módulo RDFa utiliza la misma URI para describir la página que describe al recurso y para describir al recurso ideal, tema que es tratado en la sección 6.1.1.

El uso de RDFa en SemanticMediaWiki requiere hacer un mapeo de la lógica de artículos y relaciones a objetos y predicados identificados por URIs. Para que la relación "Is captital of" sea mapeada a una URI se debe ingresar una entrada para ella en el archivo de configuración del mapeo que provee el módulo RDFa de MediaWiki.

Otra limitación de SemanticMediaWiki es que no permite crear datos anidados. La trayectoria parlamentaria de una persona posee varios recursos describiendo cada una de las ocasiones en las que ejerció como parlamentario. Con el uso de las plantillas era posible incorporar estos recursos dentro del artículo del parlamentario. En cambio, en la notación de categorías usada por SemanticMediaWiki el sujeto es siempre el artículo que se está editando, lo que impide agregar tripletas cuyo sujeto sea un recurso anidado. La única posibilidad para agregar un atributo a estos recursos anidados sería desanidarlos creando una página para cada uno de ellos.

En resumen, SemanticMediaWiki sólo permite dos tipos de tripletas:

1. <página actual> <relación> <otra página>
2. <página actual> <relación> "literal"

Esto nos dificulta seguir alguna de las dos estrategias definidas en la práctica 6.1.1 y en particular poder describir más de un recurso en una misma página, tal como se hizo en la publicación de las biografías parlamentarias (ver sección 4.4.3).

4.4.5. Conclusión

El uso de plantillas permite introducir datos semi-estructurados que luego pueden ser mapeados directamente a código HTML y, por consiguiente, a una representación en el modelo RDF de las tripletas utilizando el marcado RDFa. Además, las estructuras de control y otros elementos de MediaWiki poseen expresividad extra que permiten cubrir todas las necesidades del marcado de datos mediante RDFa. Sin embargo, si bien el trabajo realizado ha demostrado la factibilidad de esta estrategia, los cambios proyectados en el sistema de plantillas de MediaWiki [89] nos obligan a reestudiar su aplicación.

Capítulo 5

Prácticas generales

En esta sección se presentará un conjunto de buenas prácticas para la publicación de datos en general, es decir, de manera independiente a las tecnologías usadas en la publicación.

Los ocho principios de los datos abiertos (ver sección 2.3.3) nos permiten distiguir cuando un conjunto de datos es abierto. En este capítulo propondremos un modelo conceptual definido por cinco principios que nos permitirán distinguir cuando un conjunto de datos ha sido publicado con el fin de promover el uso abierto. A su vez, estos principios servirán de guía para definir las buenas prácticas de publicación que se presentarán en este capítulo.

1. *Acceso:* Los datos deben ser publicados en documentos y sistemas que estén accesibles, para la descarga, su uso y su referenciación.

2. *Formato:* Los datos deben encontrarse en formatos que permitan operar con ellos de manera automática.

3. *Confianza:* Los usuarios deben poseer medios que les permitan decidir cuánta confianza tendrán en la veracidad de los datos y en qué contextos éstos podrán ser usados.

4. *Preservación:* Los datos deben ser preservados tanto en el contenido como en las referencias que permiten accesarlos.

5. *Redistribución y remix:* Se debe permitir a los usuarios la redistribución de los mismos datos y de nuevos datasets derivados.

Los principios tienen una estrecha relación con las libertades del Software Libre presentadas en la sección 2.3.1. El principio de acceso es similar a la libertad 0, de usar el software; los principios de formato y confianza se asemejan a la libertad 1, de estudiar cómo el código funciona; mientras que el quinto principio equivale a las libertades 2 y 3. De los principios propuestos el único que no tiene par en el Software Libre es el principio de preservación. En el Software Libre uno tiene la libertad de preservar el software, pero no la obligación. En cambio, en relación a los datos abiertos debiera ser una obligación de las instituciones públicas hacerse cargo de la preservación de los datos como parte de los bienes comunes de la sociedad.

5.1. Acceso

Usar la Web Dentro del contexto actual, resulta impensable hablar de acceso a los datos sin pensar en la Web. De este modo, cualquier plataforma de publicación de datos debe considerar la descarga de los datasets mediante el protocolo HTTP y la interacción con ellos mediante sistemas que provean lenguajes de consulta o APIs Web apropiadas.

Identificar Para la publicación de datasets es también un requisito la identificación de ellos mediante URIs que permitan su referenciación. De este modo, los datos podrán ser citados y accedidos a través de estas citas. Garantizar el derecho a enlazar crea un ecosistema en el que las personas podemos compartir la información que nos resulta relevante y nuestras opiniones sobre ella.

Describir Publicar datasets sin añadir métodos para llegar a ellos es similar a la publicación de información en un sistema de registro (ver sección 3.1). Por ello resulta esencial describir los datos publicados mediante sus metadatos. Además, se debiera respetar un conjunto mínimal común de metadatos, que pueda facilitar la interoperabilidad de las búsquedas (por ejemplo Dublin Core[1] y DCat [67]).

Navegación temática Al igual que la descripción de los datos mediante metadatos, es necesario permitir la navegación temática por ellos. Esto significa que no basta con poner

[1] http://dublincore.org

todos los datasets en un mismo cajón, sino que también resulta útil contar con selecciones temáticas de ellos. Además, dentro de sistemas de gestión de datasets es recomendable utilizar tanto taxonomías creadas con vocabularios controlados como folcsonomías[2].

Portal integrador En el otro extremo de la navegación temática, los usuarios deben contar con la posibilidad de conocer la existencia de "todos" los datos que los organismos públicos han hecho disponibles. De lo contrario, la búsqueda de información puede ser una tarea demasiado compleja. En el contexto de los sistemas documentales el problema de la integración es resuelto mediante protocolos de *harvesting*, como el OAI (ver sección 2.5). Es necesario que el gobierno cuente con al menos un portal que posea punteros hacia todos los datos que se están publicando, ya sean datasets descargables o bases de datos.

Publicación oportuna Es necesario que los datos sean publicados de manera oportuna para que se garantice su valor. Eso implica publicarlos en primera instancia de la manera más sencilla posible, luego podrán publicarse versiones mejoradas. A menudo la publicación oportuna enfrenta barreras: la poca claridad sobre si los datos compremeten la privacidad de personas, la transparentación de hechos inconvenientes para quienes retienen los datos y los embargos en las investigaciones científicas (períodos en los que los investigadores tienen la primicia sobre los datos).

Herramientas de búsqueda Para poder acceder a los datos publicados es necesario que se provea de herramientas de búsqueda sobre ellos, que podrán basarse en sus contenidos o en sus metadatos. Estas herramientas de búsqueda deben ser adecuadas para el ámbito de los datos publicados.

5.2. Formatos

Usar formatos abiertos Para que todos puedan usar los datos éstos deben codificarse en formatos abiertos. De lo contrario, no se puede garantizar el acceso en el tiempo ni la libertad en la creación de aplicaciones que usen los datos.

[2]Las folcosnomías son vocabularios no controlados definidos de manera colaborativa y generalmente sin establecer relaciones entre los términos.

Usar identificadores globales En la interpretación de los datos como hechos o afirmaciones realizadas sobre recursos, el uso de identificadores globales sobre los recursos facilita la integración de los datos.

Usar documentos autocontenidos Incorporar la metadata a los documentos o conjuntos de datos de modo que sea posible comprender los contenidos de un documento permite contextualizar los datatasets aunque éstos sean encontrados fuera de un catálogo.

Microdatos Publicar los datos en su forma primaria (*raw data*), es decir, al mismo nivel de granularidad de la fuente, permite que éstos sean usados para responder preguntas distintas a las que les dieron origen, usar diferentes metodologías o validar la correctitud de los resultados de dichas agregaciones. La única justificación para no publicar los datos en su forma es el secreto estadístico en el que metodologías de agregación como las presentadas en la Sección 2.4 pueden usarse para entregar datos en niveles de desagregación que busquen un balance "apropiado" entre la privacidad y la utilidad de los datos.

5.3. Confianza

Declarar error Especialmente en datos que son inciertos, es necesario declarar el error que se estima en las observaciones, por ejemplo, introduciendo el error de los instrumentos utilizados.

Vigencia Los datos tienen validez en el tiempo. Por ejemplo, la afirmación "x preside la organización y" es valida sólo durante un período de tiempo. Describir el rango de validez en los datasets puede ayudar a personas o sistemas automatizados a no sacar conclusiones erróneas desde datos que ya no tienen vigencia.

Linaje Conocer de dónde proceden los datos es crucial en relación a la confianza que tendremos sobre ellos en un mundo donde se entremezclan contenidos de buena y mala calidad [90]. John Sheridan afirma que "el linaje[3] es el desafío número uno al que nos enfrentamos cuando

[3]En inglés al linaje se le conoce como *provenance*.

publicamos datos como datos enlazados en data.gov.uk" [91]. Además, el linaje es un requisito para permitir que las investigaciones científicas sean reproducibles y por ende es necesario codificarlo y crear punteros persistentes desde las derivaciones a sus originales [92]. Citar es un requisito para la curación [93]. Para Luc Moreau, codificar y usar el linaje es un problema cuya múltiplicidad de dimensiones queda de manifiesto en las 425 publicaciones sobre el tema publicadas hasta 2009, entre las cuales la mitad son de los últimos dos años [94].

Entre los vocabularios que suelen ser usados para agregar información relacionada con el linaje, en distintos grados de completitud, se encuentran: *Dublin Core Metadata Terms* [95], *Dublin Core Metadata Elements (legacy)* [96], *Friend of a Friend* (FOAF) [97], *Semantically-Interlinked Online Communities* (SIOC) [98], *Semantic Web Publishing Vocabulary* [99], *Web Of Trust RDF Ontology* (WOT) [100], *Prof Markup Language* (PML) [101, 102], *Ouzo Provenance Ontology* [103], *Changeset Vocabulary* [104], *Open Provenance Vocabulary* [105] y *Open Provenance Model* [106]. Varios de los vocabularios se basan en modelos abstractos de linaje, como el propuesto por Olaf Hartig [107] o el propuesto por *Open Provenance Group* [108]. De este último derivan un esquema para XML [109], un vocabulario para RDF [105] y una ontología OWL [110].

En relación a cómo aplicar los vocabularios de linaje a conjuntos de datos expresados en RDF, una idea que gana fuerza es considerar el conjunto de datos como un grafo y utilizar el nombre del grafo (una URI) en sentencias RDF que definen el linaje del conjunto de datos. Así, si además se ingresan los datos que caracterizan al conjunto de datos dentro de él mismo, se logra que el conjunto de datos sea autocontenido [111]. Además, es posible serializar un conjunto de grafos [112] y con ello crear documentos que integren otros conjuntos de datos e incluyan el provenance de los triples incluidos. No obstante, aunque gran parte de las implementaciones de SPARQL soportan los grafos con nombre, actualmente no existe forma de codificarlos en lenguajes como RDF/XML y RDFa. Nuevos formatos para RDF que soportan grafos han surgido: TriG (Turtle más grafos con nombre), TriX y N-Quads.

Otra propuesta para el uso de RDF es la de extender el lenguaje de consulta SPARQL para lidiar con los temas de linaje, este lenguaje extendido se llamaría tSPARQL [113].

5.4. Preservación

Preservar las fuentes En muchos casos los datos son generados de documentos que deben ser preservados. Por ejemplo, en el caso de las biografías parlamentarias los datos son extraídos de la documentación biográfica de los parlamentarios.

Prácticas de preservación Las prácticas habituales de la preservación consisten en la creación de respaldos y en la replicación de éstos en localizaciones diversas que minimicen el riesgo de que puedan destruirse simultáneamente.

Identificadores persistentes La creación de identificadores a nivel de documentos permite poder referenciarlos y dereferenciarlos, mientras, a nivel de datos, los identificadores pueden describir tanto a objetos de información como objetos ideales. En cualquiera de las dos situaciones, los identificadores deben ser diseñados para mantenerse estables y de ese modo permitir la integración de los datos y evitarnos el trabajo de corregir enlaces rotos.

5.5. Redistribución y remix

Declarar cambios Es necesario declarar los cambios de los datos o los cambios en el catálogo para que los usuarios puedan replicar el catálogo en tiempo real, sin necesidad de descargar los recursos que no han sufrido cambios.

Acceso al catálogo Permitir que los usuarios puedan descargar el catálogo de modo que éste pueda ser procesado de maneras que no hayan sido previstas o que no se hayan implementado por ser demasiado específicas.

Licencia libre A pesar de que resulta criticable que un dato pueda ser considerado una obra susceptible de ser licenciada, resulta común que en el mundo los datos sean publicados bajo licencias. En el caso de que el marco jurídico acepte los datos como objetos licenciables lo recomendable es que estas licencias no impidan el uso, combinación y redistribución de los datos. No basta con usar una licencia abierta en el sentido de la *Open Knowledge Definition*

[114], sino que, tal como se presenta en el cuadro 5.1, es recomendable escoger una licencia de acuerdo al tipo de objeto a licenciar.

Licencia	Dominio	By	SA	Comentarios
Creative Commons Attribution	C	Si	No	
Creative Commons Share-Alike	C	Si	Si	
Creative Commons CCZero	C+D	No	No	
GNU Free Documentation License	C	Si	Si	Sólo satisface la OKD bajo ciertas condiciones.
UK PSI Public Sector Information	C+D	Si	No	
Free Art License	C	Si	Si	
MirOS License	S+C	Si	No	
Open Data Commons Public Domain Dedication and Licence (PDDL)	D	No	No	Dedicada al dominio público, todos los derechos son cedidos.
Open Data Commons Attribution License	D	Si	No	Atribución para bases de datos
Open Data Commons Open Database License (ODbL)	D	Si	Si	Atribución-compartir igual para bases de datos
Creative Commons Zero (CC0)	C+D	No	No	Dedicada al dominio público, todos los derechos son cedidos.

Cuadro 5.1: Licencias y tipos de uso sugeridos por la *Open Knowledge Foundation* (http://opendefinition.org/licenses). Para cada una de las licencias se especifica el dominio (C: contenido, D: datos o S: código), si requieren atribución (By) y si requieren que se comparta en los mismos términos (SA).

Fomentar la participación Uno de los aspectos fundamentales de éxito de la publicación de datos públicos es la participación de la sociedad civil en el uso y desarrollo de aplicaciones sobre los datos. En el estudio realizado por Becky Hogge en mayo de 2010, Open Data Study [115], se afirma que "Hay tres partes involucradas en empujar data.gov y data.gov.uk: la sociedad civil, los profesionales de la administración pública y las lideres políticos; sin

cualquiera de estas tres capas no estaríamos donde nos encontramos ahora". Las prácticas observadas para promover la participación civil son:

1. Generar canales de comunicación para recibir ideas provenientes de la sociedad civil.

2. Permitir comentarios en las fichas de los conjuntos de datos.

3. Permitir la evaluación de los conjuntos de datos por la sociedad civil.

4. Favorecer o fomentar la generación de comunidades, ya sea soportando recursos para la comunicación de éstas o haciendo publicidad a las comunidades e iniciativas civiles existentes.

5. Organizar competiciones de desarrollo de aplicaciones que usen los datos.

6. Catalogar las aplicaciones que usan los datos, tanto las creadas dentro del gobierno como las desarrolladas por la sociedad civil. También incluir las aplicaciones que usan los datos en la metadata de las fichas de los conjuntos de datos.

Capítulo 6

Prácticas de RDF

Este capítulo presenta una serie de prácticas para ser aplicadas en la publicación de datos con el modelo RDF. Las prácticas han sido agrupadas en dos secciones. La primera de ellas describe prácticas referentes a cómo identificar los recursos y, la segunda, a cómo modelar los datos que serán publicados.

6.1. Prácticas de identificación

Uno de los aspectos fundamentales de RDF es el uso de identificadores para referirse a los recursos descritos. Los identificadores URIs son también una parte fundamental de la Web, junto con el protocolo HTTP y el lenguaje HTML (ver la sección 2.1). En la Web los identificadores son usados como medio para acceder a recursos, aunque en algunos casos no resulta claro cuál es el recurso referido. El concepto del objeto referido o identificado es aún más difuso cuando las URIs se utilizan para referirse a objetos del mundo real. David Both aborda este problema desde la perspectiva de que tal identificación en sí misma es un concepto sin un significado preciso, pero que resulta útil cuando se tiene claro el uso que se le quiere dar [116]. Dicho de otro modo, el problema no es lo que las URIs representan, sino para qué se usan. Es necesario considerar esta problemática en un contexto en el cual quienes publican, los usos y sus visiones son diversas. Dos problemas derivados de esta diversidad son las colisiones de identificadores y el garantizar la permanencia de los identificadores. Entre las colisiones de identificadores existen dos tipos: *a*) dos identificadores para un mismo

objeto ideal y *b*) dos objetos ideales para un mismo identificador. Las prácticas de este grupo tendrán como finalidad principal resolver estas dos problemáticas. Además de estos problemas, la estructura de las URIs, si bien son tratadas como opacas, pueden ayudarnos a comprender a qué corresponden los objetos referenciados [117].

6.1.1. Diferenciar recursos de páginas donde se los describe

Cómo diferenciar un recurso que es descrito en un documento, del documento que lo describe.

Uno de los problemas clásicos de la publicación de datos enlazados es la confusión entre recursos que representan a documentos y recursos que representan a entidades ideales. El problema ocurre, por ejemplo, cuando se quiere identificar una persona con la URI de su página web. Tal como se describe en [118], existen dos estrategias para no caer en este error. La primera consiste en utilizar fragmentos identificadores (lo que va en una URI luego del símbolo `#`) y, la segunda, en redirigir desde la URI A que representa a un recurso ideal B, que contendrá la descripción de A.

La primera estrategia tiene la ventaja de que nos evita tener que implementar el redireccionamiento para todos los recursos ideales hacia los documentos que describen información de ellos. Además, puede ser también útil cuando se requiere describir describir e identificar varios recursos en un mismo documento. En el desarrollo del proyecto de las biografías parlamentarias de la Biblioteca del Congreso se usó esta estrategia, pues permitía implementar, sin modificaciones a la plataforma MediaWiki, URIs asociadas a los recursos referidos mediante los fragmentos de identificadores. De ese modo las `#persona` y `#senador-2010` dentro de una página permitían identificar a la persona descrita en esa página y al cargo de senador que asumió el año 2010, sin entrar en conflicto con identificadores encontrados en otras páginas.

La segunda estrategia es usada por la DBpedia al, por ejemplo, usar la URI `http://dbpedia.org/resource/Santiago_Province,_Chile` para identificar al objeto ideal de la Provincia de Santiago, que al ser desreferenciada produce una redirección hacia la URI `http://dbpedia.org/page/Santiago_Province,_Chile`, que sí contiene la descripción del recurso referenciado. Seguir esta estrategia implica crear un redireccionador que cambie `resource` por `page`. Además, requiere que cada recurso tenga su propio documento donde es descrito.

En el caso de UK se definió una recomendación para crear URIs de objetos ideales en base a dos patrones posibles [119, 120] (ver Cuadro 6.1).

1. `http://{sector}.data.gov.uk/id/{concepto}/{identificador}`
2. `http://{sector}.data.gov.uk/id/{concepto}#{identificador}`
3. `http://{sector}.data.gov.uk/doc/{concepto}/{identificador}`
4. `http://{domain}/doc/{concepto}/{referencia}/{doc.extension}`
5. `http://{domain}/def/{concepto}`
6. `http://{domain}/doc/{concepto}`
7. `http://{domain}/set/{concepto}`

Cuadro 6.1: Patrones para identificadores en UK. Los patrones 1 y 2 corresponden a identificadores de recursos ideales; 3 a documentos; 4 a documentos descargables; 5 a documentos definiendo conceptos; 6 donde se listan instancias de un concepto y 7 a conjuntos.

Es importante notar que esta diferenciación sólo se hacen en el caso de los recursos ideales, es decir, entre los patrones 1 y 2 del Cuadro 6.1.

En el caso de la Biblioteca del Congreso Nacional de Chile también se usa la estrategia de redireccionar para distinguir entre las normas y los documentos que contienen su descripción. Por ejemplo, la Ley Nº 20.285, sobre el acceso a la información pública, es identificada por la URI `http://datos.bcn.cl/recurso/cl/ley/ministerio-secretaria-general-de-la-presidencia/2008-08-20/20285`. Al desreferenciar esta URI se nos redirige inmediatamente a la misma URI anterior, pero con el sufijo `datos.html`.

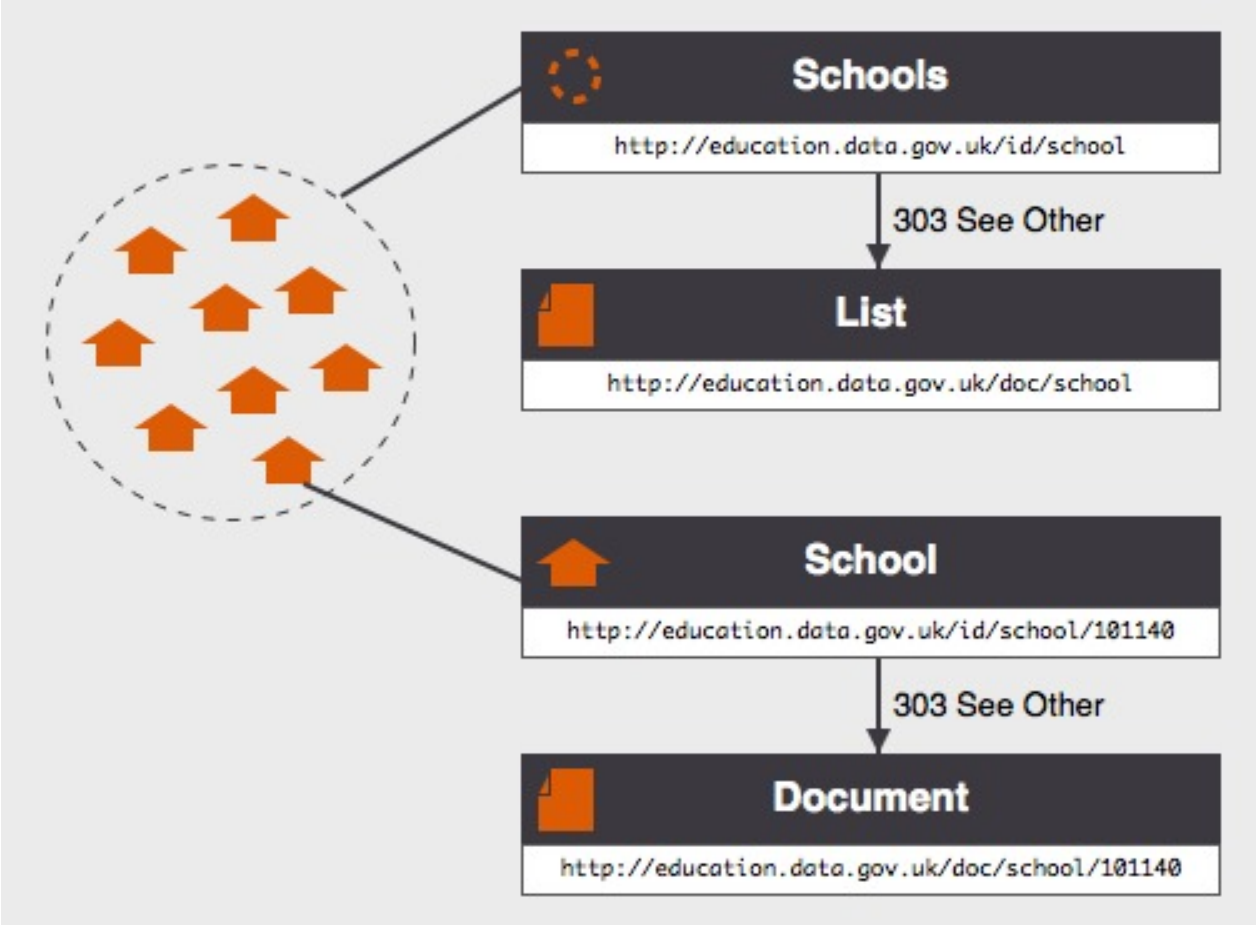


Figura 6.1: Redirecciones entre recursos ideales y documentos en borrador estándar de UK para URIs [119].

