

Introducción a la Web Semántica: *Resource Description Framework*

Alfonso Egio*

25 de Noviembre de 2005

Copyright © 2005 Alfonso Egio. Se otorga permiso para copiar, distribuir, y/o modificar este documento bajo los términos de la GNU Free Documentation License, Version 1.1 ó cualquier versión posterior publicada por la Free Software Foundation; sin secciones invariantes, ni textos de portada, o contraportada. Una copia de ésta licencia es incluida la sección titulada "GNU General Public License"

Índice

1. Introducción	3
1.1. Definición de RDF	3
1.2. Un ejemplo introductorio	3
2. Arquitectura básica de RDF	5
2.1. Amigos e identificación de recursos mediante URIs	5
2.2. URIs como identificadores de predicados	8
3. Representaciones de RDF	12
3.1. Grafos RDF	12
3.2. RDF/XML	12
4. Aplicaciones, ontologías y/o esquemas RDF	14
4.1. RDF	14
4.2. RDFS - RDF Schema	15
4.3. DC - Dublin Core Metadata Initiative	15
4.4. OWL	16
4.5. FOAF - Friend Of A Friend	17

*alfonso.egio@gmail.com

5. Software RDF	18
5.1. RDF y Perl → RDF::Simple	18
5.1.1. Código fuente del ejemplo de uso de RDF::Simple::Parser	18
5.1.2. Comentarios acerca del ejemplo	20
6. The GNU General Public License	23

1. Introducción

1.1. Definición de RDF

Resource Description Framework es un lenguaje capaz de representar información acerca de recursos en la Red. En principio fue concebido como una herramienta que permitiera representar datos acerca de las páginas web de internet (*metadatos*) de modo que sirviera para identificar la fecha de creación, autor, etc ... de una página cualquiera. No obstante, es generalizable para describir cualquier tipo de recurso **identificable** mediante la red (*y no necesariamente recuperable a través de la red*).

1.2. Un ejemplo introductorio

Hasta aquí, RDF aparenta no aportar muchas novedades al esquema actual basado en HTTP y HTML para la WWW. A la hora de confeccionar una página web, podemos hacer constar todo ese tipo de información de un modo apropiado para que otro usuario la identifique correctamente. A modo de ejemplo podemos estudiar el siguiente html correspondiente al Web Site corporativo de una empresa fabricante de muebles:

```
<html>
<head>
  <title>Mueblosa</title>
</head>
<body>
  <h1>Pagina de Mueblosa</h1>
  <ul>
    <li>Director de la empresa: Sotanito Mueblez (sm@mueblosa.es)</li>
    <li>Autor: Menganito Perez (mp@mueblosa.es)</li>
    <li>Fecha de creacion: 11/1/05</li>
  </ul>
          *   *   *   *   *
</body>
</html>
```

Cualquier persona que recupere el documento de la red a través de un navegador como Mozilla o Internet Explorer podrá ver algo similar al contenido expuesto en la figura 1

No obstante, podemos notar que el ejemplo anterior adolece de ciertos problemas. Por ejemplo, supongamos, que a un usuario que desconoce el castellano le ha gustado la maquetación de la página y quiere ponerse en contacto con el autor para ofrecerle trabajo como web master. Desgraciadamente y puesto que la información acerca de la identidad del creador de la Web y el director de la compañía ha sido expresada de un modo estructurado sólo en cuanto a la forma de visualizarlo (*lista html*) carece de elementos de juicio para averiguar que dirección utilizar pudiendo enviar la oferta de trabajo a la persona equivocada.

Página de Mueblosa

- Director de la empresa: Sotanito Muéblez (sm@mueblosa.es)
- Autor de la pagina: Menganito Pérez (mp@mueblosa.es)
- Fecha de creación: 11/1/05

Figura 1: *Ejemplo HTML sencillo visto en un navegador*

RDF pretende ser un soporte para la expresión de relaciones entre recursos de cualquier tipo con carácter universal y distribuido de modo que facilite la identificación de la información sin dar lugar a ambigüedades principalmente encaminado a ser procesado por aplicaciones en lugar de clientes humanos. De acuerdo con lo anterior, podríamos tratar de estructurar los contenidos del ejemplo en un modelo formado por un conjunto de *sentencias* compuestas por *sujetos*, *objetos* y *predicados* del tipo:

Sujeto	Predicado	Objeto
Mueblosa	es una	empresa
Mueblosa	es una	Sociedad Anónima
Mueblosa	está dirigida por	Sotanito Muéblez
La Página de Mueblosa	ha sido creada por	Menganito Pérez
sm@mueblosa.es	es la dirección de mail de	Sotanito Muéblez
mp@mueblosa.es	es la dirección de mail de	Menganito Pérez
La página de Mueblosa	fue creada el día	11/1/05

Dicha tabulación se corresponde con todo el conocimiento acerca de Mueblosa que un usuario de la web de habla hispana es capaz de adquirir mediante su capacidad de comprender el mensaje y ayudado por su sentido común a partir de la representación html realizada por un navegador. La mayor diferencia con respecto del esquema html es que en la representación de hipertexto nos preocupamos por la forma (*cabeceras, listas, etc ...*) y sin embargo, cuando intentamos crear el modelo de sentencias tabuladas, nuestro principal interés es estructurar la información en cuanto a las relaciones existentes entre los datos que presentamos.

No obstante, de modo previo a la representación de las relaciones existentes entre las distintas piezas de la información; tenemos que considerar el problema en toda su generalidad. Es decir, se requiere crear un esquema de identificación de recursos que en última instancia estará distribuido y además será de uso universal (*al igual que la WWW*) y por lo tanto para dotarlo de robustez necesitamos utilizar un convenio de identificación de recursos apropiado. Precisamente uno de los pilares que sustentan RDF es la idea de que **CUALQUIER COSA** pueda ser identificada mediante identificadores web o URIs (*Uniform Resource Identifiers*).

2. Arquitectura básica de RDF

2.1. Amigos e identificación de recursos mediante URIs

La WWW asocia cada recurso (*ya sea documento, servicio, etc ...*) recuperable a través de un protocolo de internet con un URL (*Uniform Resource Locator*) consistente en una cadena de caracteres que representa el mecanismo primario de acceso al documento y/o recurso. Ejemplos de ello pueden ser:

- http://www.mueblosa.es/catalogo_04/index.html
- <mailto:sm@mueblosa.es>
- <ftp://ftp.mueblosa.es>

Uno de los requisitos fundamentales que otorgan sentido a un URL es el hecho de que el recurso cuya ruta de acceso identifica (*página web, acceso ftp, mailbox, ...*) debe ser recuperable a través de la red o en palabras llanas, debe existir algo detrás como por ejemplo un documento HTML recuperable a través del protocolo HTTP o un servidor de correo dispuesto a entregar un mensaje a la persona a la que corresponda la dirección del destinatario.

Las direcciones basadas en URLs son un subconjunto de los conocidos como identificadores URI. Reparando en la nomenclatura empleada en los convenios podemos empezar a entender en que sentido los URIs son más generales que los URLs:

Resource Locator → Resource Identifier

Es decir, un URL es una **dirección** mediante la cual accedemos a un recurso a través de internet; en contraposición un URI es un **identificador** al que un grupo de personas conviene en otorgarle un significado concreto con objeto de convertirlo en sujeto, objeto o predicado de cualquier sentencia RDF articulando de éste modo formas de comunicación entorno a él.

Para centrar las ideas, podríamos considerar un ejemplo de aplicación de la web semántica que funciona desde aproximadamente 1998 cuando Dan Brickley publica una página HTML en su espacio web del *Institute for learning and research technology* de la Universidad de Bristol en la que hace constar una estructura RDF en la que detalla su dirección de correo electrónico y la de su página personal además de su nombre y apellidos dando lugar a lo que hoy en día se conoce como proyecto FOAF* (*Friend Of A Friend*).

El proyecto FOAF no es más que la especificación de un vocabulario RDF que permite codificar conocimiento concreto acerca de personas; *de hecho una de las clases básicas definidas dentro del vocabulario FOAF es **Person*** y bajo las instancias de dicha clase asociamos piezas de información tales como la dirección de correo electrónico de la persona que se pretende describir, enlaces con otras personas a través del predicado definido bajo la etiqueta **knows** o su género mediante el predicado **gender**.

*©Dan Brickley & Libby Miller (2000 - 2005)

Uno de los primeros problemas a resolver a la hora de levantar FOAF fue el de asignar URIs a las personas de modo que pudieran ser identificadas de un modo unívoco. Utilizar directamente su nombre y apellidos para componer un URI no parece muy buena opción cuando se aspira a crear una aplicación de ámbito mundial (*existe una probabilidad en ningún caso negligible de que dos personas se llamen igual*). Después de analizar detalladamente el problema, se llegó a la conclusión de que el URL asociado a la dirección de correo electrónico era más apropiado pues los servicios de correo electrónico **personales**, salvo en raras excepciones, suelen ser de carácter personal e intransferible. No obstante se plantea un problema pues si el convenio aceptado es que la dirección de correo electrónico pase a ser un URI de la persona, desproveemos de mecanismos para referenciar la dirección en sí misma como entidad independiente.