6.1.2. Incluir nombres de clases en identificadores

Cómo publicar colecciones de elementos de una misma clase.

En algunos casos los recursos que describimos pueden agruparse en colecciones disjuntas. En estos casos puede resultar útil incluir el nombre de las colecciones en las URIs de los recursos. Ello puede ser útil para entender a priori lo que representa el recurso.

En esta línea, el Gobierno de UK ha definido políticas específicas para la creación de identificadores, que incluyen la incorporación de los conceptos que agrupan los identificadores (ver cuadro 6.1). Por ejemplo, para un colegio el identificador podría ser:

`http://education.data.gov.uk/id/school/123065`

Este mismo protocolo también está siendo utilizado en la Biblioteca del Congreso Nacional de Chile, donde además se recomienda usar el singular [121]:

> **URIs de instancias**
>
> Para los recursos que describan instancias de alguna clase se agregan las siguientes pautas:
>
> Uso siempre del singular, ejemplo: "persona/1234" en lugar de "personas/12345"
>
> **URIs de listas**
>
> Para los recursos que describan listas de alguna clase se agregan las siguientes pautas:
>
> Uso siempre del singular pero sin un identificador, por ejemplo, para obtener el listado de personas se deberá acceder a: "persona" en lugar de "personas"

6.1.3. Crear identificadores jerárquicos

Cómo crear identificadores aprovechando estructuras jerárquicas.

En varios casos los recursos se organizan de modo naturalmente jerárquico, tal como los elementos de un libro o las secciones de una organización. En muchos de estos casos, usar identificadores construidos de manera jerárquica entrega información a quienes los leen y facilita la creación de ellos. Por ejemplo, en un libro se puede definir un patrón de URIs como el siguiente:

```
http://.../book/123/chapter/2/section/4/page/12
```

Si bien en algunos casos esta estrategia de construcción de identificadores puede resultar útil, hay que tener cuidado con objetos cuya jerarquía es susceptible de cambiar. Así, por ejemplo, en las estructuras orgánicas de los organismos un área podría moverse de un departamento a otro, por lo que si se mantiene la URI esta nomenclatura podría resultar engañosa. De igual modo, la ciudad de Valdivia se encontró en un momento incluida en la Región de los Lagos, pero ahora se encuentra en la Región de los Ríos. No obstante, conceptualmente puede ser

que prefiramos pensar que dicha ciudad conforma una unidad lógica, es decir, no deseamos pensar que se trata de dos ciudades diferentes.

6.1.4. Crear identificadores desde llaves o parámetros

Cómo crear URIs a partir de identificadores en otros formatos o de palabras que pueden identificar los recursos.

En muchos casos los recursos poseen atributos que nos permiten identificarlos. Por ejemplo, las leyes tienen sus números, las comunas y las regiones tienen sus nombres, los eventos tienen fechas, etc. Todos estos elementos nos permiten crear identificadores, definiendo patrones paramétricos para la familia de URIs asociada a una colección de recursos. Por ejemplo, en la sección 4.4.3 se describe cómo los períodos de ejercicio de parlamentarios se definieron utilizando como parámetros el cargo y el año de inicio:

```
#{cargo}-{año-inicio}
```

De este modo, el cargo de un diputado que se inicia el año 1831 será identificado dentro de la página por `#Diputado-1831`.

6.1.5. Crear identificadores proxy

Cómo crear URIs sobre recursos que pertenecen a terceros y que no están construidos de manera estándar.

Un ejemplo de objetos que no son identificados de manera estándar son los formatos de recursos en Internet (Internet Media Types) [122]. No existe una manera estándar de representar estos identificadores [123]. Existen propuestas que no se han estandarizado, tales como el servicio que se ofrece en http://mediatypes.appspot.com. Ante tal falta de estandarización, una solución es utilizar URIs que nos permitan, en el caso de establecerse un estándar, relacionarse con los identificadores propuestos por éste. Lo que se busca de este modo es publicar datos que sean consistentes de manera interna y, a su vez, permitir que haya una estructura que facilite una posterior interrelación con otros conjuntos de datos.

Por ejemplo, se pueden crear identificadores locales para los formatos y usarla del siguiente modo:

```
ex:anImage a foaf:Image;
  dc:format <http://.../media-types/image/jpeg>
```

Más adelante, estos identificadores pueden ser relacionados con estándares globales.

6.1.6. Compartir llaves de URIs

Cómo compartir llaves de URIs entre distintas publicaciones.

Cuando existe una fuente de recursos estable (las URIs asociadas a los recursos no cambian) y definida en base a llaves, es recomendable usar las mismas llaves para crear URIs que representen los recursos que allí se presentan. Por ejemplo, la IMDb[1] publica la información de la película *Nosferatu: Phantom der Nacht* con la URI http://www.imdb.com/title/tt0079641/. Si quisiéramos basarnos en esa URI para construir nuestra propia URI para identificar la película (recordemos de la sección 6.1.1 la diferencia entre URIs para páginas y para recursos ideales) podremos utilizar la llave tt0079641 como parte de ella. De esta forma, un programador que requiera utilizar ambas fuentes no necesitará acceder a consultas SPARQL para poder interrelacionar ambos recursos.

[1] Internet Movie Database: Servicio que publica información sobre películas (http://www.imdb.com).

6.2. Prácticas de modelamiento

En su núcleo, RDF está basado en el modelo Entity-Attribute-Value (EAV) y, particularmente, en su extensión, el modelo EAV/CR, que son principalmente utilizados en la construcción de registros médicos y en otra información científica [23]. Estos modelos almacenan tripletas compuestas de una entidad, un atributo y un valor. En el modelo EAV las dos primeras componentes de las tripletas son siempre identificadores, mientras que la tercera corresponde a un literal. En cambio, el modelo EAV/CR permite también incorporar identificadores de otras entidades como valores de las tripletas, lo que establece relaciones entre objetos por medio de llaves foráneas.

El modelamiento en RDF ha adoptado muchas de las técnicas que se usan en el modelamiento orientado a objetos, debido a que los lenguajes de modelamiento RDF Schema y OWL hacen uso extensivo de la noción de clases y de la noción de atributos. De este modo, se ha inclinado la balanza hacia el concepto de clase por sobre el de prototipo. Esta disputa entre los modelos de clase y prototipo son discutidos desde sus aspectos filosóficos en [29].

Otra interpretación de las bases de datos RDF es que éstas corresponden a bases de predicados lógicos sobre los cuales se puede hacer inferencias y sobre los cuales se pueden definir conceptos mediante ontologías. Por ejemplo, la definición de subclases conduce a inferencias. Si A es subclase de B entonces cualquier elemento de A es también un elemento de B. Esto se expresa como la siguiente fórmula de inferencia.

$$\frac{x \in A \wedge A \subset B}{x \in B}$$

En el caso de la orientación de objetos, este tipo de inferencia es utilizado para determinar las propiedades que los objetos tienen, compactando el esfuerzo descriptivo al incorporar la definición de las propiedades de los elementos en un único lugar, arriba en la jerarquía de clases.

A diferencia del modelamiento en OOP, donde los modelos son desarrollados para resolver un problema específico mediante una aplicación, en el modelamiento RDF éstos suelen ser más generales, pues se busca que los vocabularios sean ampliamente reutilizados a lo largo de diversos datasets. Esta mayor necesidad de integración tiene por consecuencia que, en el compromiso entre reusar vocabularios o refinarlos, el reuso sea preponderante.

6.2.1. Usar vocabularios comunes

Cómo facilitar la integración entre datasets publicados por múltiples organizaciones y con múltiples sistemas.

Usar esquemas o vocabularios comunes que puedan ser extendidos para cubrir la temática de los datos publicados sin pérdida de información, es una práctica generalmente aceptada por la comunidad de la Web Semántica. En particular, Jeni Tennison recomienda [124]:

> El reuso de vocabularios existentes facilita la integración de diversos dominios en RDF, haciendo los datos más reusables. Por ejemplo, un mapeo del OPM que hace énfasis en el reuso de FOAF para modelar las personas y las organizaciones ahorra tiempo y esfuerzo para los desarrolladores del vocabulario RDF de OPM. De otro modo, modo se habría incurrido en un gasto de tiempo innecesario modelando los detalles de los agentes. Además, ello permite que todos los agentes que son descritos como el vocabulario de OPM estarán automáticamente disponibles como agentes de la más amplia nube de FOAF. Lo mismo ocurre con el uso de DOAP para describir software.

A pesar de que usar un vocabulario común es una herramienta poderosa para la integración, la inexistencia de un único modelo universal óptimo impide que ésto pueda hacerse en todos los casos. Antero Tailvasaari hace una resumen de los problemas filosóficos detrás de esta imposibilidad, en el contexto de una discusión frente a los modelos basados en prototipos versus los basados en clases [29]. En el trabajo de Tailvasaari se describen las dificultades que se han presentado en varios ámbitos, desde problemas mas difusos como el lenguaje natural (tener un único idioma común) hasta problemáticas como contar con una buena jerarquía de objetos.

> Otra implicación que deriva del modelo clásico Aristotélico y su adopción en los lenguajes orientados a objetos actuales es el hecho de que no hay una jerarquía de clases "óptima". Esto es fácil de observar a diario en el diseño e implementación de sistemas orientados a objetos. En muchas situaciones, una jerarquía de clases natural e intuitiva desde el punto de vista conceptual no es reusable, extensible o eficiente; la que sí lo es puede no resultar conceptualmente elegante; o la más eficiente no es ni elegante ni extensible. En general, el diseño de jerarquías de clases implica compromisos.

Como consecuencia de la imposibilidad de este modelo "óptimo" surgen dos problemáticas: *a*) la constante necesidad de rediseñar el software para que se adapte a nuevos requerimientos y *b*) la necesidad de integrar distintos sistemas. La primera de las dos problemáticas requiere de métodos y herramientas que nos faciliten el proceso de modificación del software y que han sido estudiadas por varios investigadores, incluyendo Bergstein [125], Casais [126] and Opdyke [127, 128]. Mientras, un área importante de la integración de sistemas es la transformación de datos de un modelo a otro, que ha sido estudiado y desarrollado por varios investigadores, como por ejemplo en [129] para el caso de los datos estructurados y con el desarrollo de lenguajes como XSL [130] o bXid [131] para datos semi-estructurados.

En RDF la forma natural de transformar conjuntos de triples es aplicar reglas de inferencia. Estas tienen dos usos: *a*) traducir un grafo RDF G a otro G', codificado en un vocabulario alternativo, y *b*), dada una base definida por un grafo G y un conjunto de reglas de inferencia R, hacer consultas sobre G_R, entendido como el grafo que es posible inferir de G usando R. No obstante, el costo que puede tener la extensiva definición de reglas y la insuficiente madurez en el soporte a reglas de inferencia en lenguajes de consulta y repositorios de tripletas, hacen recomendable la reutilización de vocabularios por sobre la creación de nuevos, aunque no podrá evitarse el compromiso con la naturalidad de los vocabularios frente al uso.

En la sección 7 se presentará una serie de vocabularios que conforman un núcleo común que se propone para su uso en la publicación de gran parte de la información pública en Chile.

6.2.2. Separar clases, predicados e instancias

Cómo modelar sin utilizar elementos que introducen complejidad en la computabilidad.

El problema descrito ocurre por ejemplo en elementos que son agrupados utilizando múltiples clasificaciones. Por ejemplo, los proyectos evaluados para impacto ambiental poseen múltiples clasificaciones ortogonales: territorio, industria y tipo de proyecto. Existe la tentación de, por ejemplo, definir la clase "Regiones" dentro de la cual agregar las clases "Región de Tarapacá", "Región de Antofagasta", "Región de Atacama", etc. Con este modelo un proyecto p que se encuentre en una región r dentro del cojunto C de todas las regiones implicará las relaciones:

$$p \in r \in R$$

El problema de esto es que la clase r cumple simultáneamente los roles de clase (por incluir a p) e instancia (por pertenecer a R). En la especificación de RDF Schema se indica que el predicado `rdf:type`, usado para expresar la relación $x \in A$ implica que $A \in$ `rdfs:Class` [132].

La especificación de OWL recomienda explícitamente que las clases, las propiedades y las instancias sean conjuntos disjuntos [133]. El problema de entregar mayor expresividad permitiendo que las clases puedan ser tratadas como instancias es que se salta del lenguaje OWL DL a OWL Full, lo que implica un compromiso con la decibilidad [134].

Una solución a este problema particular de agregar clasificaciones a un objeto se desarrolla en la sección 6.2.3, donde se propone usar SKOS como una alternativa para definir clasificaciones y taxonomías.

6.2.3. Usar SKOS para clasificar elementos

> Cómo indicar que un elemento es de un determinado tipo y a la vez poder utilizar ese tipo como una instancia de otra clasificación. Dicho de otro modo, cómo codificar una jerarquización en tres niveles o más.

En la sección 6.2.2 se describe el problema de clasificar un elemento usando múltiples clasificaciones. Allí se indica que no resulta adecuado utilizar la relación $\in$, expresada como el predicado rdf:type, pues implica utilizar la expresividad de OWL Full. Para resolver este problema, en esta sección se entregará una solución alternativa, que hace uso del vocabulario *Simple Knowledge Organization System* (SKOS) [135]. Por ejemplo, para clasificar los proyectos de evaluación de impacto ambiental se podría crear un concepto "Regiones de Chile", más amplio que los conceptos asociados con cada región y luego indicar que el proyecto en particular es también un concepto particular de la región, tal como se presenta en la figura 6.2.

Figura 6.2: Clasificación de proyectos por regiones. Un proyecto es representado como un concepto más refinado que el de la "Región de Tarapacá" que, a su vez, es una especialización del concepto "Regiones de Chile".

Es necesario recalcar que el predicado skos:broader no es una relación transitiva, por lo que

en el ejemplo presentado en la Figura 6.2 es correcto en el sentido que los proyectos no son una especialización del concepto "Regiones de Chile". SKOS provee otros predicados para definir jerarquías transitivas de elementos: skos:broaderTransitive y skos:narrowerTransitive.

El modelamiento que se ofrece en la figura 6.2, pese a ser correcto, no aparece en los usos observados de SKOS, en cambio, se suele extender la clase de skos:Concept a una clase especializada, como org:Role, o agrupando los términos en una instancia de skos:Collection. Ambos ejemplos son presentados en la figura 6.3.

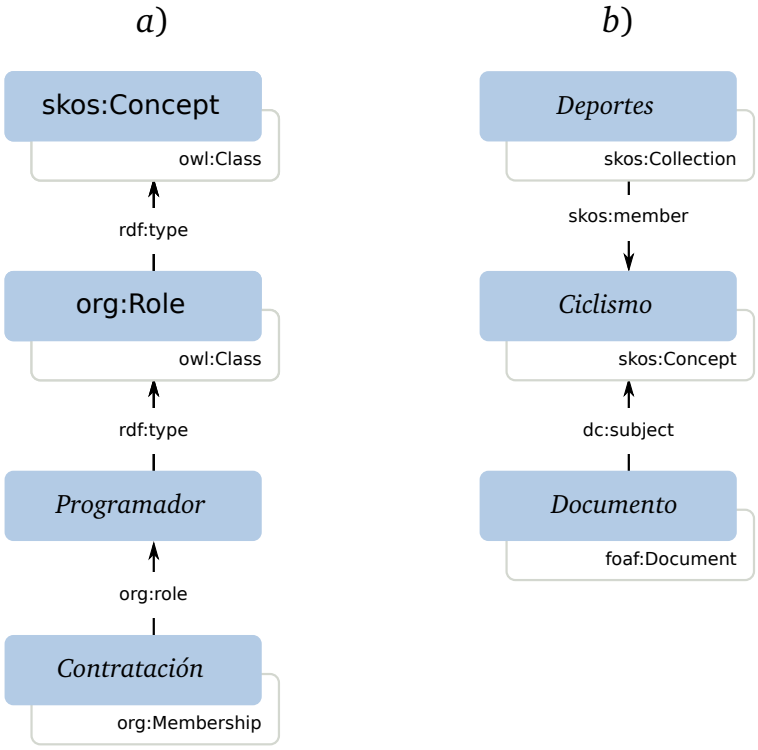


Figura 6.3: Esquemas de uso de SKOS. En el ejemplo *a*) el elemento que agrupa los conceptos es la creación de una clase particular org:Role, en cambio en *b*) tal clase es reemplazada por una colección. Ambos usos tienen en particular la creación de los predicados org:role y dc:subject para relacionar los recursos con los conceptos, es decir, prefieren no entregar a los recursos la cualidad de conceptos.

6.2.4. Usar CamelCase

Cómo facilitar la identificación entre predicados y clases del vocabulario y a la vez uniformar las nomenclaturas en la creación de vocabularios.

Varios vocabularios han optado por usar la notación CamelCase, entre ellos FOAF. Por lo tanto, para ser consistentes con las convenciones que se han impuesto, se recomienda usar UpperCamelCase para definir nombres de clases y lowerCamelCase para definir nombres de predicados.

Esta notación también puede ser útil para diferenciar clases de predicados que poseen el mismo nombre, tal como en la ontología para organizaciones propuesta por Dave Reynolds [136], donde se usa CamelCase para diferenciar el predicado `organization` de la clase `Organization`.

El uso de esta notación es además consistente con la notación empleada actualmente en la definición de esquemas XML desarrollados por organismos de gobierno, que fue establecida en la Guía de Desarrollo y Uso de Esquemas de Gobierno [137].

6.2.5. Evitar la pérdida de información

Cómo entregar información cuando los vocabularios no alcanzan para expresar los datos que poseemos.

Puede que un vocabulario popular no sea capaz de expresar con exactitud los datos que poseemos, llevándonos a perder información. Un ejemplo común de esto es lo que pasa con los nombres de las personas y para lo cual se presenta una solución en la sección 7.3. Las personas pueden ser identificadas, dentro del vocabulario FOAF, por los predicados `foaf:name`, `foaf:givenName` y `foaf:familyName`. Al usar sólo el predicado `foaf:name` se pierde la distinción entre las palabras que forman al nombre de pila y las palabras que forman los apellidos. Esta separación puede recuperarse si se usan los predicados `foaf:givenName` y `foaf:familyName`. Siguiendo el mismo ejemplo, en Chile es común diferenciar entre el apellido paterno y el materno, por lo que vocabularios como los propuestos por la Biblioteca del Congreso Nacional [138] y por la Poderopedia [139] definen extensiones al término `foaf:familyName`.

6.2.6. Usar términos integradores

Cómo definir un conjunto de términos que sean usados en todos los esquemas de modo que las aplicaciones puedan asumir que tales términos aparecerán independientemente del contexto.

En el ejemplo anterior se recomendó usar predicados más específicos (ver sección 6.2.5) con el fin de evitar la pérdida de la información. Sin embargo, ello puede significar dejar de lado términos de vocabularios populares. No resulta sencillo definir un punto de equilibrio para este compromiso entre la precisión y la integrabilidad, pues ello depende de los lenguajes que usemos para consultar los datos y, en especial, de su capacidad para hacer inferencias sobre las jerarquías de clases y predicados.

Una solución alternativa a este problema es incorporar tanto términos usando los vocabularios especializados como términos utilizando aquellos más generales. Sin embargo, en este caso la integrabilidad presenta un compromiso con la redundancia en nuestros datos.

Algunos ejemplos de términos integradores son aquellos que se definen en los vocabulario RDFS y SKOS, tales como rdfs:label y skos:prefLabel, que pueden ser de gran utilidad para programas visualizadores de los grafos RDF.

6.2.7. Componer o heredar

Cómo decidir cuándo componer y en qué caso heredar.

La discusión entre componer o heredar ha sido común en el modelamiento orientado a objetos [140]. Los que apoyan la composición buscan tener objetos pequeños y con semántica precisa, mientras, quienes prefieren heredar logran menos clases y con jerarquías más planas.

En el caso del modelamiento en RDF la discusión aparece en casos como el siguiente: En el vocabulario para la transparencia de la Biblioteca del Congreso Nacional (BCN) [141] la clase bcnt:PositionPeriod extiende a la clase `time:Interval`[2] para agregar en ella propiedades de la contratación como la región, la categoría, el segmento, un indicador sobre si tiene salario y relaciones con instancias de las clases bcnt:Position y bcnt:Functionary. Este tipo de

[2] En la versión 1.1 del vocabulario bcnt se indica que extiende a la clase time:Instant pero ello es un error que debiera ser corregido en la siguiente revisión del vocabulario.

extensión, en la que un objeto extiende a otro que conceptualmente no es un subtipo, no es recomendable. Daniel Halbert y Patric O'Brien detallan en [140] el por qué esta práctica, que llaman *subtiping by variance*, no es recomendable.

Un diseño alternativo habría sido agregarle al intervalo un predicado desde la clase bcnt:PositionPeriod hacia la clase time:Interval, como se hace en el caso de la ontología ORG, que relaciona org:Membership con time:Interval a través del predicado org:memberDuring, o como también lo hace el vocabulario OPMV, para relacionar instancias de opmv:Process con time:TemporalEntity a través del predicado opmv:wasPerformedAt. Los ejemplos de los vocabularios OPMV y ORG siguen un patrón que se observa en varios vocabularios populares y que es recomendable seguir. Una discusión más detallada sobre este caso se hará en la sección 7.5.

En resumen, se recomienda extender las clases sólo cuando la clase heredada corresponde conceptualmente a un subtipo de la clase original. Cuando esto no ocurra, lo más recomendado es utilizar la composición.

Casos comunes

1. Para indicar el tiempo en eventos lo más conveniente es establecer una relación de composición entre el evento y el tiempo.

2. Para indicar una relación entre un lugar y sus coordenadas, lo más conveniente es usar una relación de composición.

6.2.8. Generalizar el vocabulario y precisar los datos

Cómo lidiar con modelos con demasiadas restricciones y que podrían cambiar en el tiempo.

Quienes definen un vocabulario RDF pueden caer en la tentación de que éste se ajuste demasiado al uso que van a darle. Sin embargo, ésto podría tener consecuencias negativas, en términos de que el vocabulario pueda extenderse o que pueda usarse en otros contextos. Por ello, se recomienda definir vocabularios que sean flexibles en cuanto a las posibilidades de codificación de datos, de modo que las relaciones entre las instancias se presentan más a través del uso que en restricciones impuestas por el esquema.

Un caso que ejemplifica este problema es el del vocabulario de la BCN para la transparencia [141], que se discute con mayor detalle en la sección 7.5. En él, las posiciones dentro del organigrama se relacionan con la clase `bcnt:SubOrganization`, que es una subclase (indirectamente) de la clase `org:Organization`. El restringir el recorrido de este predicado, tiene como consecuencia que éste no pueda ser usado correctamente en organizaciones que no se definan como instancias de esta clase. Por lo tanto, se dificulta el uso del vocabulario desarrollado fuera de la BCN.

La pregunta que surge inmediatamente es cuál fue el motivo para definir la clase bcnt:SubOrganization y sus dos derivadas bcnt:Direction, bcnt:Department, bcnt:SubHead y bcnt:Area siendo que la relación de anidación organizacional ya se encontraba en el vocabulario ORG, que sirve de base para el vocabulario de transparencia de la BCN. Una justificación posible pareciera ser el deseo de permitir definir predicados tales como bcnt:hasArea que relaciona a departamentos con áreas o modela restricciones tales como que sólo los departamentos tienen una subjefatura y que ésta es una unidad organizacional intermedia entre los departamentos y las áreas. Modelar todas estas particularidades en la estructura organizacional de la BCN hace que el vocabulario no sea útil para otros usos fuera de la BCN.

La solución a este problema de sobre especificación es usar el vocabulario más general posible. En la sección 7 veremos que el mismo vocabulario ORG es suficiente para definir la estructura organizacional. Acompañado de SKOS se puede categorizar los tipos de unidades organizacionales dentro de la BCN. De esta forma, no se requiere agregar relaciones especiales entre dos tipos de unidades organizacionales, pues basta saber el tipo de unidad, expresado con SKOS, para interpretar el tipo de relación. El uso del vocabulario SKOS, nos permite evitar pérdida de información en cuanto a los tipos de organización.

Uno de los posibles motivos de la sobre especificación es traer las prácticas aprendidas en el modelamiento OOP y de bases de datos relacionales al modelamiento de vocabularios RDF. En el mundo de las bases de datos relacionales, quienes modelan están acostumbrados a ser estrictos en la definición del recorrido y dominio de las relaciones usando llaves foráneas. Eso define modelos rígidos para un uso particular y que se comportan eficientemente con la creación de índices apropiados. En cambio, en el mundo RDF, se debe buscar la generalidad del vocabulario, de ese modo la estructura de los datos estará expresada por ellos mismos y no por los esquemas que los contienen. En nuestro ejemplo de la estructura organizacional, más vale que la regla "las subjefaturas son unidades intermedias entre los departamentos y las áreas" quede expresada implícitamente a través de los casos, que mediante restricciones explícitas en el modelamiento, pues de ese modo podremos usar el modelo en casos donde tal

regla no sea válida.

Por último, no todas las reglas que se satisfacen en un modelo necesitan ser expresadas explícitamente. Junto con el esfuerzo que ello significa, puede exigir una expresividad innecesaria al lenguaje de modelamiento. Por ejemplo, en el modelo relacional nos es imposible definir que una relación de recurrencia definida mediante una llave foránea a una misma tabla tenga un largo máximo determinado o que no presente ciclos. Además, no hay que olvidar que la expresividad tiene un compromiso con la computabilidad. En el caso de ser necesario verificar que cierta regla sobre los datos se cumpla, puede resultar más práctico hacerlo a nivel de aplicación, validando que se satisfaga un determinado patrón mediante un consulta.

6.2.9. No crear modelos efímeros

Cómo escribir modelos que sean estables en el tiempo.

Un corolario del problema planteado en la práctica de la sección anterior (6.2.8) es que sobre especificar un modelo puede obligarnos a cambiarlo con frecuencia. La inestabilidad de los modelos tiene un costo para quienes desarrollan aplicaciones, pues deberán reimplementar las operaciones que hacen sobre ellos. Para evitar esto, es recomendable que los modelos sean generales y que, tal como se recomendó en la práctica anterior, que la particularidades sean definidas en lo posible a través de los datos.

6.2.10. Establecer equivalencias débiles

Cómo establecer relaciones de equivalencia sin introducir inferencias erróneas.

Uno de los problemas centrales de la integración de datos es cómo identificar que dos objetos descritos en dos fuentes de datos distintos corresponden a un mismo objeto del mundo real. Este problema es conocido como Identity Resolution (ER). En general este problema es descrito como la definición de algoritmos que nos permiten identificar un recurso r_1 proveniente de una fuente con un recurso r_2 proveniente de otra, para generar un nuevo recurso r_3 que integre los atributos de ambos. Además del proceso de encontrar elementos similares y realizar la operación de mezcla, el problema de ER suele incluir un trabajo con la confianza

que se tiene sobre los datos originales (¿cuán ciertos son los atributos de r_1 y r_2?) y con la confianza sobre la operación de mezcla [142].

En el mundo de los datos enlazados el manejo de los niveles de confianza sobre los datos y sobre las operaciones que nos permiten generar nuevos datos a partir de datos previos es rara vez abordado. En cambio, las recomendaciones se centran en cómo actuar cuando la certeza de que se trata de los mismos objetos es absoluta. L. Doods y I. Davis, proponen el uso de predicados tales como owl:sameAs, skos:exactMatch para indicar la equivalencia entre dos recursos [122]. No obstante, esto suele ser una fuente de problemas. Por ejemplo, en la DBPedia aparece una equivalencia entre las siguientes URIs usando el predicado owl:sameAs:

1. http://www4.wiwiss.fu-berlin.de/flickrwrappr/photos/Santiago_de_Chile
2. http://dbpedia.org/resource/Santiago_de_Chile

El problema entre ligar estas URIs es que la primera responde a una petición, por lo tanto representa a un documento (en este caso una lista de fotos), mientras, la segunda responde con una redirección hacia el documento que describe el recurso, por lo cual se trata de un recurso ideal. Es decir, tal como se discutió en la sección 6.1.1, se están mezclando identificadores de documentos con identificadores de objetos ideales.

Pero el problema que implica el uso owl:sameAs va más allá. La primera pregunta que hay que responder es si tiene sentido decir que dos objetos que son descritos en datasets distintos, producidos por personas que poseen distintas maneras de comprender el mundo, corresponden a un mismo objeto. A diferencia de los modelos habituales donde se plantea el problema de ER, en el mundo de los datos enlazados el universo de los modelos está abierto no sólo a la publicación de datos de manera distribuida sino también en los modelos que describen sus objetos. Supongamos por ejemplo, que en un dataset se indica que Alicia nació en Paillaco, en la Región de los Lagos. Esa región más tarde fue dividida y Paillaco queda actualmente en la Región de los Ríos. La pregunta es si debemos afirmar si la Región de los Lagos orginal corresponde o no a la misma Región de los Lagos luego de su cambio. Si suponemos que son las mismas podríamos llegar a errores lógicos, como suponer que aquellos hechos que estaban circunscritos a la región lo siguen estando. En cambio, suponer que son distintas regiones implica tener que usar distintos identificadores y, si queremos indicar que poseen algo en común, debemos señalarlo con alguna relación.

Si usamos la relación owl:sameAs para relacionar dos identificadores x e y estaremos asumiendo que los recursos son idénticos, es decir, que todas nuestras afirmaciones hechas sobre

x serán también válidas sobre y. Este hecho queda expresado en la ley de los indicernibles (también conocida como la Ley de Leibnitz):

$$\forall x \forall y \, (x = y \rightarrow \exists P(P(x) \leftrightarrow P(y)))$$

De este modo, introducir esta relación en el caso de las regiones nos lleva a posibles errores lógicos, pues, dado que no son entidades realmente equivalentes, ciertas afirmaciones hechas sobre la antigua Región de los Lagos podrían no ser válidas sobre la nueva.

En [143] se discuten los malos usos del predicado owl:sameAs, se propone *Similarity Ontology* (SO) para describir distintos grados de similaridad y se explora el uso práctico de esta ontología. La figura 6.4 muestra las relaciones de herencia (relación rdfs:subPropertyOf) entre los distintos predicados de la ontología propuesta.

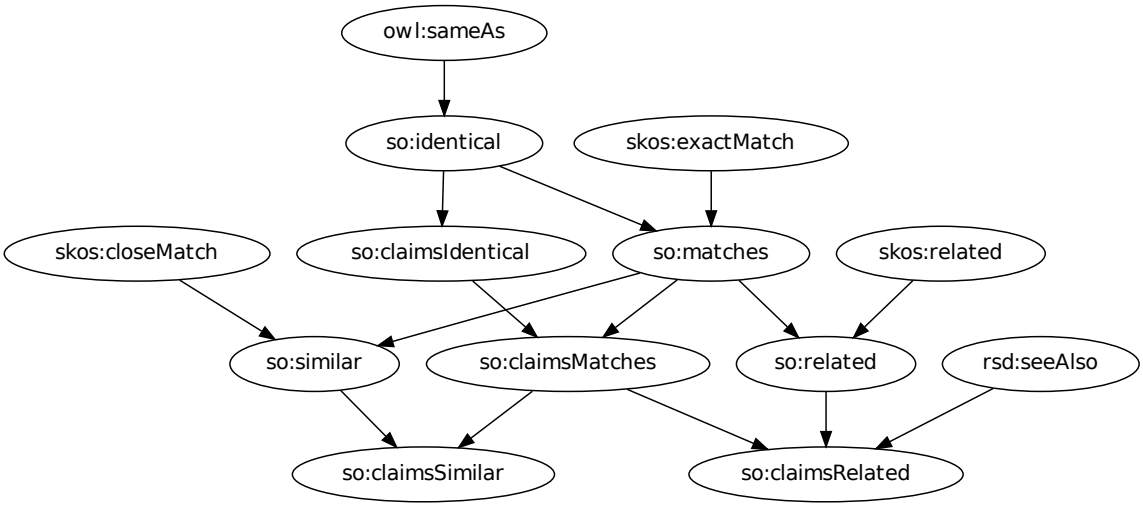


Figura 6.4: Relaciones de herencia entre los diferentes predicados de *Similarity Ontology* (SO).

Gran parte de los problemas de los datos enlazados tienen que ver con el contexto en el cual las afirmaciones fueron realizadas. Por ejemplo, los predicados foaf:age y foaf:homepage están íntimamente vinculados con el contexto y son afirmaciones que están sujetas a cambios en su validez en el tiempo. De igual modo, un estudio sobre delincuencia depende del significado de los delitos dentro del país, por lo que comparar tales estudios entre países donde las definiciones difieren, es una tarea que hay que realizar con cuidado. Lo mismo ocurre si se desean comparar estudios basados en encuestas cuyas preguntas o metodologías no son las mismas. Este contexto puede identificarse en RDF mediante el grafo que contiene el dataset,

razón por la cual en [143] se discuten trabajos futuros que relacionan la identificación grafos con las relaciones de equivalencia.

Si bien la ontología SO provee de predicados que pueden servir para establecer relaciones de equivalencia[3] entre recursos, no siempre este tipo de relaciones son lo que se necesita para indicar que dos URIs poseen cierta similitud. En el caso de la Región de los lagos, por ejemplo, se puede utilizar una relación de modificación provista por el vocabulario Open Provenance Model (OPMV) [105], tal como opmv:wasDerivedFrom o una instancia de opmv:Process para establecer la relación de derivación.

6.2.11. Afirmar hechos independientes del tiempo

Cómo hacer afirmaciones cuya validez depende del tiempo.

En el lenguaje común solemos hacer afirmaciones tales como Miguel tiene 15 años, Alicia estudia en la Universidad o Francisco será el próximo director de la organización. Estas afirmaciones dependen del contexto en el que se hacen y particularmente del tiempo. A diferencia del lenguaje natural, en RDF las tripletas no dependen del orden en las que se ponen ni del contexto del grafo en el que se proponen. Es esperable que en RDF, dados dos grafos G_1 y G_2 puedan tomarse dos subgrafos de ellos $G'_1 \subset G_1$ y $G'_2 \subset G_2$ y que el grafo $G'_1 \cup G'_2$ no introduzca hechos inconsistentes con la realidad descrita[4].

Las afirmaciones cuya validez depende del tiempo no pueden ser formuladas directamente mediante un sólo triple, sino que deben ser pensadas como relaciones N-arias. Por ejemplo, "Alicia estudió en la Universidad entre 2005 y 2012". En tales relaciones, el predicado debe transformarse en una instancia que pueda ser caracterizada con sus propios atributos [144].

A pesar de que el que las tripletas sean independientemente válidas parece un requerimiento básico, varios de los vocabularios más usados incluyen tales predicados. Ejemplos de ello son los predicados foaf:age y org:headOf. Por esta razón es recomendable no usar tales vocabularios y en cambio usar términos que sean independientes del tiempo como bio:birth y :headOf[5].

[3] Recordar que una relación R es de equivalencia cuando es refleja: $\forall x(xRx)$, simétrica: $\forall x \forall y(xRy \rightarrow yRx)$ y transitiva: $\forall x \forall y \forall z(xRy \wedge yRz \rightarrow xRz)$.

[4] Cuando se usan nodos blancos tal unión suele ser en realidad una mezcla en la que reescriben los símbolos de los nodos blancos.

[5] El vocabulario :headOf se propone en esta tesis en la sección 7.4.

Capítulo 7

Vocabulario propuesto

Este capítulo presenta cómo aplicar vocabularios RDF a la publicación de datos para los casos de estudio presentados en el capítulo 4. En particular, los ámbitos tomados como casos de estudio son la publicación de datos de transparencia y las biografías parlamentarias. La elección de estos casos de estudio se justifica en que la información de transparencia y las biografías parlamentarias forman un núcleo que toca los temas políticamente sensibles en la publicación de datos y que se entrelaza con otros conjuntos de datos que se refieren a personas y organizaciones, como por ejemplo los datos de la Poderopedia.

7.1. Vocabularios usados

A lo largo de este capítulo usaremos y discutiremos repetidamente sobre vocabularios existentes, que serán referidos a través de los prefijos que presentaremos a continuación. En particular, el prefijo vacío ":" será utilizado para los términos que propondremos para ser agregados y que se documentan en mayor detalle en la sección A.

xsd: http://www.w3.org/2001/XMLSchema#

Es el espacio de nombre para los tipos definidos en *XML Schema* para ser valores de literales en XML.

rdf: http://www.w3.org/1999/02/22-rdf-syntax-ns#

Vocabulario básico que describe aspectos básicos de RDF.

rdfs: http://www.w3.org/2000/01/rdf-schema#

RDF Schema, un lenguaje para definir esquemas en RDF. La documentación se encuentra en http://www.w3.org/TR/rdf-schema.

owl: http://www.w3.org/2002/07/owl#

Web Ontology Language pemite describir ontologías para RDF. La documentación se encuentra en http://www.w3.org/2004/OWL/.

skos: http://www.w3.org/2004/02/skos/core#

Simple Knowledge Organization System es una ontología para describir esquemas conceptuales. Su documentación se encuentra en http://www.w3.org/TR/skos-reference/.

dbpo: http://dbpedia.org/ontology/

Prefijo para los términos del vocabulario de la DBpedia.

dbpr: http://dbpedia.org/resource/

Prefijo para los recursos de la DBpedia.

dc: http://purl.org/dc/terms/

Los *DCMI Metadata Terms* constituyen un vocabulario que es ampliamente usado para describir recursos en bibliotecas. La documentación de ellos está en http://dublincore.org/documents/2012/06/14/dcmi-terms/.

foaf: http://xmlns.com/foaf/0.1/

Friend of a Friend es un vocabulario para describir y entrelazar personas y organizaciones. Su documentación se encuentra en http://xmlns.com/foaf/spec/.

bio: http://purl.org/vocab/bio/0.1/

Un vocabulario basado en foaf para información biográfica. Su documentación se encuentra en http://vocab.org/bio/0.1/.html.

gr: http://purl.org/goodrelations/v1#

Vocabulario para describir relaciones comerciales. Documentado en http://www.heppnetz.de/ontologies/goodrelations/v1.html.

lode: http://linkedevents.org/ontology/

Un vocabulario minimal para describir eventos. Su documentación se encuentra en http://linkedevents.org/ontology/.

rel: http://purl.org/vocab/relationship/

Un vocabulario para codificar relaciones entre personas. Su documentación se encuentra en http://vocab.org/relationship/.html.

org: http://www.w3.org/ns/org#

Vocabulario para organizaciones desarrollado para la descripción de organizaciones en el contexto de la publicación de datos en UK. La documentación puede encontrarse en http://www.epimorphics.com/public/vocabulary/org.html.

opmv: http://purl.org/net/opmv/ns#

Vocabulario creado en base al *Open Provenance Model*. Su documentación puede encontrarse en http://purl.org/net/opmv/ns.

bcnt: http://datos.bcn.cl/ontologies/bcn-transparency#

Vocabulario de la BCN para publicar los datos de transparencia, versión 1.1. Su documentación está en http://datos.bcn.cl/ontologies/bcn-transparency/doc/.

bcnb: http://datos.bcn.cl/ontologies/bcn-biographies#

Vocabulario de la BCN para publicar los datos de la Historia Parlamentaria. Su documentación puede encontrarse en http://datos.bcn.cl/ontologies/bcn-biographies/doc/.

pod: http://poderopedia.com/vocab/

Vocabulario para la Poderopedia. Sus versiones pueden revisarse en https://github.com/poderopedia/PoderVocabulary.

7.2. Notación

Varios de los ejemplos de este capítulo serán descritos con ayuda de grafos. Para ellos se introducirá una notación que simplificará la identificación de las instancias y sus clases. La figura 7.1 presenta las notaciones usadas.

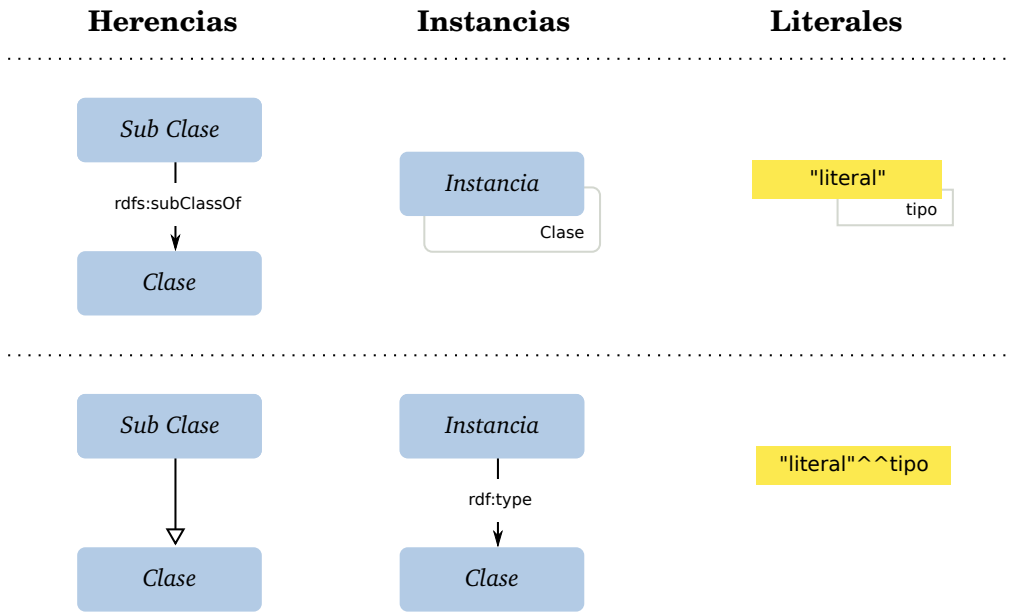


Figura 7.1: Notación de las figuras. Las cursivas en instancias indican que se trata de símbolos usados sólo para los ejemplos.

7.3. Personas naturales

Algunos vocabularios para describir personas son foaf, rel, bio y pod. El primero de ellos es la base de la mayoría de los vocabularios que se refieren a las personas. El vocabulario bio es una extensión de foaf, pero a diferencia de él, que está orientado a publicar datos vivos, es decir, información actual, bio está orientado a publicar información contextualizada en el

tiempo. Esta diferencia se hace notoria al contrastar los predicados foaf:age y bio:birth. El primero, indica la edad actual de una persona, lo que para ser interpretado debe tener en cuenta el contexto en el cual se presenta la información. En cambio, el segundo indica la fecha de nacimiento, de la cual siempre resulta posible deducir la edad, sin requerir conocer el contexto.

Nombres de las personas. En general, para nombrar recursos existen los predicados rdfs:label y foaf:name que corresponden a nombres de los dominios son rdfs:Resource y owl:Thing, respectivamente. Predicados más específicos son foaf:givenName y foaf:familyName para nombres y apellidos respectivamente. Sin embargo, en Chile es común que a las personas se las identifique por sus nombres y sus apellidos, distinguiendo entre el apellido paterno y el materno. Esta distinción es un detalle cultural que no está presente en el vocabulario foaf, que busca ser lo más universal posible. En consecuencia, resulta práctico añadir predicados más específicos al vocabulario, tal como se muestra en la figura 7.2.

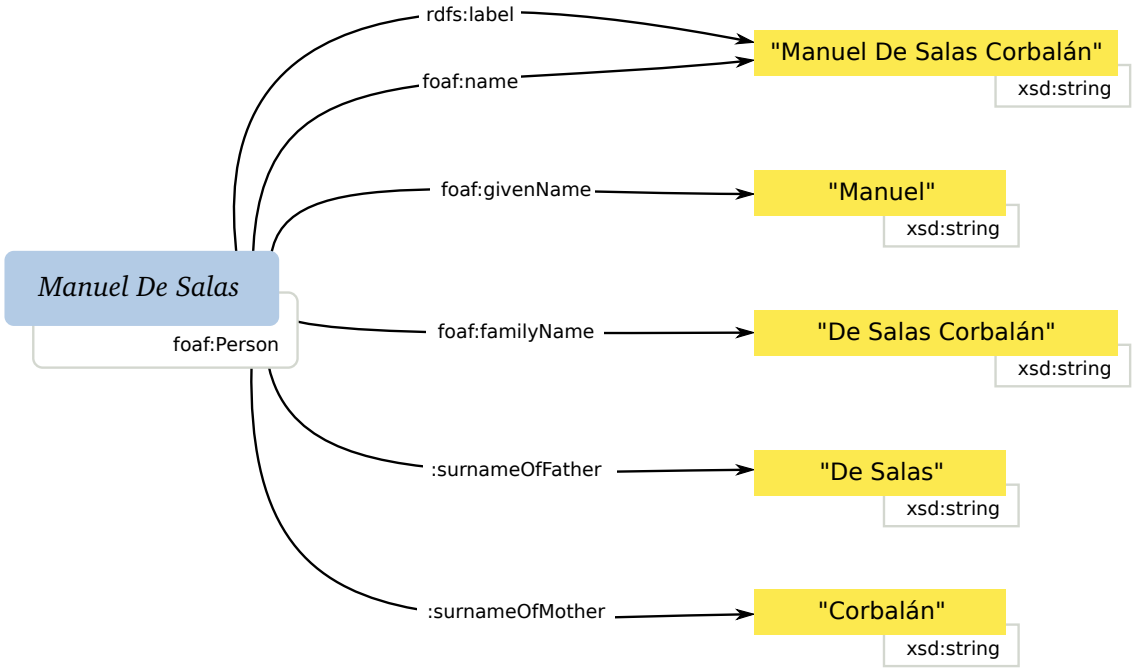


Figura 7.2: Ejemplo de uso del vocabulario propuesto para definir los nombres de las personas.

Relaciones entre personas. Las relaciones entre las personas pueden ser organizadas en dos grupos: el de aquellas cuya validez no depende del tiempo, como ser hijo o hermano, y las relaciones temporales como los matrimonios o ser vecinos. Esta diferencia hace que el primer tipo de relaciones pueda ser modelado mediante relaciones binarias, mientras que el

segundo, para incluir el contexto, debería ser modelado mediante la creación de instancias a las cuales anexar estos atributos, como lo hace por, ejemplo el vocabulario bio con la clase bio:Relationship o el vocabulario rel con la clase rel:Relationship. En general, en el modelamiento de redes sociales es común utilizar un grafo bipartito en el cual las relaciones también son nodos a los cuales se les pueden agregar atributos [145].

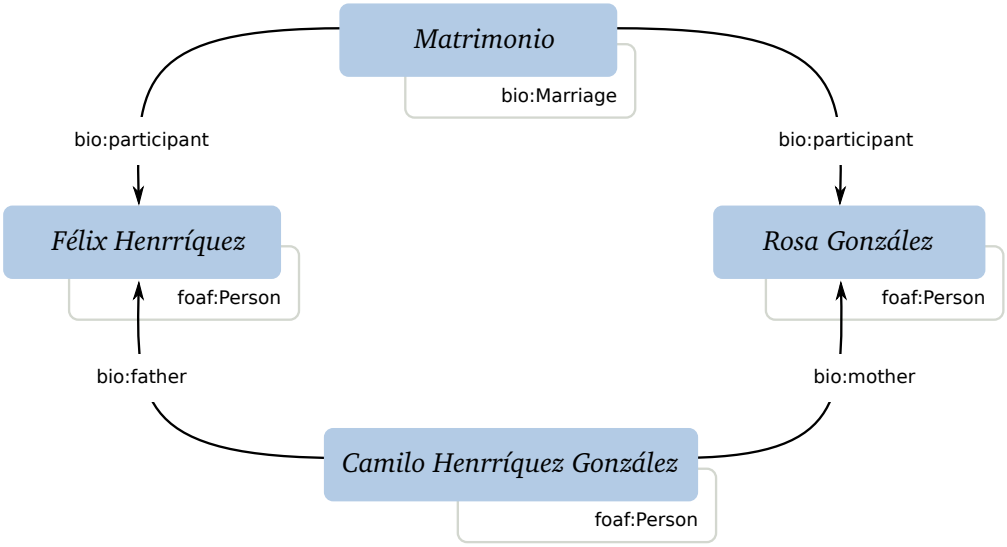


Figura 7.3: Relaciones entre personas. Se muestran dos tipos de relaciones entre Felix y Rosa. La relación es modelada por una instancia, pues podría tener otros atributos, en cambio, las relaciones de parentezco son modeladas sólo mediante arcos, pues no requieren de más atributos.

Eventos. Entre los vocabularios orientados a describir eventos de personas se encuentran bio y lode. En el caso de las personas, los eventos más relevantes a considerar son el nacimiento y la muerte. Ambos eventos pueden ser descritos mediante las instancias bio:Birth y bio:Deadth. Entre los atributos de estos eventos, la fecha (bio:date) y el lugar (bio:place) son suficientes. En particular el rango de los lugares está abierto y en el ejemplo de la figura 7.4 se usan lugares definidos dentro de la DBpedia.

7.4. Personas jurídicas

En Chile se distinguen dos tipos de personalidades legales, la persona natural, que tiene una semántica similar a la clase foaf:Person y la persona jurídica que corresponde a entidades abstractas tales como organizaciones o empresas, que han sido registradas como tales me-

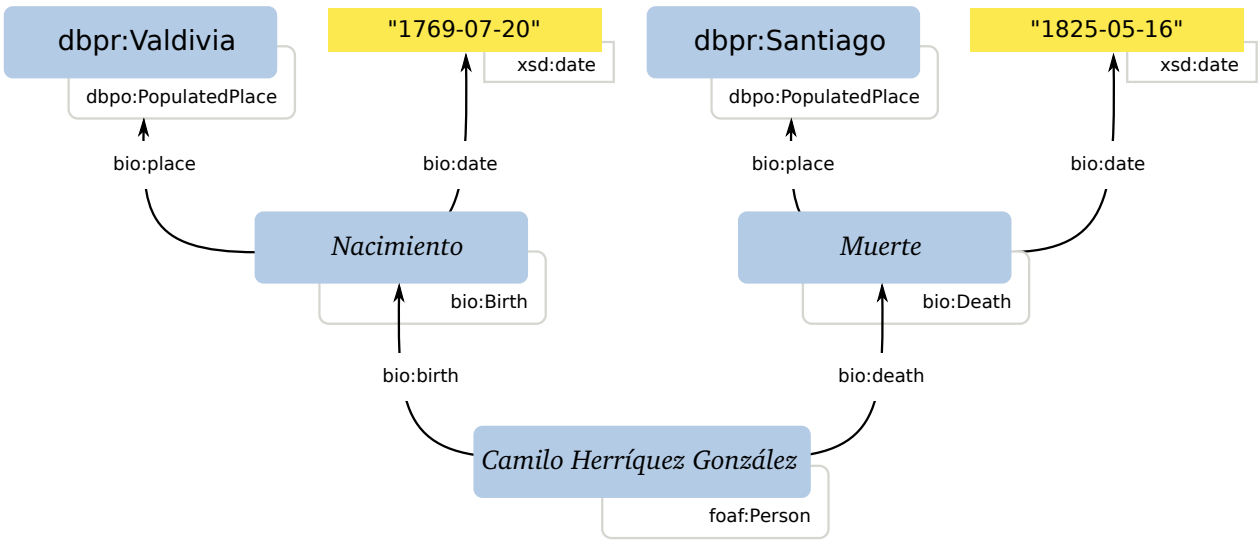


Figura 7.4: Nacimiento y muerte de una persona.

diante escrituras públicas, registros de comercio y en el Servicio de Impuestos Internos. Estas entidades poseen características, como el RUT, que las distinguen de las clases foaf:Agent, foaf:Person y foaf:Organization. Estas cararacterísticas particulares podrían impulsar a que alguien agregue clases más específicas como *Persona Tributaria*, *Persona Natural* y *Persona Jurídica*, que se relacionarían con las anteriores de la manera que se grafica en la figura 7.5. No obstante, si seguimos la práctica de ser precisos en los datos y no en el vocabulario (ver sección 6.2.8) llegaremos a que tal construcción es innecesaria.

Es necesario recalcar que en RDF no existen los problemas que la múltiple herencia genera en varios lenguajes de programación [146], por lo que la construcción anterior no sería incorrecta. No obstante, a pesar de su correctitud, implica costos al agregar más términos al vocabulario, obligar a formular consultas más complejas o requerir reglas de inferencia extras para la integración.

Identificación y nombres. El RUT es el elemento que nos permite identificar unívocamente a todas las personas legales (naturales o jurídicas). En cambio, las personas jurídicas tienen al menos dos nombres que pueden ser usados en contextos diferentes: la razón social y el nombre de fantasía. Sin embargo, dado que en los casos observados en la transparencia activa lo que habitualmente se usa es la razón social, resulta poco relevante hacer tal distinción entre los dos nombres. En resumen, para el Rut necesitamos crear el predicado :rut y para el nombre nos basta con reusar el predicado foaf:name.

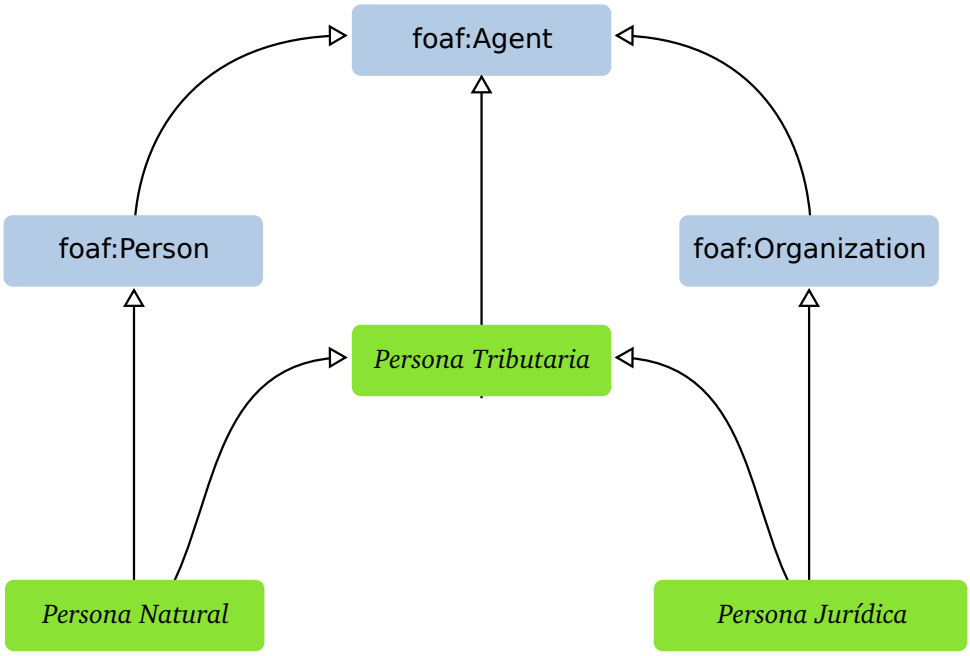


Figura 7.5: Ejemplo de jerarquía para entidades tributarias relacionada con el vocabulario foaf, que no sería necesaria pues se puede codificar la misma información siendo precisos en los datos en vez de en el esquema (ver sección 6.2.8).

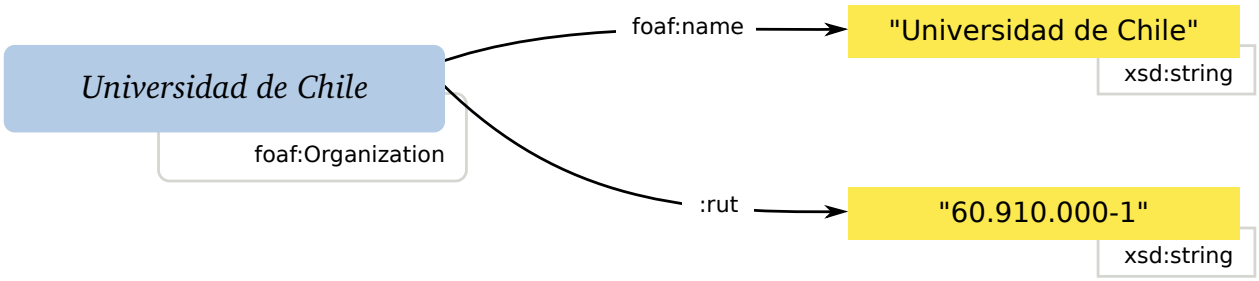


Figura 7.6: Nombre e identificador de personas jurídicas.

Estructuras orgánicas Una de las obligaciones de la transparencia activa es la publicación de las estructuras orgánicas de las instituciones públicas. Sin embargo, no existe un estándar sobre las relaciones que pueden existir entre las unidades organizacionales. Por ejemplo, en la estructura orgánica que se publica en el portal de la Biblioteca del Congreso Nacional aparecen: ser unidad dentro de otra unidad (los departamentos tienen áreas), dirigir (el consejo dirige a la biblioteca) y proveer (el área de sistemas y servicios de información en red provee sus servicios a todas las demás unidades). La relación más común de las estructuras orgánicas es la de composición, cuya modelación en la ontología org resulta suficiente para las necesidades de los casos de estudio. Incluso la Biblioteca del Congreso, con la diversidad

de relaciones, no ha omitido aquellas características que no son modeladas por el vocabulario org, proponiendo su propia ontología (bcnt), que se basa en org, pero tal como se verá a continuación, no agrega expresividad suficiente como para justificar un nuevo vocabulario y, además, va contra el principio de generalidad expuesto en la sección 6.2.8. La única debilidad visualizada en el vocabulario org es la falta de términos para definir los roles dentro de las organizaciones. En particular no resulta posible indicar que un rol se define dentro de una unidad organizacional o que puede tener la responsabilidad de dirigirla, funcionalidades que son entregadas por los términos :headOf y :belongsTo que se proponen en la figura 7.7. En cambio, el vocabulario org posee el predicado org:headOf que le entrega a instancias de foaf:Agent el rol de dirigir una organización. Esto introduce un predicado cuyo valor será acotado en el tiempo, problema que se discutió en la sección 6.2.11.

7.5. Cargos

Bajo el concepto de transparencia activa [147], los organismos deben describir a su personal. Tales contrataciones constituyen relaciones N-arias en las que se relacionan elementos como las personas contratadas, los cargos, las organizaciones y los tiempos en los que ocurren tales contrataciones. Tal relación requiere la creación de instancias a las que adjuntar los atributos de la relación N-aria. En las ontologías bcnt y bcnb, que se basan en la ontología org, se define la relación entre una persona, su cargo y la organización en la que lo desempeña. Sin embargo, tal como se presenta en la figura 7.8, la ontología bcnt agrega demasiados términos extra, ocho contra sólo dos del vocabulario que se propone en esta tesis.

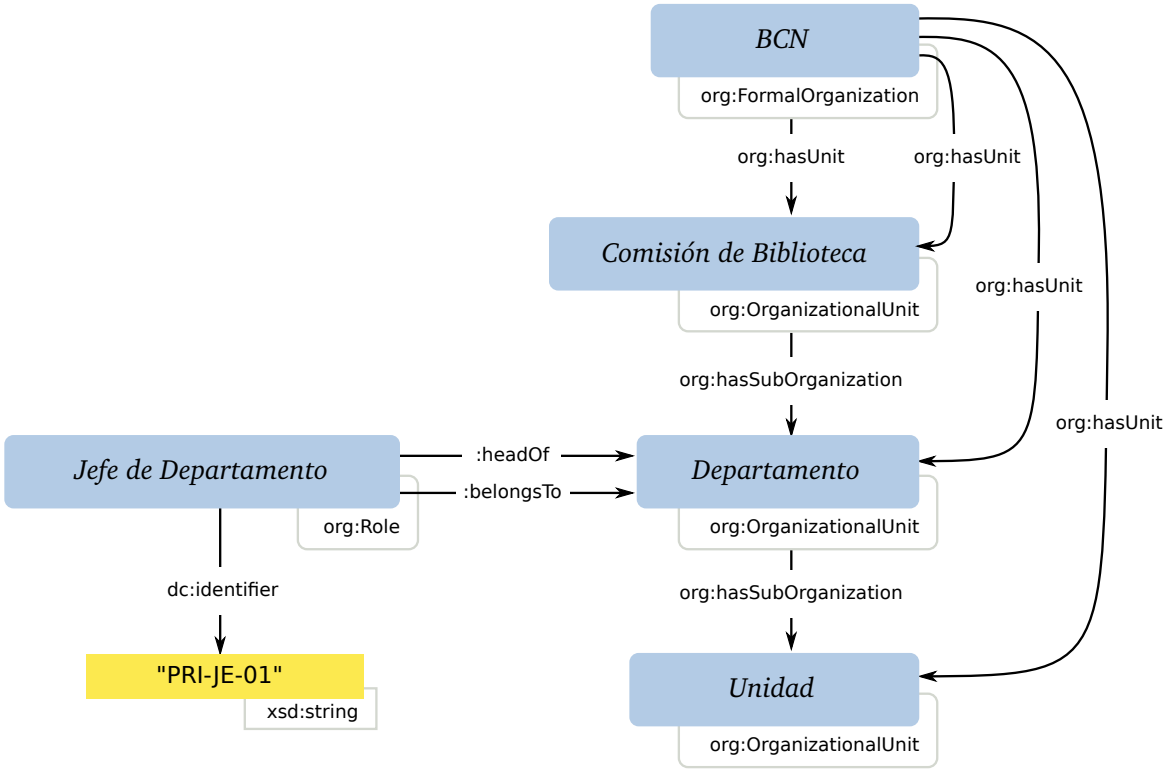


Figura 7.7: Fragmento de la estructura orgánica de la Biblioteca del Congreso Nacional. La Biblioteca tiene como unidad (org:hasUnit) a la Comisión de la Biblioteca, el Departamento de Producción de Recursos de Información y la Unidad de Arquitectura de Información. Esta relación org:hasUnit entre la Biblioteca y sus unidades es distinta de org:hasSubOrganization. Mientras la primera es usada para indicar las unidades de una organización formal, la segunda establece la jerarquía organizacional entre dichas unidades. El Departamento es dirigido por el rol de Jefe de Departamento. Para esta relación se ha introducido el predicado :headOf, pues el predicado org:headOf no posee el dominio adecuado, pues está asociado directamente con la persona que ejerce el cargo (ver discusión en esta misma sección). En cambio, esta propuesta independiza completamente la estructura orgánica de las contrataciones particulares que se establezcan. Además, se ha agregado el predicado :belongsTo para indicar la ubicación en la estructura orgánica donde se incertan los roles.

Las figuras 7.9, 7.10 y 7.11 muestran cómo es posible, de manera uniforme, usar la clase org:Membership para relacionar personas con sus cargos y acotar esta relación a los períodos de ejercicio. En particular, la figura muestra cómo es posible utilizar el vocabulario propuesto no sólo en el lugar de bcnt sino también en reemplazo del vocabulario bcnb.

El vocabulario propuesto no sólo es mejor que los de la Biblioteca por ser más general, sino también porque se observan problemas conceptuales en ellos, como por ejemplo, que la clase

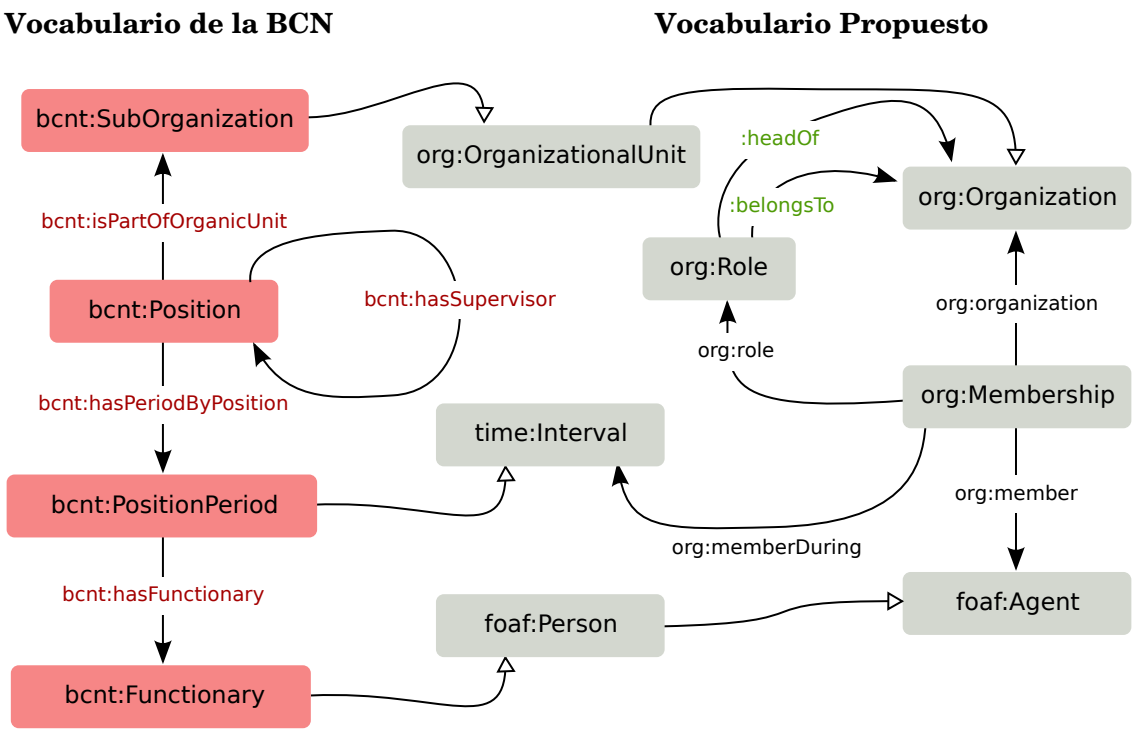


Figura 7.8: Comparación entre el vocabulario bcnt (en color rojo) y el propuesto en esta tesis (en color verde). Mientras el vocabulario propuesto reusa la clase org:Membership para agrupar las propiedades de las contrataciones, el vocabulario bcnt define la clase bcnt:PositionPeriod. Además, mientras la propuesta es reusar la clase org:Role, bcnt agrega la clase bcnt:Position. Las únicas dos funcionalidades visibles que el vocabulario bcnt agrega son la asociación de roles con la estructura orgánica (bcnt:isPartOfOrganicUnit) y la posibilidad de indicar la jerarquía entre los cargos (bcnt:hasSupervisor). Ambas funcionalidades son implementadas en esta propuesta a través de los predicados :belongsTo y :headOf, respectivamente. Este último no define la jerarquía como una relación directa entre los cargos, sino que lo hace definiendo supervisores de una organización en la que a su vez se definen otros cargos.

bcnt:Position extienda a time:Interval[1] con el fin de acotar la contratación a un período de tiempo. Este tipo de herencia es similar a extender una ventana desde un rectángulo, error de modelamiento que es discutido en la sección 6.2.7. El mismo problema de modelamiento se encuentra en el vocabulario bcnb con la clase bcnb:PositionPeriod y, además, se aprecian una falta de generalidad y redundancia al no contar con la misma clase en ambos vocabularios, como también ocurre con las clases bcnt:Position y bcnb:Cargo.

Los cargos también son un problema en ambas ontologías. En particular, la clase bcnb:PresidenteDelSenado hereda indirectamente de bcnb:Cargo, cuyas instancias se usan como atributo de las instancias de bcnb:Position. De este modo, la clase bcnb:PresidenteDelSenado tendrá una sóla

[1] En la versión 1.1 del vocabulario bcnt se indica que extiende a la clase time:Instant pero probablemente lo que se deseaba era extender de time:Interval.

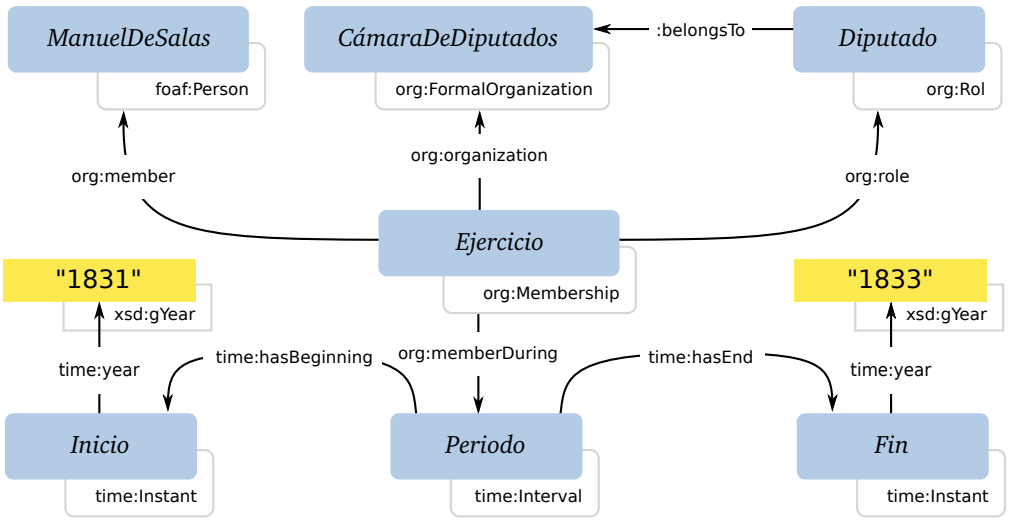


Figura 7.9: Ejemplo de uso de vocabulario propuesto para describir el ejercicio de *Manuel De Salas* en el cargo de *Diputado* entre los años 1831 y 1833.

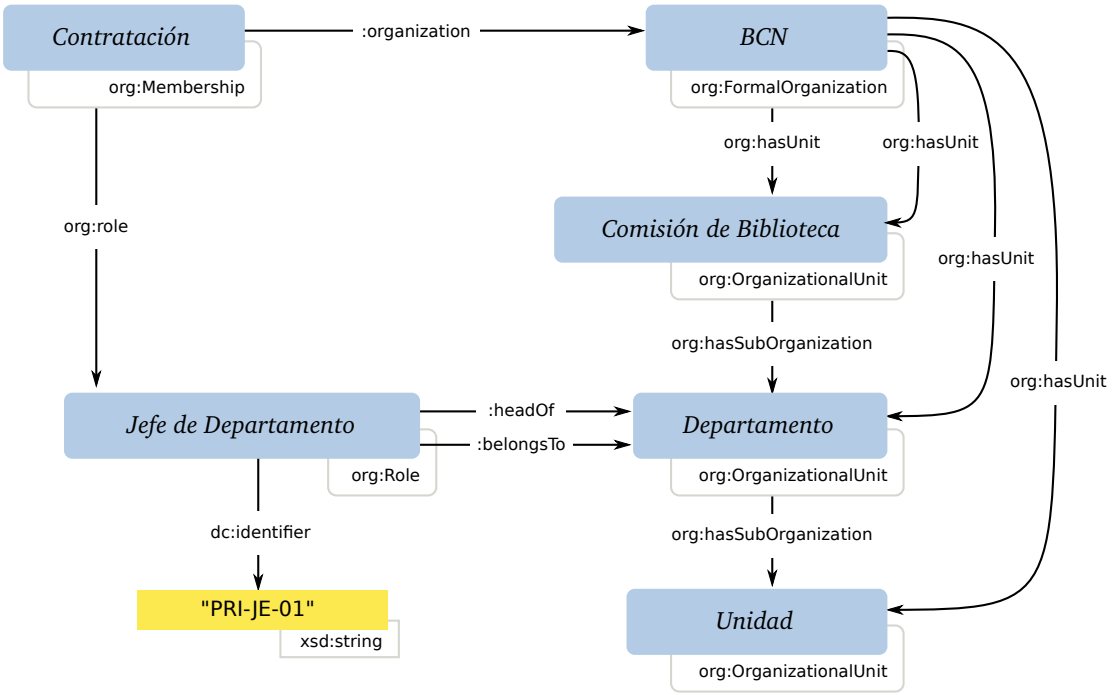


Figura 7.10: Inserción de las contrataciones en la estructura orgánica. En este ejemplo se basa en la figura 7.7.

instancia, pues es un único cargo. Detrás de la definición de tal singletón sin duda hay un problema de modelamentiento. A pesar de que todos los cargos también son instancias de skos:Concept, por compartir el atributo skos:prefLabel, no se usa jerarquía de los conceptos (con skos:broaderTransitive), sino que se estableció una jerarquía entre las clases (con

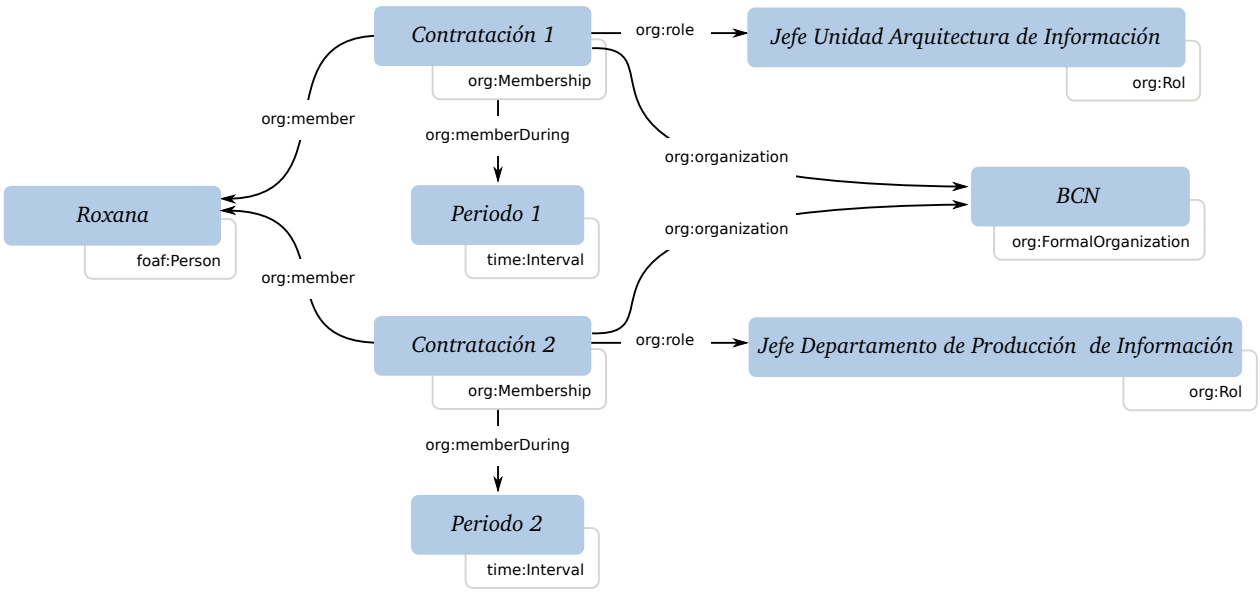


Figura 7.11: Ejemplo que muestra dos de las contrataciones de Roxana dentro de la BCN. Ambas contrataciones representan a dos perídos consecutivos en los cuales es contratada (aunque no se muestran las fechas para no sobrecargar el diagrama). Ambos períodos difieren en el rol, pero comparten la organización que es la BCN.

rdfs:subClassOf). Tal jerarquía de cargos, además de los problemas de confundir clases con instancias, tiende a definir vocabularios efímeros, que cambiarán tan pronto como cambién los ministerios (ver discusión en la sección 6.2.9).

7.6. Atributos de los ejercicios parlamentarios

Otro elemento en discusión es cómo agregarle atributos específicos a las instancias de org:Membership. Algunos ejemplos de atributos que se necesitan son el partido político del parlamentario al postularse al cargo y el territorio que está representando, actualmente distrito o circunscripciones, según se trate de diputados o senadores. Para ambas clases se propone utilizar de manera homogénea los predicados :affiliationWhenElected y :electedFromTerritory.

Con el objetivo de que estos predicados puedan ser usados en el contexto más general posible, incluso incluyendo la elección de cargos fuera del parlamento, como es el de presidente o alcalde, los rangos de ambos predicados serán abiertos. Con respecto a la afilición política, el vocabulario bcnbado los partidos políticos. En cambio, en esta tesis el vocabulario utiliza el predicado bcnb:hasPoliticalParty con rango en bcnb:PoliticalParty, una clase representado los

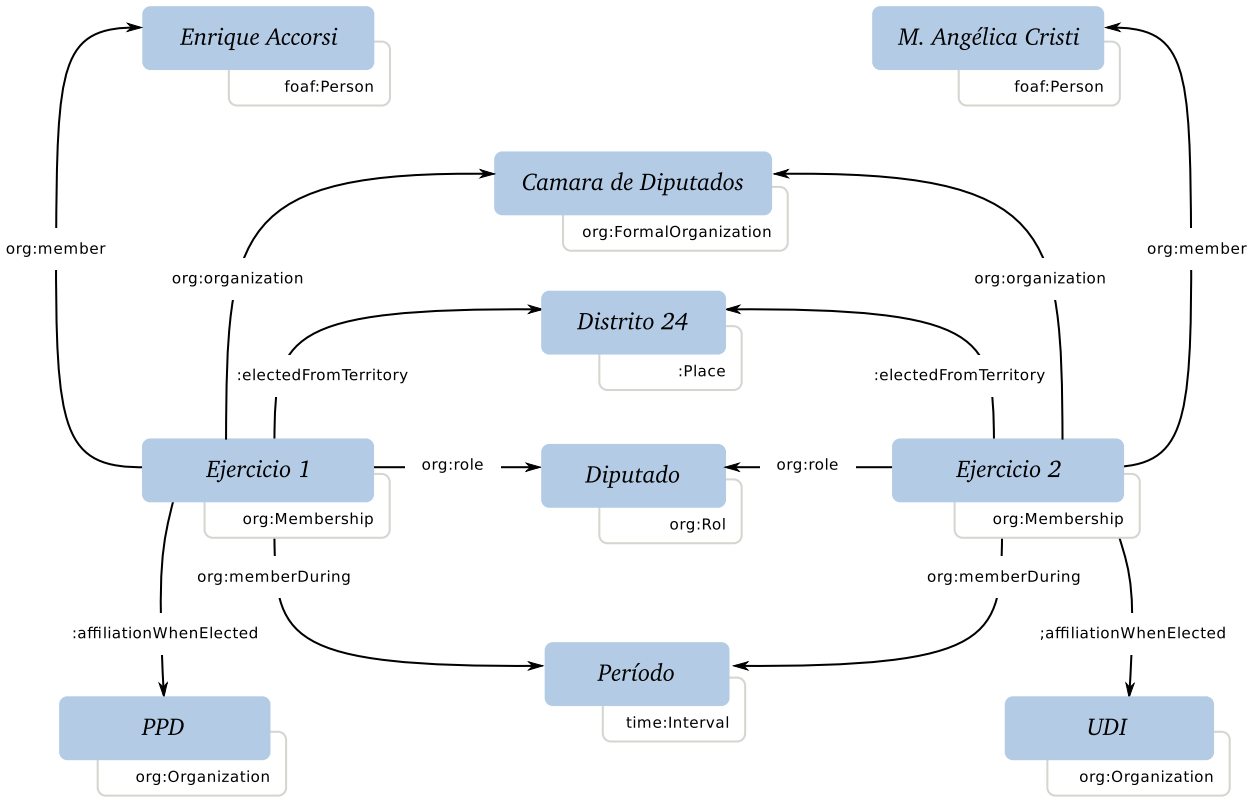


Figura 7.12: Ejemplo que muestra a dos diputados que comparten el cargo y el distrito, durante el mismo período parlamentario

partidos políticos. En cambio en esta tesis el vocabulario :affiliationWhenElected tiene rango indefinido para incorporar también las opciones de "independiente" o de algún movimiento político (práctica que era usada antes de que se reglamentaran los partidos). Mientras, en el caso del territorio, se ha escogido que rango también quede abierto pues no se dispone de la suficiente claridad como para escoger una clase apropiada para los lugares[2].

7.7. Atributos de las contrataciones

Las remuneraciones forman parte de la información que los organismos públicos están obligados a declarar por la Ley de Transparencia. En ella se definen cuatro tipos de contrataciones: *a*) de planta, *b*) a contrata, *c*) a honorarios y *d*) otras contrataciones por el código del trabajo. Esta taxonomía se define en un instructivo del Consejo para la Transparencia en que se presentan las disposiciones generales sobre qué información se debiera publicar como concepto

[2]Una decisión similar fue tomada en el vocabulario bio al no escoger un rango para el predicado bio:place.

de transparencia activa [147].

Para codificar estos tipos de contrataciones se propone usar un predicado :contractualMode cuyo recorrido sea la clase :ContractualMode que, siguiendo el patrón de los roles que usa el vocabulario org, se define como una subclase de skos:Concept. Esta estrategia difiere de la seguida en el vocabulario bcnt, donde se prefirió que el atributo equivalente, bcnt:contractualMode, tuviera como recorrido los literales de tipo xsd:string. La razón de usar instancias de skos:Concept es que éstas pueden ser organizadas en modelos conceptuales (ver sección 6.2.3).

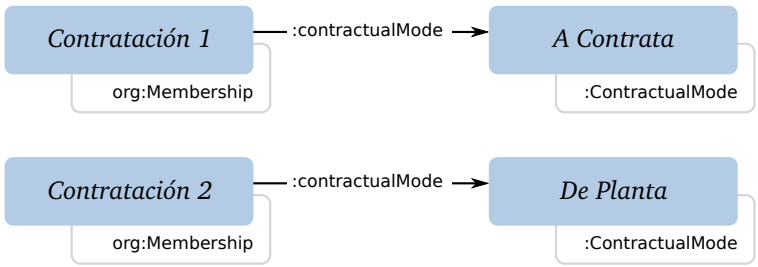


Figura 7.13: Ejemplo que muestra cómo dos contrataciones que aparecen pueden ser caracterizadas como a contrata o de planta.

En la ficha de Roxana que es publicada por la Biblioteca del Congreso Nacional aparece el cuadro 7.1. Los únicos datos que aún no son cubiertos por el vocabulario propuesto son el grado y el estamento. Siguiendo el mismo esquema que en el caso de la modalidad contractual, se definen los predicados :hasGrade y :hasStratum (ver figura 7.14).

Inicio	Término	Grado	Codigo Cargo	Cargo	Estamento
01/01/2009	31/12/2009		AT-JE-01	Jefe Unidad de Arquitectura de Información	
01/01/2010	31/12/2010	F	AT-JE-01	Jefe Unidad de Arquitectura de Información	Profesional
01/01/2011	31/12/2011	D	AT-JE-01	Jefe Unidad de Arquitectura de Información	Profesional
01/01/2012	30/04/2012	D	AT-JE-01	Jefe Unidad de Arquitectura de Información	Profesional
01/05/2012		D	PRI-JE-01	Jefe Departamento de Producción de Recursos de Información	Directivos

Cuadro 7.1: Histórico de cargos de Roxana dentro de la Biblioteca del Congreso Nacional.

7.8. Remuneraciones

Los datos en las planillas de las remuneraciones han sido precisados en una serie de normativas que sucesivamente han introducido correciones sobre las anteriores. En particular, dentro del

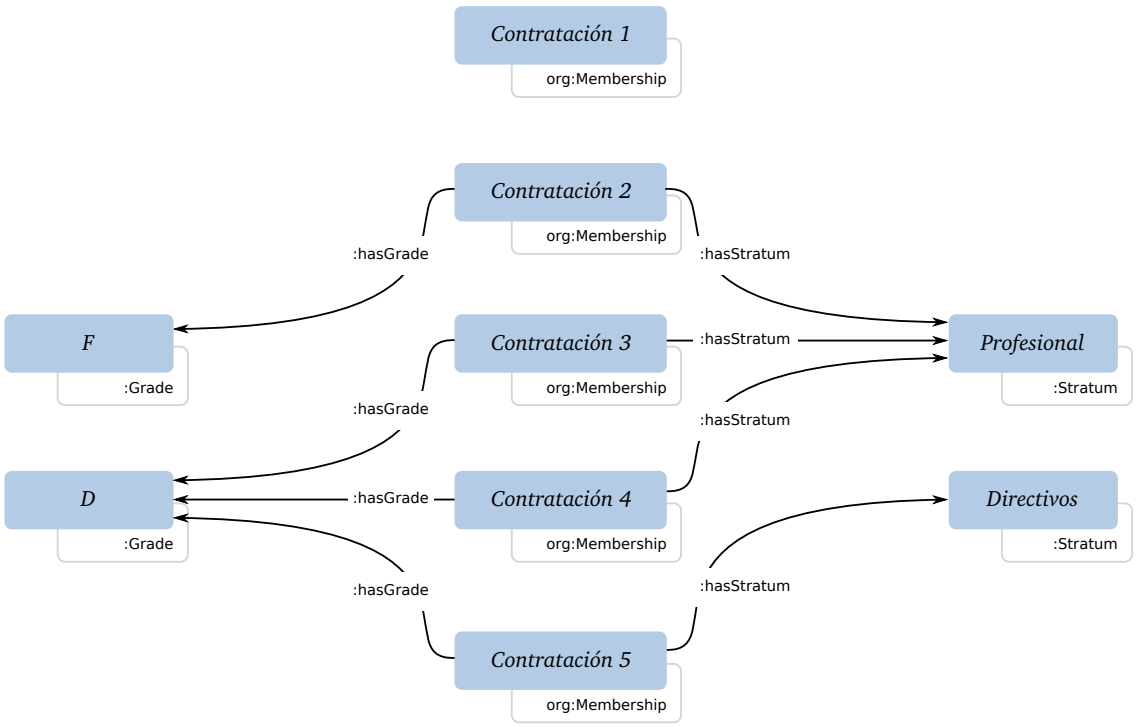


Figura 7.14: Ejemplo que muestra cómo las contrataciones pueden ser descritas respecto a su grado y estamento (predicados :hasGrade y :hasStratum). Además, muestra como en la primera contratación tales atributos no se encuentran presentes.

gobierno, el documento vigente que indica cómo publicar los datos de las remuneraciones (artículos 6 y 7 de la Ley Nº 20.285) es el Oficio Ordinario Nº 870 del Ministerio Secretaría General de la Presidencia, del 18 de junio de 2010. En la descripción de las remuneraciones se simplifica a sólo tres casos que son presentados en el cuadro 7.2.

De los atributos presentados en el cuadro 7.2 sólo los números 7, 9, 12, 13, 14 y 15 corresponden a atributos para los cuales aún no hemos definido cómo publicarlos.

Para codificar la calificación profesional o formación (atributo 7) utilizaremos un predicado :qualification cuyo rango será abierto. Nótese que el dominio de este predicado no es la persona sino la contratación, por ende, la cualificación corresponderá a la presentada al momento de ser contratado.

Para modelar la región (atributo 9) basta con seguir el patrón que se ha usado para los territorios que eligen a los parlamentarios y alcaldes (predicado :electedFromTerritory), es decir, basta con crear un predicado :worksInTerritory cuyo rango también será indefinido.

		De planta y a contrata	*A honorarios*	*Sujeto al código del trabajo*
1.	Tipo de contrato (planta o contrata)	X		
2.	Estamento (directivo, profesional, técnico, auxiliar o fiscalizador)	X		
3.	Apellido Paterno	X	X	X
4.	Apellido Materno	X	X	X
5.	Nombres	X	X	X
6.	Grado EUS	X	X	X
7.	Calificación profesional o formación (se deberá indicar el título técnico o profesional, grado académico y/o experiencia o conocimientos relevantes)	X	X	X
8.	Cargo o Función	X	X	X
9.	Región	X	X	X
10.	Fecha de inicio	X	X	X
11.	Fecha de término	X	X	X
12.	Observaciones	X	X	X
13.	Unidad monetaria		X	X
14.	Honorario bruto (En el caso de no ser mensual deberá indicarse el total y señalarse esto en el campo "pago mensual". En el caso de estar asimilado al grado deberá indicarse en la misma celda.)		X	X
15.	Pago mensual (si/no)		X	

Cuadro 7.2: Atributos en las remuneraciones según el Oficio Ordinario N° 870 del Ministerio Secretaría General de la Presidencia, del 18 de junio de 2010. Las X marcan los tipos de contrataciones en los que los atributos deben indicarse.

Para las observaciones (atributo 12) basta con usar el predicado rdfs:comment.

Las remuneraciones (atributos 13, 14 y 15) presentan un trato diferenciado que puede agruparse en tres casos: *a*) se especifica la remuneración mensualizada, *b*) se especifica la remuneración total y *c*) se especifica que la remuneración se hará de acuerdo a una tabla de remuneraciones. En cualquiera de los tres casos la diferencia se hará a través del tipo de objeto del predicado org:remuneration, que posee un rango abierto precisamente para permitir tal variedad.

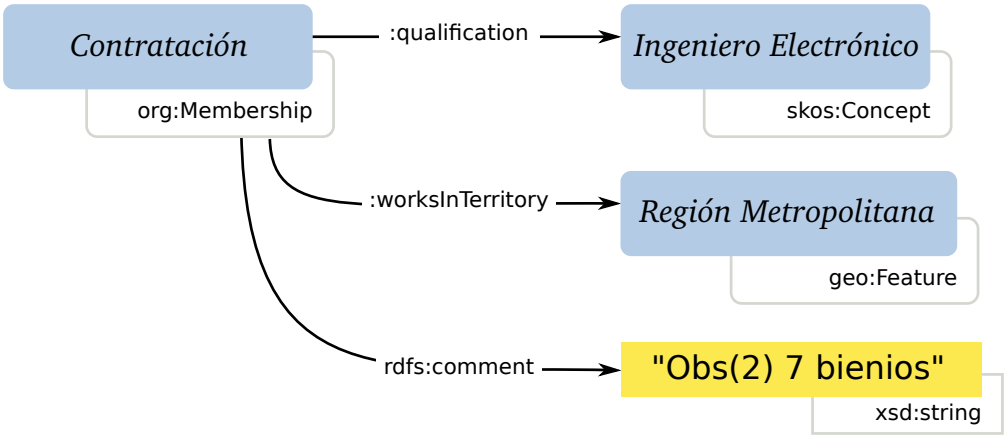


Figura 7.15: Calificación profesional, región y comentarios en una contratación.

El caso *c*) es el más complejo de codificar, pues se debe permitir que dados los atributos de la contratación pueda deducirse de manera automática la remuneración que le corresponde según la escala de reuneraciones asociada. Según el Oficio Ordinario N° 870 del Ministerio Secretaría General de la Presidencia, los registros de tales escalas deben ser: el estamento, el grado (o jornadas en el caso de los médicos), la unidad monetaria, las componentes de la remuneración y la remuneración bruta mensualizada. Además, se indica que las componentes a las que un funcionario accede deben indicarse en la columna observaciones. Es importante recalcar que la remuneración bruta percibida por la persona no será la que se indica en la tabla como total bruto, sino la suma del base más las asignaciones particulares a las que tiene derecho. Para explicar éste y otros detalles tomaremos como ejemplo el caso que presentamos en el cuadro 7.3[3].

Al pinchar sobre el enlace en la tabla del funcionario se abre una ventana con los datos que aparecen en el cuadro 7.4. Estas asignaciones corresponden a un desglose de parte de las remuneraciones recibidas por el funcionario, que suman $4.727.154 faltando sólo $626.107 para completar el sueldo recibido.

[3]Estos datos fueron tomados desde http://transparencia.mop.cl/dotacion/planta/2012/201206.html.

Estamento	Directivo
Apellido paterno	Beretta
Apellido materno	Riquelme
Nombres	Guillermo
Grado EUS	3
Calificación profesional	Ingeniero Comercial
Cargo o función	Secretario Regional Ministerial
Región	XV
Asignaciones especiales	Aquí se pone un enlace al cuadro 7.4.
Unidad monetaria	pesos
Remuneración bruta mensualizada	$5.353.261
Horas extraordinarias	NO
Fecha de inicio	16-04-2010
Fecha de término	Indefinida
Observaciones	Funcionario conserva cargo en la planta del Ministerio de Planificación Nacional; funcionario percibe asignación de función crítica de un 25 %; funcionario percibe asignación de responsabilidad de un 40 %.

Cuadro 7.3: Registro de remuneraciones de Guillermo Beretta Riquelme que aparece como directivo en la planilla de contrataciones del Ministerio de Obras Públicas para Junio de 2012.

Art.10 Ley 18675	$163.958	*A. Zona*	$272.302
Asignación Sustitutiva	$907.917	*Asignación Gastos Representación*	$0
Asignación Antigüedad	$0	*Asignación Responsabilidad*	$194.501
Asignación Profesional	$389.006	*Bonif. Compensatoria Ley 19553*	$0
Bonificación Única Ley 18717	$0	*B. Colectivo*	$474.642
Base Adicional Art. 11	$0	*B. Institucional*	$450.909
Bonif. Art. 19	$0	*Función Crítica*	$803.436
Comp. Base Ley 19553	$889.956	*Alta Dirección*	$0
Asignación. D.L. 3551	$0	*Dirección Superior*	$0
Asignación Maquinaria Pesada	$0	*Asignación Ley 19699*	$0
A. Pérdida de Caja	$0	*A. Zona Extrema*	$180.527

Cuadro 7.4: Asignaciones especiales de Guillermo Beretta Riquelme en junio de 2012. Estos datos fueron tomados de las publicaciones del Ministerio de Obras Públicas y corresponden a valores específicos para el funcionario.

La figura 7.16 ejemplifica cómo modelar la publicación explícita de la remuneración y de sus componentes. A diferencia de lo que recomienda el vocabulario org, usar la clase gr:UnitPriceSpecification, hemos preferido usar la clase intermedia :Remuneration. La motivación de esto es que permite aislar el concepto de monto de dinero al igual que se hace uso de las instancias temporales en la ontología time. Además, es necesario advertir, que aunque se haya usado el vocabulario gr este está orientado a describir compra y venta de productos, por lo que la semantica de sus predicados se alejan de la representación de datos en conceptos como los de las contrataciones públicas.

El vocabulario de la Biblioteca del Congreso Nacional para transparencia, bcnt, usa una estrategia diferente para la definición de los sueldos; para ello define la clase bcnt:Category de

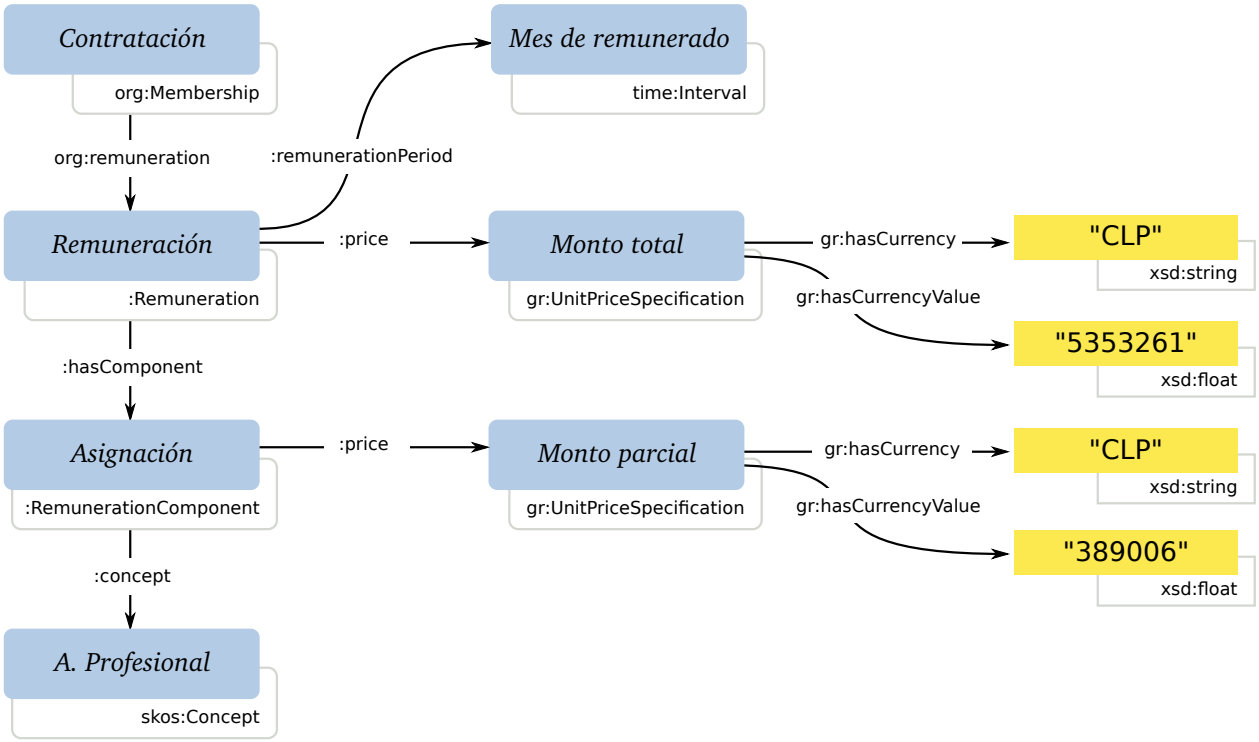


Figura 7.16: Descripción de una remuneración en la que se especifica directamente su monto bruto y también una de las componentes de este.

la cual salen una serie de predicados cuyo rango es un terminal que especifica un monto. De este modo las contrataciones se asocian a las categorías para poder especificar los montos de remuneraciones de todas las contrataciones que se encuentren asociadas a la misma categoría. La razón por la que hemos descartado este esquema es que genera demasiados predicados, muchos de los cuales son particulares del organismo en cuestion. Dicho de otro modo, el esquema que proponemos en esta tesis prioriza la generalidad. Además, el vocabulario propuesto puede también entregar la funcionalidad que entrega bcnt; para ello basta que más de una contratación comparta la misma remuneración.

La fortaleza del vocabulario bcnt es que permite definir escalas de remuneraciones en forma de conjuntos de categorías, es decir, entrega una forma de publicar las remuneraciones en el esquema *c*) cuando éstos deben deducirse al hacer un *join* entre los atributos de la contratación con los atributos que son llave en las escalas de remuneraciones. En general, estos atributos son el grado EUS y el estrato, por lo que la propuesta de esta tesis es modelar tales escalas siguiendo el esquema que se propone en la figura 7.17.

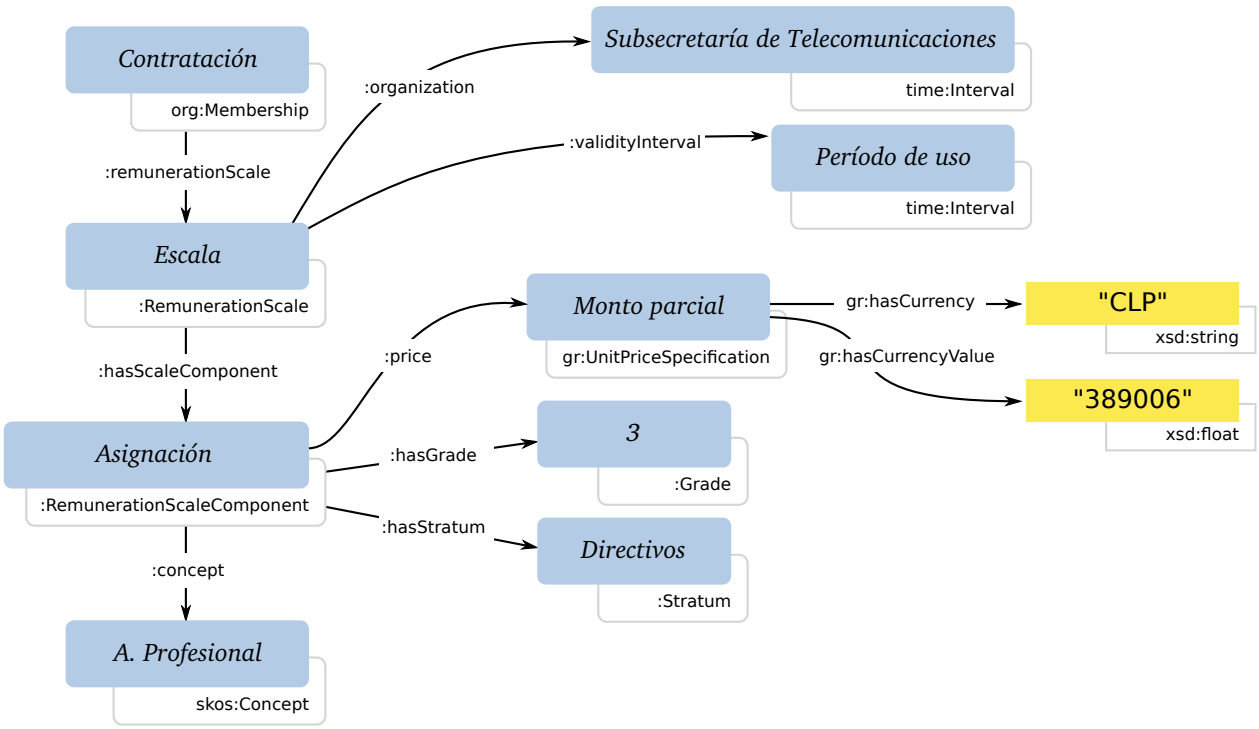


Figura 7.17: Descripción de una escala de remuneraciones. En este ejemplo una contratación se asocia a la escala, lo que facilita la búsqueda de las componentes de su remuneración a partir del grado y del estrato. En el caso contrario, si no se indica la escala que rige una remuneración ni sus componentes de manera directa, la única opción es utilizar el rango de tiempo sobre el cual la escala es válida y la unidad correspondiente.

7.9. Partidos políticos

En la publicación de las Biografías parlamentarias se indica cuales son los referentes de los partidos políticos en su formación, es decir, se describe la formación de un partido a partir de otros. Esta relación es modelada en el vocabulario bcnb mediante el predicado bcnb:hasReferring. Sin embargo, tal predicado no es necesesario pues en el vocabulario en el cual se basa org ya se describe cómo realizar tales derivaciones, que tratan a las organizaciones como instancias de opmv:Artifact. En particular, el predicado opmv:wasDerivedFrom cumple la misma función que bcnb:hasReferring.

Otro de los elementos presentes en el vocabulario de las biografías es el uso del predicado bcnb:composedBy, que es usado para definir coaliciones de partidos. No obstante, tampoco este predicado es necesario, pues el vocabulario org posee predicados más precisos que pueden ser utilizados en su lugar: org:memberOf y org:hasSubOrganization y org:subOrganizationOf. No obstante, sólo recomendamos el uso de estos predicados cuando la vida de las coaliciones

esté acotada a un punto en el tiempo, por ejemplo, durante una elección. Para coaliciones que se extiendan en el tiempo se recomienda agrupar la información de las pertenencias a ella mediante instancias de la clase org:Membership, que como ya hemos presentado en más de una ocasión, permiten señalar sus tiempos de inicio y término.

Capítulo 8

Conclusiones

A lo largo de esta tesis se propuso un modelo que permite distinguir entre los conceptos de documento y dato. Este modelo ha sido empleado para contextualizar las prácticas de publicación que luego se presentaron en forma de recomendaciones, organizadas en tres niveles de generalidad: la publicación de datos en general, la publicación con RDF y la publicación con un conjunto de vocabularios de RDF. Cada uno de estos niveles de generalidad de las recomendaciones logra uno de los objetivos específicos de la tesis. Con ellos, esta tesis cumple con sentar un precedente para la discusión de prácticas de publicación de datos públicos en Chile.

Las prácticas y el vocabulario propuestos en esta tesis no han sido presentados aún a los organismos que podrían usarlos. Los datos que motivan el vocabulario son publicados por los organismos del Gobierno (transparencia) y por la Biblioteca del Congreso (transparencia y biografías parlamentarias). Los primeros se encuentran en documentos PDF y HTML, siendo algunos procesados para poblar una base de datos central que es mantenida por el gobierno. Los segundos, son convertidos a RDF y se ofrece un acceso a ellos mediante un SPARQL endpoint. Sin embargo, tal como se discutió a lo largo de la presentación del vocabulario propuesto, el vocabulario usado actualmente por la BCN define términos demasiado particulares que podrían dificultar la integración con las otras fuentes de datos. En cambio, el vocabulario propuesto se basa en el practica de modelar con los datos y no con el esquema, con el fin de facilitar la integración.

Un trabajo futuro, que deberá hacerse para facilitar la publicación de datos por parte del Gobierno y otras organizaciones, es la difusión de las prácticas propuestas mediante la con-

fección de una guía de publicación. Esta guía, a diferencia de esta tesis, que pone énfasis en la discusión de las prácticas, deberá presentar las prácticas y el vocabulario de manera resumida incorporando sólo aquellos detalles necesarios para la implementación.

En las experiencias de modelamiento y observación de vocabularios desarrollados se observó que RDF resulta novedoso para muchos que están habituados a modelar datos con otros modelos. Por ello, para entenderlo se suelen arrastrar las estrategias de los modelos conocidos al modelo RDF. En varios casos se observó problemas cuando no se abordan las particularidades propias del modelo RDF. Por ejemplo, en el modelo relacional es común intentar que el modelo acote las posibilidades a lo que es permitido en el dataset. En cambio, en RDF lo apropiado es no introducir restricciones particulares y, de ese modo, permitir que el vocabulario pueda adaptarse a una mayor cantidad de usos.

En la sección 8.1 se presentarán algunos conflictos surgidos al arrastrar prácticas desde el lenguaje natural y en la sección 8.2 se presentarán conflictos que surgen de intentar aplicar las prácticas usadas en el modelo relacional y en la definición de clases de objetos. Los conflictos detectados no sólo implican la necesidad de contar con técnicas y patrones de modelamiento para RDF, sino que también discuten la conveniencia de las características actuales que presenta el modelo RDF.

Por último, las conclusiones terminan con una discusión sobre la importancia que tiene el establecer relaciones entre la información y los datos (sección 8.4) para la publicación de datos generados por comunidades.

8.1. RDF y el lenguaje natural

Una de las formas con las que las personas son introducidas al lenguaje RDF es a través de sus semejanzas con el lenguaje natural. Las tripletas son entendidas como frases, con un sujeto, un objeto y un predicado, por lo que se genera la impresión de que se está escogiendo una gramática precisa y formal para poder expresar las ideas que los seres humanos pensamos en un lenguaje más sencillo y preciso, que puede ser procesado por máquinas. No obstante, si bien la comparación puede ser útil para adquirir una intuición inicial, conlleva también la tentación de introducir tripletas de la misma forma en las que se generarían las frases, sin tener en cuenta las diferencias.

8.1.1. Sustantivación

En el lenguaje natural no es necesario decir cuándo un hecho ocurre o en qué lugar, pues podemos acompañar nuestras frases con otras que agregarán los detalles. De este modo, nuestras frases no pueden ser tomadas fuera del contexto en el que las decimos ni cambiar el orden de las oraciones; el lenguaje usa estas herramientas para que no tengamos que incurrir en redundancias innecesarias para nuestras capacidades humanas. A diferencia de ello, en el modelo RDF el orden con el que las tripletas son dispuestas en un archivo no influye en el mensaje. Dicho de otro modo, los datasets en RDF son vistos como conjuntos, y como tales, la unión es la operación que nos permite integrar los datos. De esta manera, los átomos del modelo RDF, las tripletas, toman valor sin requerir de un contexto. Por ejemplo, la oración "Eran las 12:00 cuando Alicia tomó el tren a vapor rumbo a Curicó" predispone a separar el relato en las siguientes oraciones: {Alicia tomó el tren X / El tren X salió a las 12:00 / El tren X funciona a vapor / El tren X viaja a Curicó}. Si bien tales oraciones parecen tripletas, no pueden ser entendidas como tales, pues sus interpretaciones dependen del contexto. En efecto, si unimos estos datos con los de otro viaje de Alicia, en el mismo tren pero con diferente destino y horario, nos quedará una historia como la siguiente: {Alicia tomó el tren X / El tren X salió a las 12:00 / El tren X funciona a vapor / El tren X viaja a Curicó / El tren X salió a las 11:00 / El tren X viaja a Chillán}, en la cual perdemos la capacidad de discernir entre los horarios en que partió a cada ciudad. En consecuencia, debiera haber una instancia para representar el viaje y otra que nos indique la máquina que realiza el viaje. De este modo, el viaje puede tener como atributos su fecha y destino y, a su vez, podemos caracterizar al tren como a vapor: {Alicia tomó el viaje V / el viaje V usó la máquina X / El viaje V partió a las 12:00 / El tren X funciona a vapor / El viaje V va a Curicó / Alicia tomó el viaje U / El viaje U usó la máquina X / El viaje U partió a las 11:00 / El viaje U va a Chillán }. Este ejemplo muestra que en el lenguaje natural en más de una ocasión se usa el mismo identificador para describir a más de un recurso (la máquina y el viaje) y que nuestras capacidades humanas nos permiten entender a cuál de los recursos corresponde cada atributo sin necesidad de agregar identificadores extra.

El que las tripletas sean libres de contexto implica que los predicados N-arios deben ser tratados de una manera especial. Ya no podemos decir que una persona integra una determinada organización (verbo), sino que debemos crear una instancia (sustantivo) que modele la instancia de dicha relación; instancia sobre la que agregaremos atributos que podrán caracterizarla. Existen otras dos estrategias, aparte de la sustantivación, para entregar el contexto faltante a las tripletas, una es la reificación y la otra es contextualizar las tripletas a través

de la descripción del grafo mediante el cual se publican. Sin embargo, dado que minimiza las funcionalidades requeridas en los lenguajes de consulta, la sustantivación de los predicados es muchas veces la preferida.

A pesar de que en muchas ocasiones pareciera evidente la necesidad de sustantivar, hay vocabularios populares que poseen relaciones como rel:neighborOf, rel:worksWith, org:headOf y foaf:member, que no precisan los rangos temporales en los cuales son válidas, pese a que se evidencia que son afirmaciones cuya validez es acotada en el tiempo. Esto nos lleva a la conclusión de que no basta con la popularidad de los lenguajes, debemos observar cuidadosamente la forma en la que éstos están construidos para no incorporar tripletas que introduzcan complejidades innecesarias al momento de la integración por olvidar que en su modelo las tripletas están libres de contexto.

8.1.2. Sobrecarga de significados

En el lenguaje natural la interpretación de los términos depende del contexto en el cual éstos son usados. Por ejemplo, el verbo tomar tiene dos significados distintos entre "Alicia tomó el tren" y "Alicia tomó el lápiz". En el caso de RDF es deseable que todos los predicados posean un significado claro, no obstante, hay vocabularios en los que los predicados pueden tener significados distintos según el recurso que actúe como sujeto. Por ejemplo, en Good Relations cuando el predicado gr:elegibleDuration tiene como sujeto a una instancia de gr:License sabemos que se trata de la duración del período durante el cual se puede usar un producto luego de su compra, en cambio, si el sujeto es una instancia de gr:Offering, el período corresponderá a la vigencia de la oferta.

Antes de discutir sobre la conveniencia de tal sobrecarga de significados en un predicado, es necesario discutir qué entendemos por significado y recordar la discusión planteada por Booth en la que afirmaba que el problema no es qué es lo que significan los recursos, sino para qué los usamos.

8.1.3. La evolución de los vocabularios

Nuestro lenguaje evoluciona constantemente. Basta comparar un libro actual con uno de hace más de 100 años para notar que han surgido palabras nuevas, otras se han dejado de

usar y otras se usan con otros significados. Tal variación no ha sido discutida en el ámbito de RDF. Muchos vocabularios agrupan los términos en URIs con un prefijo común, prefijos que contienen la versión del vocabulario. De este modo, al hacer modificaciones del vocabulario se publica un nuevo conjunto de términos. Esta política, común en el desarrollo de APIs y en la gestión documental, tiene la ventaja de que permite estudiar los vocabularios históricos y que los datos codificados en un vocabulario anterior se mantengan en los términos bajo los cuales fueron definidos, eliminando el riesgo de introducir falsas interpretaciones producto de un cambio en los significados. Sin embargo, el aspecto negativo es que los datos quedan separados por las versiones de los vocabularios que usan, agregando complejidad en la tarea de integrar los datos. En conclusión, el tema de cómo gestionar la evolución de los vocabularios RDF se encuentra aún abierto.

8.1.4. Las palabras tienen dueños

En el lenguaje natural, quien inventa una palabra no puede impedir que otros la usen ni obligar a los demás a usarla de una manera determinada. A pesar de haya academias de lengua que parecen desear tener el control sobre las definiciones de las palabras y sus usos, y empresas que pagan por ser dueñas de ellas en forma de marcas, en el uso común las palabras no tienen dueño. La cita "la palabra es mitad de quien la pronuncia, mitad de quien la escucha" atribuida a Montaigne, da cuenta de esta idea.

Contrario al carácter público de las palabras, en RDF los términos tienen dueño: quienes se encargan de gestionar la dereferenciabilidad de sus URIs. La dereferenciabilidad es una de las cualidades centrales de las URIs. Sin embargo, la propiedad de las URIs puede ser también un freno a que los datos sean realmente públicos. Cuando usamos el identificador foaf:Organization aceptamos las convenciones tomadas por la gente que desarrolló el vocabulario FOAF y les entregamos el poder de hacerse cargo del vocabulario. Consecuencia de ello es la aparición de clases repetidas con el mismo significado y sufijo, que son creadas como un *proxy* a otras que ya existían. Por ejemplo, se crea la clase org:Organization que es subclase de foaf:Agent, siendo que ya existía una clase foaf:Organization y la Biblioteca del Congreso crea su propia clase bcnt:suborganization que deriva de org:OrganizationalUnit. La creación de clases proxy es también común en el mundo de la orientación a objetos. El objetivo es usar una biblioteca de clases sin depender demasiado de ella. De este modo, podrá ser más simple cambiar las clases que hacen de base a las estructuras que hemos definido sin cambiar la estructura interna de nuestro modelo. Sin embargo, tal práctica contradice el objetivo central

de los vocabularios en RDF, el de facilitar la integración a través del uso de un vocabulario común.

Esta tensión de poder que surge desde el uso de dominios propios podría llegar a jugar en contra de la interoperabilidad de los datos. En particular, en Chile, podría vislumbrarse que, de instalarse el modelo RDF como un estándar de publicación de datos abiertos, los organismos terminen creando sus propios vocabularios, cada uno queriendo mantener sus cuotas de control. En conclusión, modelar datos con RDF no se trata sólo de enseñar una tecnología, sino también de tener políticas para abrirse a compartir los esfuerzos.

Un ejemplo aún más extremo es el de vocabularios como el de *Good Relations*, que presenta un modelo cerrado sobre sí mismo, en el que los predicados usan como dominio y rango a las clases definidas dentro del mismo vocabulario (salvo excepciones como el uso del vocabulario Schema.org). Good Relations sigue una práctica que lo hace especialmente cerrado: define rangos y dominios de varios predicados como uniones de clases a través de la estructura owl:unionOf. Por ejemplo, el predicado gr:elegibleDuration tiene como dominio a la unión de las clases gr:License, gr:Offering y `http://schema.org/Offer`. Este predicado es usado para identificar las duraciones de licencias u ofertas de venta, es decir, posee como dominio la unión de dos clases que no heredan de una clase en común. Este tipo de construcción no facilita el que el predicado pueda ser usado en casos en los que las instancias no se asemejen completamente a las clases del dominio.

El uso de la unión de clases sin una clase base en dominios de predicados es un patrón de modelamiento sospechoso. Inmediatamente surge la pregunta de si no sería mejor modelar tal predicado mediante dos predicados independientes, cada uno con su propio dominio. En cierto modo, el predicado se comporta como si tuviera una semántica distinta según la clase (ver sección 8.1.2).

Para evitar que los vocabularios tengan dueños habría que utilizar un sistema de identificadores dereferenciables que no dependieran de las URIs. En un momento OASIS propuso los identificadores XRI para tal propósito, pero fueron rechazados como estándar por el TAG del W3C [148, 149, 150]. Podría ser que en algún momento se reabra la discusión para proponer formas de dereferenciar los identificadores usando sistemas que no dependan de las URIs.

8.2. RDF y otros modelos

Una segunda fuente de tentaciones en el modelamiento con RDF son las prácticas que se arrastran del modelo relacional. Hay varias diferencias entre ambos modelos: *a*) En el modelo relacional, visto como una transformación del modelo ER, cada instancia lo es de una única entidad, en cambio, en RDF un recurso puede pertenecer a múltiples clases. *b*) En el modelo relacional se modela un universo particular, en cambio, en RDF se modela una parte de un universo mayor. *c*) Validar que una tabla satisfaga un esquema relacional es una tarea trivial, en cambio, validar que una tabla valide definiciones expresadas en RDF Schema o en OWL es una tarea compleja.

8.2.1. La tentación de restringir

Para quienes vienen de trabajar con bases de datos, especialmente del modelo relacional, pueden caer en la tentación de establecer esquemas que impongan el mismo tipo de restricciones que estaban acostumbrados a definir en las bases de datos relacionales, donde la consistencia muchas veces juega un rol prioritario. En cambio, en RDF la consistencia aún es un tema en discusión y muchos vocabularios tienden a ser lo más abiertos posible en los dominios y rangos de sus predicados con el fin de ampliar las posibilidades de uso. La diferencia clave entre el modelamiento en el modelo relacional y el modelamiento en RDF es que, mientras las bases de datos relacionales son hechas para describir dominios pequeños, las publicaciones de datos RDF tienen como objetivo la integración con otras fuentes de dato, constituyendo componentes de un dominio abierto. De este modo, en el modelamiento de RDF el foco debe ser generar vocabularios que nos permitan expresar nuestro dataset, antes de restringir el vocabulario para que no se puedan expresar datos que no sean consistentes con las reglas que observamos en él. Al hacer esto, será posible que el vocabulario que hayamos creado pueda ser usado para describir otro dataset, en el cual quizá tales reglas no se cumplan.

El dilema entre cuán general debe ser nuestro vocabulario pasa por distinguir entre cuáles son las propiedades generales que son compartidas por múltiples datasets y cuáles son las particulares del nuestro; límite que es imposible de saber pues no podemos conocer de antemano todos los datasets existentes ahora y que aparecerán en el futuro. Por ejemplo, si nuestra organización tiene tres niveles organizacionales, podríamos querer definir tres clases de unidad organizacional, una para cada nivel. Sin embargo, este modelamiento no nos ser-

virá para describir organizaciones con un diferente número de niveles. En cambio, si definimos una sola clase para todas las unidades organizacionales nos bastará un predicado para indicar cuando una unidad es parte de otra para poder deducir los niveles organizacionales y nuestro vocabulario servirá para un número mayor de casos.

8.2.2. La tentación de heredar

Una variante de la tentación de restringir es la tentación de derivar una clase por la sola razón de definir un nuevo atributo en ella. Por ejemplo, una clase general para organizaciones A puede ser extendida a otra clase B de las personas jurídicas chilenas, a la que se agrega el atributo RUT. De este modo, el predicado agregado tendrá como dominio a la clase B. Esta práctica se asemeja a la extensión de clases en programación orientada a objetos. Si bien el modelo parece correcto, usar clases derivadas dificulta la integración de los modelos, pues mientras más clases haya mayor será el esfuerzo de comprender el modelo.

Cuando tenemos una gran cantidad de instancias con un conjunto de atributos $a_1, a_2, \ldots, a_n$ en las cuales la asignación de atributos es heterogénea, resulta natural querer definir clases para cada combinación de atributos. El deseo de definir clases resulta mayor cuando poseemos reglas que caracterizan los atributos para determinados tipos de instancias, como ocurre con los tipos de contrataciones en la transparencia activa. A su vez, el contar con clases ya definidas nos puede llevar a querer ser riguroso en la definición de los dominios de los predicados. Por ejemplo, podemos tener las clases disjuntas para las contrataciones: A, B y C, donde el predicado p tiene como dominio a $A \cup B$ y el predicado c tiene como dominio a $B \cup C$. Sin embargo, aunque precisa, tal definición nos lleva a que cuando aparezca una nueva clase D (por ejemplo sujeta a un cambio en la Ley) que deba agregarse al dominio de p, debamos redefinir el dominio de p. En cambio, si hubiésemos tenido una única clase X para todas las contrataciones, no necesitaríamos tal esfuerzo en la mantención del vocabulario, pues independientemente de los tipos de contrataciones, los atributos tendrán a X como dominio.

8.3. Patrones de modelamiento en RDF

Cuando observamos que los vocabularios usan diversas estrategias de modelamiento, nos damos cuenta de la falta de patrones para el modelamiento en RDF. Además, el caso de RDF es más complejo que el modelamiento general en bases de datos, pues mientras las bases de datos están orientadas a resolver un problema particular, los datos en RDF tienen la misión de integrarse con datos de otras fuentes. El problema es que muchas veces la calidad de un vocabulario no guarda relación con sus propiedades internas, sino con cómo se integra con otros vocabularios y a cuáles ha escogido para integrarse, de entre vocabularios que se solapan en sus ámbitos.

En el caso de la programación orientada a objetos, contamos con métricas para comprender cuándo un modelo es reusable, cuándo las clases se encuentran demasiado acopladas o cuándo una clase es candidata a ser dividida en dos. Resulta natural preguntarse si métricas similares pueden desarrollarse para evaluar la calidad de los vocabularios RDF.

Uno de los factores que más dificulta la creación de patrones de modelamiento que sean ampliamente aceptados es que el modelo de RDF posee muchas maneras de hacer las cosas; más que el modelamiento usando el modelo EAV/CR. Por ejemplo:

1. Las afirmaciones pueden ser descritas mediante reificación, grafos o usando la sustantivación de predicados.

2. Podemos codificar los datos como instancias de clases (p. ej. time:Instant) o mediante la sintaxis de los literales (p. ej. xsd:date).

3. Se puede modelar usando muy pocas clases básicas y construir sobre ellas objetos heterogéneos o agrupar objetos similares dentro de clases más particulares.

4. Es posible modelar con los predicados en un sólo sentido o agregar los inversos de cada uno.

5. Se pueden especificar clases para los dominios y recorridos de los predicados o éstas se pueden dejar abiertas.

6. Se pueden sobrecargar semánticamente los predicados, agregando interpretaciones diferentes según las clases de las instancias que hacen de dominio y recorrido.

7. Se pueden definir atributos como literales o como conceptos del vocabulario SKOS.

8. En varios casos escoger el vocabulario para reusar o basarse no resulta sencillo, pues hay varios vocabularios con roles similares, como por ejemplo el OPMV[1] y el X-Prov[2]. Ambos vocabularios tienen como objetivo describir cambios sufridos en artefactos por la acción de agentes, sin embargo poseen diferencias menores, como por ejemplo, que el primero usa la clase time:Instant para los tiempos, mientras el segundo el tipo xsd:dateTime.

Varios de los aspectos mencionados anteriormente hacen que RDF sea una tecnología compleja, que requiere una formación extra en quienes publicarán y consumirán los datos.

Esta tesis, con las prácticas recomendadas para modelar en RDF, contribuye a traer el tema del modelamiento a la discusión. Se hacen necesarios una discusión aún más detallada sobre las prácticas de modelamiento y más estudios empíricos sobre cómo se están ensamblando y usando los vocabularios.

La breve historia de la creación de vocabularios para RDF está mostrando lo que ya sabíamos: "no existe un modelo universal para expresar los datos". Como consecuencia de esta observación, surge la necesidad de transformar datos de un modelo a otro. Una orientación es seguir los mismos conceptos de los *mappings* del mundo relacional y otra es intentar considerar que los datasets expresados con vocabularios distintos se integran mediante reglas de deducción. La diferencia entre ambas opciones es que la primera trabaja sobre grafos RDF que pueden ser transformados desde un vocabulario a otro, mientras en el segundo, se opera al interior de un sólo grafo visto como el producto de la disolución de las fronteras de los grafos. El problema de la primera estrategia es que volvemos al modelo de concentrar los datos de manera centralizada, en cambio, el segundo se enfrenta al problema de la complejidad de ir traduciendo los datos a medida que se los navega. El problema está abierto y su resolución es crucial para la integración de los datos.

8.4. De información a datos

El modelo propuesto por Tuomi [65], en el cual los datos son un paso posterior de convertir conocimiento en información e información en datos, fue practicado en el caso de las biografías

[1] http://purl.org/net/opmv/ns
[2] http://vocab.deri.ie/w3p

parlamentarias de la Biblioteca del Congreso Nacional, donde a partir de las fichas de los parlamentarios se detectan datos comunes que son susceptibles de ser usados para sumarizar la información. Tales datos se convierten en objetos con valor por sí mismos en la medida en que construimos herramientas que nos permiten operar automáticamente sobre ellos. De este modo, la relación entre la información y los datos no es algo que deba perderse. El linaje de los datos, explicitado manteniendo enlaces hacia sus fuentes y mediante la preservación de ellas, es parte fundamental de los sistemas de datos que son construidos y mantenidos por personas. Por ello, tal relación debiera ser estudiada en proyectos locales como el de la Curicopedia, la Poderopedia, Verdata y una futura Chilepedia[3]. Estos proyectos se beneficiarán permitiendo a la comunidad registrar información, enlazarla y etiquetarla. En general los datos y la información publicados por los organismos públicos carecen de la posibilidad de que la comunidad los reuse in situ. Nos vemos obligados a copiarlos o enlazarlos desde otro sistema en el cual datos e información sí puedan ser trabajados. Por ejemplo, si vemos los datos de los archivos judiciales, de los estudios de impacto ambiental o de la superintendencia de valores y seguros, encontraremos un sin número de datos que probablemente por razones más políticas que técnicas se entregan en una forma que resulte inútil a los intentos de escrutinio. En consecuencia, la sociedad civil tendrá un rol principal en la integración de tales fuentes de datos.

[3]La Curicopedia es un proyecto que busca la publicación de datos sobre Curicó (http://curicopedia.org). La Poderopedia es un proyecto que investiga relaciones entre personas que ostentan el poder en Chile que aún no ha sido lanzado (http://poderopedia.org). Verdata y la Chilepedia son también dos proyectos en construcción que tienen el objetivo de visualizar datos y de curar información y datos.

Bibliografía

[1] A.J.G. Hey, S. Tansley, and K.M. Tolle. *The fourth paradigm: data-intensive scientific discovery*. Microsoft research Redmond, WA, 2009.

[2] John F. Gantz, David Reinsel, Wolfgang Schlichting, John McArthur, Stephen Minton, Irida Xheneti, Anna Toncheva, and Alex Manfrediz. The expanding Digital Universe. 2007. Accesible en http://www.emc.com/collateral/analyst-reports/expanding-digital-idc-white-paper.pdf.

[3] Tim O'Reilly. What Is Web 2.0. 2005. Accesible en http://oreilly.com/web2/archive/what-is-web-20.html.

[4] R. Agrawal et al. The Claremont Database Research Self-Assessment Meeting. 2008. Accesible en http://db.cs.berkeley.edu/claremont.

[5] A. Szalay and J. Gray. 2020 computing: Science in an exponential world. *Nature*, 440(7083):413–414, 2006.

[6] A. Szalay G. Bell, T. Hey. Beyond the Data Deluge. *Science*, 323:1297–1298, Marzo 2009.

[7] D. Lazer, A. Pentland, L. Adamic, S. Aral, A.L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, et al. Computational social science. *Science*, 323(5915):721–723, 2009.

[8] Tim Berners-Lee and R. Cailliau. WorldWideWeb: Proposal for a HyperText Project, noviembre 1990. Disponible en http://www.w3.org/Proposal.html.

[9] T. Berners-Lee, R. Fielding, and L. Masinter. Uniform Resource Identifier (URI): Generic Syntax. Technical report, Enero 2005. Disponible en http://tools.ietf.org/html/rfc3986.

[10] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol — HTTP/1.1. Technical report, Junio 1999. Disponible en http://www.ietf.org/rfc/rfc2616.txt.

[11] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[12] J. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. The web as a graph: Measurements, models, and methods. *Computing and Combinatorics*, pages 1–17, 1999.

[13] R. Albert, H. Jeong, and A.L. Barabási. The diameter of the world wide web. *Arxiv preprint cond-mat/9907038*, 1999.

[14] T. Berners-Lee. The World Wide Web — Past Present and Future: Exploring Universality, Commemorative Lecture, 2004. Accesible en http://www.w3.org/2002/04/Japan/Lecture.html.

[15] Chris Anderson and Michael Wolff. The Web Is Dead. Long Live the Internet. Septiembre 2010. Accesible en http://www.wired.com/magazine/2010/08/ff_webrip/all/1.

[16] Rob Beschizza. Is the web really dead? *BoingBoing*, Agosto 2010. Accesible en http://boingboing.net/2010/08/17/is-the-web-really-de.html.

[17] Chris Anderson, Tim O'Reilly, and John Battelle. The Web Is Dead? A Debate, Agosto 2010. Debate publicado en http://www.wired.com/magazine/2010/08/ff_webrip_debate/all.

[18] Tim Berners-Lee. Long Live the Web: A Call for Continued Open Standards and Neutrality: Scientific American. *Scientific American Magazine*, Noviembre 2010.

[19] Eli Paliser. Beware online "filter bubbles", 2011. Accesible en http://www.ted.com/

talks/eli_pariser_beware_online_filter_bubbles.html.

[20] Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. Technical report, W3C, Febrero 1999. Accesible en http://www.w3.org/TR/1999/REC-rdf-syntax-19990222.

[21] Tim Berners-Lee. Notation 3 Logic. Agosto 2005. Publicado en http://www.w3.org/DesignIssues/Notation3.html.

[22] Tim Berners-Lee, Sandro Hawke, and Dan Connolly. Primer: Getting into RDF & Semantic Web using N3. Agosto 2005. Publicado en http://www.w3.org/2000/10/swap/Primer.

[23] P.M. Nadkarni, L. Marenco, R. Chen, E. Skoufos, G. Shepherd, and P. Miller. Organization of heterogeneous scientific data using the eav/cr representation. *Journal of the American Medical Informatics Association*, 6(6):478–493, 1999.

[24] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[25] R. Angles and C. Gutierrez. Survey of graph database models. *ACM Computing Surveys (CSUR)*, 40(1):1, 2008.

[26] Drummod Reed and Markus Sabadello. The XDI RDF Model. Technical report, 2006.

[27] E. Cano and G. Burel. XDI Trust Information—A Trustability Protocol for Validating Distributed Information. *W3C Workshop on the Future of Social Networking*, 2008. Disponible para descarga en http://www.w3.org/2008/09/msnws/papers/ecano-gburel-xti.pdf.

[28] T. Lee. Atribution Principles for Data Integration: Policy Perspectives, Febrero 2002.

[29] A. Taivalsaari. Classes vs. prototypes: Some philosophical and historical observations. *Prototype-Based Programming: Concepts, Languages and Applications*, pages 3–16, 1996.

[30] S. Muñoz, J. Pérez, and C. Gutiérrez. Simple and Efficient Minimal RDFS. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):220–234, 2009.

[31] E. Prud'hommeaux and A. Seaborne. SPARQL query language for RDF. *World Wide Web Consortium (W3C) Recommendation*, 2008. Publicado en http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/.

[32] M. Salvadores, G. Correndo, S. Harris, N. Gibbins, and N. Shadbolt. The design and implementation of minimal rdfs backward reasoning in 4store. *The Semanic Web: Research and Applications*, pages 139–153, 2011.

[33] J. Broekstra and A. Kampman. Serql: A second generation rdf query language. In *Proc. SWAD-Europe Workshop on Semantic Web Storage and Retrieval*, pages 13–14, 2003.

[34] F. Bry, T. Furche, C. Ley, B. Linse, and B. Marnette. Rdflog: It's like datalog for rdf. In *Proc. Workshop on (Constraint) Logic Programming (WLP)*, 2008.

[35] Axel Polleres. Using datalog for rule-based reasoning over web data: Challenges and next steps, 2010. Presentación realizada en el Datalog 2.0 Workshop y disponible en http://datalog2.com/slides/polleres.pdf.

[36] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked Data-The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[37] T. Berners-Lee. Design issues: Linked data. 2006. Disponible en http://www.w3.org/DesignIssues/LinkedData.html.

[38] R. Stallman et al. The free software definition. *Free Software Free Society: Selected Essays of Richard M. Stallman*, pages 41–44, 1996. Accesible en su versión actualizada por la Fundación del Software Libre en http://www.gnu.org/philosophy/free-sw.html.

[39] L. Lessig. *Free culture: The nature and future of creativity*. Penguin Group USA, 2005.

[40] Carl Malamud, Tim O'Reilly, Greg Elin, Micah Sifry, Adrian Holovaty, Daniel X. O'Neil, Michal Migurski, Shawn Allen, Josh Tauberer, Lawrence Lessig, Dan Newman,

John Geraci, Edwin Bender, Tom Steinberg, David Moore, Donny Shaw, JL Needham, Joel Hardi, Ethan Zuckerman, Greg Palmer, Jamie Taylor, Bradley Horowitz, Zack Exleyand Karl Fogel, Michael Dale, Joseph Lorenzo Hall, Marcia Hofmann, David Orban, Will Fitzpatrick, and Aaron Swartz. Open Government Data Principles, diciembre 2007. Accesible en https://public.resource.org/8_principles.html.

[41] Bibi van den Berg, Ronald Leenes, Stefanie Pötzsch, Martin Pekárek, Arnold Roosendaal, Aleksandra Kuczerawy, Katrin Borcea-Pfitzmann, Filipe Beato, and Sandra Olislaegers. Privacy Enabled Communities. Abril 2010. Disponible en http://www.primelife.eu/results/documents/95-121d.

[42] Alejandro Hevia. Entendiendo la privacidad hoy. *Revista Bits de Ciencia*, (6):58–65, 2011. Disponible en http://www.dcc.uchile.cl/bits-de-ciencia.

[43] S.D. Warren and L.D. Brandeis. The right to privacy. *Harvard law review*, 4(5):193–220, 1890.

[44] C. Paine, U.D. Reips, S. Stieger, A. Joinson, and T. Buchanan. Internet users perceptions of 'privacy concerns' and 'privacy actions'. *International Journal of Human-Computer Studies*, 65(6):526–536, 2007.

[45] Mireille Hildebrandt. Technology and the end of law. *Facing the limits of the law*, 2009.

[46] A. F. Westin. *Privacy and Freedom*. Atheneum, 1967.

[47] H. Nissenbaum. Protecting privacy in an information age: The problem of privacy in public. *Law and Philosophy*, 17(5):559–596, 1998.

[48] H. Nissenbaum. Privacy as contextual integrity. *Wash. L. Rev.*, 79:119, 2004.

[49] P. Golle. Revisiting the uniqueness of simple demographics in the us population. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 77–80. ACM, 2006.

[50] L. Sweeney. Simple demographics often identify people uniquely. *Health (San Francisco)*, pages 1–34, 2000.

[51] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In

Security and Privacy, 2008. SP 2008. IEEE Symposium on, pages 111–125. Ieee, 2008.

[52] R.J. Bayardo and R. Agrawal. Data privacy through optimal k-anonymization. In *Data Engineering, 2005. ICDE 2005. Proceedings. 21st International Conference on*, pages 217–228. IEEE, 2005.

[53] K. LeFevre, D.J. DeWitt, and R. Ramakrishnan. Mondrian multidimensional k-anonymity. In *Data Engineering, 2006. ICDE'06. Proceedings of the 22nd International Conference on*, pages 25–25. Ieee, 2006.

[54] J.W. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-anonymization using clustering techniques. In *Proceedings of the 12th international conference on Database systems for advanced applications*, pages 188–200. Springer-Verlag, 2007.

[55] K. Wang, B.C.M. Fung, and P.S. Yu. Handicapping attacker's confidence: an alternative to k-anonymization. *Knowledge and Information Systems*, 11(3):345–368, 2007.

[56] D. Boyd and A. Markwick. Social Privacy in Networked Publics: Teen's Attitudes, Practices and Strategies. 2011. Paper citado por A. Hevia en [42].

[57] D. Boyd. Privacy and Publicity in the Context of Big Data. Abril 2010. Charla en WWW2010, disponible en http://www.danah.org/papers/talks/2010/WWW2010.html.

[58] G.M. Purushothama and MK Bhandi. Metadata harvesting and the open archives initiative. 2006.

[59] R.W. Moore, A. Rajasekar, and M. Wan. Data grids, digital libraries, and persistent archives: an integrated approach to sharing, publishing, and archiving data. *Proceedings of the IEEE*, 93(3):578–588, 2005.

[60] D.A. Koutsomitropoulos, G.D. Solomou, and T.S. Papatheodorou. Semantic interoperability of dublin core metadata in digital repositories. In *Innovations in Information Technology, 2008. IIT 2008. International Conference on*, pages 233–237. IEEE, 2008.

[61] T. Hey and A. Trefethen. The data deluge: An e-science perspective. In *Grid computing*, pages 809–824. Wiley Online Library, 2003.

[62] P. Eglitis and D. Suchar. Historical lessons, inter-disciplinary comparison, and their application to the future evolution of the eso archive facility and archive services. *Ensuring Long-Term Preservation and Adding Value to Scientific and Technical Data*, 2009.

[63] Real Academia Española. *Diccionario de la Lengua Española, edición 22*. 2001.

[64] Organización Internacional de Normalización. *Norma ISO 15489-1*. Septiembre.

[65] I. Tuomi. Data is more than knowledge: Implications of the reversed knowledge hierarchy for knowledge management and organizational memory. In *System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, pages 12–pp. IEEE, 1999.

[66] C.D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*. Cambridge University Press Cambridge, 2008.

[67] Fadi Maali, John Erickson, and Phil Archer. Data Catalog Vocabulary (DCAT), Abril 2012. Accesible en http://www.w3.org/TR/vocab-dcat.

[68] A. Chervenak, I. Foster, C. Kesselman, C. Salisbury, and S. Tuecke. The data grid: Towards an architecture for the distributed management and analysis of large scientific datasets. *Journal of network and computer applications*, 23(3):187–200, 2000.

[69] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke. A security architecture for computational grids. In *Proceedings of the 5th ACM conference on Computer and communications security*, pages 83–92. ACM, 1998.

[70] I. Foster, C. Kesselman, C. Lee, B. Lindell, K. Nahrstedt, and A. Roy. A distributed resource management architecture that supports advance reservations and co-allocation. In *Quality of Service, 1999. IWQoS'99. 1999 Seventh International Workshop on*, pages 27–36. IEEE, 1999.

[71] R. Wolski. Dynamically forecasting network performance using the network weather service. *Cluster Computing*, 1(1):119–132, 1998.

[72] B. Tierney, W. Johnston, B. Crowley, G. Hoo, C. Brooks, and D. Gunter. The netlog-

ger methodology for high performance distributed systems performance analysis. In *High Performance Distributed Computing, 1998. Proceedings. The Seventh International Symposium on*, pages 260–267. IEEE, 1998.

[73] D.A. Reed, PC Roth, R.A. Aydt, K.A. Shields, LF Tavera, RJ Noe, and BW Schwartz. Scalable performance analysis: The pablo performance analysis environment. In *Scalable Parallel Libraries Conference, 1993., Proceedings of the*, pages 104–113. IEEE, 1993.

[74] B.P. Miller, M.D. Callaghan, J.M. Cargille, J.K. Hollingsworth, R.B. Irvin, K.L. Karavanic, K. Kunchithapadam, and T. Newhall. The paradyn parallel performance measurement tool. *Computer*, 28(11):37–46, 1995.

[75] A. Sim, H. Nordberg, LM Bernardo, A. Shoshani, and D. Rotem. Storage access coordination using corba. In *Distributed Objects and Applications, 1999. Proceedings of the International Symposium on*, pages 168–175. IEEE, 1999.

[76] Network common data form (netcdf). Sitio web: http://www.unidata.ucar.edu/software/netcdf.

[77] The hdf group. Sitio web: http://www.hdfgroup.org.

[78] R. Ferreira, T. Kurc, M. Beynon, C. Chang, A. Sussman, and J.H. Saltz. Object-relational queries into multidimensional databases with the active data repository. *Parallel Processing Letters*, 9(2):173–195, 1999.

[79] E.F. Codd. Derivability, redundancy, and consistency of relations stored in large data banks. *IBM Research Report RJ*, 599, 1969.

[80] D. Florescu and D. Kossmann. Storing and querying xml data using an rdmbs. *IEEE Data Engineering Bulletin*, 22(3):27–34, 1999.

[81] S. Jayavel, T. Kristin, H. Gang, Z. Chun, D.W. David, N. Jeffrey, et al. Relational databases for querying xml documents: Limitations and opportunities. 1999.

[82] M. Arenas and L. Libkin. A normal form for xml documents. *ACM Transactions on Database Systems (TODS)*, 29(1):195–232, 2004.

[83] V. Fionda, C. Gutierrez, and G. Pirró. Semantic navigation on the web of data: Specification of routes, web fragments and actions. *Arxiv preprint arXiv:1111.4316*, 2011.

[84] Deutsche Forshungemeinshaft. Proposals for Safeguarding Good Scientific Practice. Technical report, Enero 1998.

[85] Christian Sifaqui, Eridan Otto, Felipe Almazán, and Daniel Hernández. El camino hacia la web semántica: experiencias de la biblioteca del congreso nacional de chile. *Bits de Ciencia*, (6), 2011.

[86] Harry Burt. Why "Luaïs on everybody's lips, and when to expect MediaWiki 1.19. Technical report, Enero 2012. Publicado por la fundación WikiMedia en http://en.wikipedia.org/wiki/Wikipedia:Wikipedia_Signpost/2012-01-30/Technology_report.

[87] Francisco Cinfuentes Silva. Ontología de biografías parlamentarias para la biblioteca del congreso nacional de chile, 2012. Disponible en http://datos.bcn.cl/portal/ontologias/modelo-de-biografias.

[88] Comunidad de MediaWiki. Help: Templates. http://www.mediawiki.org/wiki/Help:Templates.

[89] Harry Burt. Why "Luaïs on everybody's lips, and when to expect MediaWiki 1.19. http://en.wikipedia.org/wiki/Wikipedia:Wikipedia_Signpost/2012-01-30/Technology_report.

[90] Viston Cerf. 40 things you need to know about the next 40 years. *Smithsonian*, julio 2010.

[91] John Sheridan, febrero 2010.

[92] Jill Mesirov. Accessible reproducible research. *Science*, enero 2010. Publicado en: http://www.sciencemag.org/content/327/5964/415.summary e identifica por DOI:10.1126/science.1179653.

[93] Jeff Jarvis. The importance of provenance. publicado en su blog: BuzzMachine, junio 2010.

[94] Luc Moreau. The Foundations of Provenance on the Web. In *Foundations and Trends in Web Science*, volume 2, noviembre 2009.

[95] DCMI Metadata Terms, octubre 2010. Publicado en: http://dublincore.org/documents/dcmi-terms/.

[96] The Dublin Core Metadata Element Set, Versión 1.1, octubre 2010. Publicado en: http://dublincore.org/documents/dces/.

[97] Dan Brickley and Libby Miller. FOAF Vocabulary Specification 0.98, Agosto 2010. Disponible en http://xmlns.com/foaf/spec.

[98] Uldis Bojars, John G. Breslin, Diego Berrueta, Dan Brickley, Stefan Decker, Sergio Fernández, Christoph Görn, Andreas Harth, Tom Heath, Kingsley Idehen, Kjetil Kjernsmo, Alistair Miles, Alexandre Passant, Axel Polleres, and Luis Polo. SIOC Core Ontology Specification, marzo 2010. Publicado en: http://rdfs.org/sioc/spec/.

[99] J. J. Carroll, C. Bizer, P. Hayes, and P. Stickler. Named Graphs, Provenance and Trust. In *14th International World Wide Web Conference (WWW)*. ACM Press, mayo 2005.

[100] Web Of Trust RDF Ontology (WOT). Disponible en http://xmlns.com/wot/0.1/.

[101] Deborah L. McGuinness, Paulo Pinheiro da Silva, and Li Ding. Proof Markup Language (PML) Primer, Octubre 2007. Publicado en http://inference-web.org/2007/primer/.

[102] P.P. Da Silva, D.L. McGuinness, and R. Fikes. A proof markup language for semantic web services. *Information Systems*, 31(4-5):381–395, 2006.

[103] J. Zhao. *A conceptual model for e-science provenance*. PhD thesis, the University of Manchester, 2007.

[104] Sam Tunnicliffe and Ian Davis. Changset, Mayo 2009.

[105] Jun Zhao. Open Provenance Model Vocabulary Specification, octubre 2010. Publicado en:

http://purl.org/net/opmv/ns-20101006.

[106] Luc Moreau. Open Provenance Model, octubre 2010. Publicado en: http://openprovenance.org/model/opmo.

[107] Olaf Harting. Provenance Information in the Web of Data. In *Linked Data on the Web (LDOW)*, abril 2009.

[108] Luc Moreau, Ben Clifford, Juliana Freire, Joe Futrelle, Yolanda Gil, Paul Groth, Natalia Kwasnikowska, Simon Miles, Paolo Missier, Jim Myers, Beth Plale, Yogesh Simmhan, Eric Stephan, and Jan Van den Bussche. The open provenance model core specification (v1.1). *Future Generation Computer Systems*, 2010. Publicado en: http://eprints.ecs.soton.ac.uk/21449/.

[109] Luc Moreau, Paul Groth, Ben Clifford, and Simon Miles. Open Provenance Model (OPM) XML Schema Specification, octubre 2010. Publicado en: http://openprovenance.org/model/opmx-20101012.

[110] Luc Moreau, Li Ding, Joe Futrelle, Daniel Garijo Verdejo, Paul Groth, Mike Jewell, Simon Miles, Paolo Missier, Jeff Pan, and Jun Zhao. Open Provenance Model (OPM) OWL Specification, octubre 2010. Publicado en: http://openprovenance.org/model/opmo-20101012.

[111] Olaf Hartig and Jun Zhao. Publishing and Consuming Provenance Metadata on the Web of Linked Data. In *International Provenance and Anotation Workshop*, 2010. Publicado en: http://olafhartig.de/files/HartigZhao_Provenance_IPAW2010_Preprint.pdf.

[112] Alexander, K., Cyganiak, R., Hausenblas, M., and Zhao, J. Describing linked datasets, 2009.

[113] Olaf Hartig. Querying Trust in RDF Data with tSPARQL. In *The Semantic Web: Research and Applications*. SpringerLink, 2009. Publicado en: http://www.springerlink.com/content/b7x755151711ku65/.

[114] Open Knowledge Foundation. Open Knowledge Definition. Publicado en: http://www.opendefinition.org/okd/.

[115] Becky Hogge. Open Data Study, mayo 2010. Transparency and Accountability Initiative. Publicado en: http://www.soros.org/initiatives/information/focus/communication/articles_publications/publications/open-data-study-20100519/open-data-study-100519.pdf.

[116] D. Booth. Uris and the myth of resource identity. In *Proceedings of Identity, Reference, and the Web Workshop at the WWW Conference*, 2006.

[117] Sauermann, L., Cyganiak, R., Danny Ayers, and Völkel, M. Cool uris for the semantic web. Diciembre 2011.

[118] S. Auer, J. Lehmann, and A.C. Ngonga Ngomo. Introduction to linked data and its lifecycle on the web. *Reasoning Web. Semantic Technologies for the Web of Data*, pages 1–75, 2011.

[119] Creating URIs.

[120] Paul Davidson. Designing uri sets for the uk public sector.

[121] Biblioteca del Congreso Nacional de Chile. Convenciones. publicadas en http://datos.bcn.cl/portal/es/documentacion/convenciones/.

[122] L. Dodds and I. Davis. Linked data patterns. *A pattern catalogue for modelling, publishing, and consuming Linked Data*, 2011.

[123] Stuart Williams. Mapping between URIs and Internet Media Types, Abril 2002. Discusión publicada en http://www.w3.org/2001/tag/2002/01-uriMediaType-9.

[124] Jeni Tennison. Translating Existing Models to RDF, marzo 2010. Publicado en http://www.jenitennison.com/blog/node/142.

[125] P.L. Bergstein. Object-preserving class transformations. In *ACM SIGPLAN Notices*, volume 26, pages 299–313. ACM, 1991.

[126] E. Casais. An incremental class reorganization approach. In *ECOOP'92 European Conference on Object-Oriented Programming*, pages 114–132. Springer, 1992.

[127] W.F. Opdyke and R.E. Johnson. Creating abstract superclasses by refactoring. In *Proceedings of the 1993 ACM conference on Computer science*, pages 66–73. ACM, 1993.

[128] W.F. Opdyke. *Refactoring object-oriented frameworks*. PhD thesis, University of Illinois, 1992.

[129] M. Arenas, J. Pérez, J.L. Reutter, and C. Riveros. Foundations of schema mapping management. In *Proceedings of the twenty-ninth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems of data*, pages 227–238. ACM, 2010.

[130] J Clark. Xsl transformations (xslt) version 1.0. *W3C recommendation*, noviembre 1999.

[131] S. Kawanaka and H. Hosoya. bixid: a bidirectional transformation language for xml. *ACM SIGPLAN Notices*, 41(9):201–214, 2006.

[132] RV Guha and B. McBride. Rdf vocabulary description language 1.0: Rdf schema. *W3 Technical Reports*, 2004.

[133] S. Bechhofer, F. Van Harmelen, J. Hendler, I. Horrocks, D.L. McGuinness, P.F. Patel-Schneider, and L.A. Stein. Owl web ontology language reference. *W3C recommendation*, 2004.

[134] Michael K. Smith, Chris Welty, and Deborah L. McGuinness. Owl web ontology language overview. *W3C recommendation*, 2004.

[135] A. Isaac and E. Summers. Skos simple knowledge organization system primer. *W3 Technical Reports*, 2009.

[136] Dave Reynolds. An organization ontology, 2010. Publicada en http://www.epimorphics.com/public/vocabulary/org.html.

[137] Renzo Angles, Alex Bórquez, Claudio Gutiérrez, Andrés Neyem, Sergio F. Ochoa, Andrés Pereira, and Edgard Pineda. Guía de desarrollo y uso de esquemas de gobierno, Junio 2009. Ministerio de Economía Fomento y Reconstrucción.

[138] Francisco Cifuentes Silva. Ontología de Biografías Parlamentarias para la Biblioteca del Congreso Nacional de Chile, marzo 2012.

[139] Alvaro Graves. PoderVocabulary: A Vocabulary for Poderopedia. Su repositorio se encuentra en el repositorio https://github.com/poderopedia/PoderVocabulary y se ha revisado el commit d18d5e147a.

[140] D.C. Halbert and P.D. O'Brien. Using types and inheritance in object-oriented programming. *IEEE Software*, 4(5):71–72, 1987.

[141] Francisco Cifuentes Silva. Ontología de Transparencia para la Biblioteca del Congreso Nacional de Chile, octubre 2011.

[142] David Menestrina, Omar Benjelloun, and Hector Garcia-Molina. Generic entity resolution with data confidences. Technical Report 2005-35, Stanford InfoLab, 2005.

[143] H. Halpin, P. Hayes, J. McCusker, D. Mcguinness, and H. Thompson. When owl:sameAs isn't the same: An analysis of identity in linked data. *The Semantic Web–ISWC 2010*, pages 305–320, 2010.

[144] Pat Hayes and Chris Welty. Defining N-ary Relations on the Semantic Web. Technical report, Abril 2006. Publicado por el W3C en http://www.w3.org/TR/swbp-n-aryRelations/.

[145] M. San Martın, C. Gutierrez, and P.T. Wood. Snql: A social networks query and transformation language. *5th. Alberto Mendelzon Workshop on Foundations of Databases*, 2011.

[146] E. Truyen, W. Joosen, B.N. Jørgensen, and P. Verbaeten. A generalization and solution to the common ancestor dilemma problem in delegation-based object systems. In *Dynamic Aspects Workshop (DAW04)*, page 6, 2004.

[147] Instrucción General Nº 9 Del Consejo para la Transparencia que modifica las instrucciones generales Nº 4 y Nº 7 sobre Transparencia Activa. *Diario Oficial de Chile*, agosto 2010.

[148] XRI Technical Comite. XRI solves real problems. Disponible en https://wiki.oasis-open.org/xri/XriSolvesRealProblems.

[149] XRI Technical Comite. XRI TC – W3C TAG. Disponible en https://wiki.

oasis-open.org/xri/XriTcW3cTag.

[150] Vicent Quint. XRI 2.0 Review by the W3C TAG. Disponible en http://lists.w3.org/Archives/Public/www-tag/2005Apr/0095.html.

Apéndices

Apéndice A

Términos del vocabulario propuesto

El vocabulario propuesto reusa vocabularios existentes y define nuevos términos. Este anexo describe los términos nuevos del vocabulario propuesto, que se codifican con el prefijo vacío (:) para diferenciarlos de términos de vocabularios ya existentes.

:surnameOfFather	:hasGrade	:price
:surnameOfMother	:Grade	:hasComponent
:rut	:hasStratum	:concept
:electedFromTerritory	:Stratum	:RemunerationScale
:affiliationWhenElected	:qualification	:RemunerationScaleComponent
:headOf	:worksInTerritory	:remunerationScale
:belongsTo	:Remuneration	:organization
:contractualMode	:RemunerationComponent	:validityInterval
:ContractualMode	:remunerationPeriod	:hasScaleComponent

Para cada uno de estos términos se definirán las componentes básicas (dominio, rango, super predicado y super clase), una descripción y términos existentes en otros vocabularios con roles similares. Para cada uno de los elementos definidos se usará un encabezardo de la forma: $t\ e \subset a$, donde t puede ser "Predicado" o "Clase" y la relación $\subset$ indica que el elemento definido e es más preciso que el elemento ya existente a. Además, en el caso de los predicados se introducirá la notación $d \rightarrow r$ indicando que el dominio (d) y el recorrido (r) del predicado definido. En particular, el simbolo $*$ en el dominio o en el recorrido indicará que ese valor no se encuentra acotado.

Predicado :surnameOfFather $\subset$ foaf:familyName
(foaf:Person $\rightarrow$ xsd:string)

El predicado :surnameOfFather indica el apellido del padre, que en Chile también es el primer apellido.

Elementos similares en otros vocabularios

- bcnb:surnameOfFather
- pod:firstLastName

Predicado :surnameOfMother $\subset$ foaf:familyName
(foaf:Person $\rightarrow$ xsd:string)

El predicado :surnameOfMother indica el apellido de la madre, que en Chile también es el segundo apellido.

Elementos similares en otros vocabularios

- bcnb:surnameOfMother
- pod:secondLastName

Predicado :rut $\subset$ dc:identifier
foaf:Agent $\rightarrow$ xsd:string

El predicado :rut indica el Rol Único Tributario que identifica a las personas naturales y jurídicas.

Elementos similares en otros vocabularios

- pod:hasTaxId
- org:identifier (sólo en el caso de personas jurídicas)

Predicado :electedFromTerritory

foaf:Membership $\rightarrow *$

El predicado :electedFromTerritory indica el territorio de representación de un cargo público como, por ejemplo, el de distrito en el caso de diputados, la circunscripción en el caso de senadores o la comuna en los alcaldes. La semántica de este predicado guarda relación con el territorio al cual pertenecen los electores en un sistema político con representaciones territoriales.

Elementos similares en otros vocabularios

- bcnb:hasRepresentationIn Este predicado tiene una semántica relacionada, pero no similar, pues sirve para indicar que un determinado partido político posee representación sobre una región. En cambio, el predicado :electedFromTerritory tiene como dominio a instancias de org:Membership con el fin de relacionar a la persona que ocupa el cargo con el territorio.
- bcnb:hasParliamentaryRepresentationIn Este predicado también está relacionado con la representación de los partidos políticos, pero se acota al ámbito parlamentario y su dominio son los partidos.

Predicado :affiliationWhenElected

foaf:Membership $\rightarrow *$

Indica el partido al que representó la persona electa cuando era candidato. Es necesario indicar que este partido podría cambiar a lo largo del período de ejercicio del cargo del candidato, por lo que este dato se restringe sólo al momento de la elección.

Elementos similares en otros vocabularios

- bcnb:hasPoliticalParty

Predicado :headOf

org:Role → org:Organization

Indica que un rol tiene la función de dirigir una organización. Puede ser usado para indicar el rol que dirige una organización o una unidad organizacional, como puede ser una municipalidad o el jefe de un departamento.

Elementos similares en otros vocabularios

- org:headOf Este predicado del vocabulario org es diferente, pues su dominio es foaf:Agent, lo que no permite caracterizar el período en el que una persona se mantiene como líder de una organización.

Predicado :belongsTo

org:Role → org:Organization

Indica la organización en la que un rol fue definido.

Elementos similares en otros vocabularios

- bcnt:isPartOfOrganicUnit (Aunque su dominio es bcnt:Position)

Predicado :contractualMode

org:Membership → :ContractualMode

El predicado :contractualMode indica la modalidad contractual de una contratación.

Elementos similares en otros vocabularios

- bcnt:contractualMode

Clase :ContractualMode $\subset$ skos:Concept

La clase :ContractualMode permite definir tipos de modalidades contractuales.

Predicado :hasGrade
org:Membership $\rightarrow$:Grade

Indica el grado de una contratación.

Elementos similares en otros vocabularios

- bcnt:hasGrade

Clase :Grade $\subset$ skos:Concept

Describe grados en las contrataciones.

Predicado :hasStratum
org:Membership $\rightarrow$:Stratum

Indica el estamento de una contratación.

Elementos similares en otros vocabularios

- bcnt:hasStratum

Clase :Stratum $\subset$ skos:Concept

Describe estamentos en las contrataciones.

Predicado :qualification
org:Membership $\rightarrow$ skos:Concept

Indica la calificación profesional o formación de una persona al momento de asumir un cargo.

Elementos similares en otros vocabularios

- bcnb:profession

Predicado :worksInTerritory
org:Membership $\rightarrow *$

Indica el territorio en el que una persona ejerce un cargo.

Elementos similares en otros vocabularios

- bcnt:hasRegion

Class :Remuneration

Permite agrupar la información de las remuneraciones que recibe una persona.

Class :RemunerationComponent

Corresponde al desglose de una remuneración.

Predicado :remunerationPeriod

org:Membership → time:Interval

Indica el intervalo de tiempo al que está asociado una remuneración.

Elementos similares en otros vocabularios

- bcnt:hasPositionPeriod

Predicado :price

* → gr:UnitPriceSpecification

Asocia cualquier elemento con un monto monetario. Puede ser usado para indicar montos en remuneraciones.

Predicado :hasComponent

:Remuneration → :RemunerationComponent

Permite desglosar remuneraciones en componentes.

Predicado :concept

:RemunerationComponent → skos:Concept

Permite indicar un concepto por el cual se desglosa una remuneración.

Class :RemunerationScale

Corresponde a una escala de remuneraciones.

Class :RemunerationScaleComponent

Corresponde al desglose de una remuneración.

Predicado :remunerationScale
org:Membership → :RemunerationScale

Relaciona una contratación con la escala de remuneraciones de la cual se pueden deducir las remuneraciones de la persona contratada.

Predicado :organization
:RemunerationScale → org:Organization

Indica la organización en la que se define una escala de remuneraciones.

Predicado :validatyInterval
:RemunerationScale → time:Interval

Indica el intervalo de tiempo sobre el cual una escala de remuneraciones se encuentra vigente.

Predicado :hasScaleComponent
:RemunerationScale → :ScaleComponent

Permite desglozar escalas de remuneraciones en componentes.