Es decir, supongamos el caso en que un Virus informático que se transmite por correo electrónico se propague dentro de una Intranet corporativa; supongamos además que existe un registro RDF de las incidencias ocurridas en el que consta un listado de las direcciones de correo electrónico a las que ha afectado. En dicho registro encontraremos sentencias del estilo de:

Sujeto	Predicado	Objeto
El virus ZX27	ha infectado a	sm@mueblosa.es

en caso de que el URI de la persona sea su dirección de correo electrónico, la sentencia equivale a decir que dicha persona está afectada de un virus cuando en realidad queríamos referir el hecho de que el virus se ha propagado a través de su dirección y por lo tanto el que puede estar afectado en todo caso es su software en lugar de su propio organismo.

FOAF aporta la solución para el problema de modo que en ningún momento se asigna un URI a la persona. La propuesta consiste en establecer juegos de sentencias del tipo:

Sujeto	Predicado	Objeto
X	pertenece a la clase	persona
X	se llama	Sotanito Muéblez
X	tiene una dirección de correo electrónico	sm@mueblosa.es

Donde **X** es lo que se conoce como un nodo vacío dentro de la jerga de RDF y que veremos más adelante con mayor detalle. De momento, nos basta con apreciar la capacidad de evitar hacer mención de un URI para cada persona de manera que el modo de formular consultas o establecer nuevas sentencias del tipo de Sotanito conoce a Menganito pasa por un esquema del tipo: Sotanito (*que es una persona cuya dirección de correo electrónico es sm@mueblosa.es*) conoce a Menganito (*que se trata de otra persona cuya dirección de correo electrónico es mp@mueblosa.es*).

Un ejemplo aún más esclarecedor podría ser el URL utilizado para identificar a una empresa. Si consideramos `http://www.mueblosa.es` como candidato para la identificación de Mueblosa nos encontramos con problemas similares a la hora de establecer sentencias como:

Sujeto	Predicado	Objeto
http://www.mueblosa.es	fecha de creación	1/1/2005

pues podría dar lugar a malentendidos de la misma índole: al identificar la empresa y su página web bajo un mismo URL desconocemos exactamente cual es el significado de sentencias que citen el recurso <http://www.mueblosa.es>. Lo que nos lleva a plantear varias posibilidades como la creación de un directorio RDF de empresas basado en los registros de actividad comercial vigentes o por el contrario recurrir a una estrategia similar a la de FOAF refiriéndose en todo momento a cualquier corporación *indirectamente* a través del URL de su Web Site corporativa.

2.2. URIs como identificadores de predicados

Hasta el momento, todos los ejemplos expuestos implicaban recursos familiares como páginas web o direcciones de correo electrónico. Además en ellos se ha utilizado el lenguaje común para describir los predicados (*p. ej. tiene una dirección de correo electrónico*). No obstante, **RDF sólo contempla la posibilidad de utilizar cadenas de texto literales a modo de objeto de las sentencias**. Es decir, *resulta lógico pensar que puesto que RDF ha sido concebido para describir recursos a través de internet, la forma de identificar dichos recursos (sujetos de las sentencias) debe consistir en un esquema general que permita referirlos de forma inequívoca, universal y distribuida (i.e. URIs)*. Los **objetos** por el contrario *pueden consistir en cadenas de texto en caso de que no sea necesario establecer más referencias al respecto*.

Atención especial merecen los **predicados**, pues el hecho de que al igual que los sujetos requieran un esquema URI de identificación proporciona a RDF su verdadera dimensión semántica. El hecho de identificar los predicados de las sentencias mediante URIs podría implicar la solución al problema descrito por Umberto Eco en su ensayo *La Búsqueda de una Lengua Perfecta* cuando cita la crítica de los enciclopedistas del siglo XVIII a los partidarios de las lenguas filosóficas a priori del siglo XVII:

El hecho es que la cultura del siglo XVIII ha cambiado, respecto a la cultura del siglo anterior, el foco de su atención respecto al lenguaje. Ahora se sostiene que pensamiento y lenguaje se influyen mutuamente y actúan al mismo tiempo, o bien el lenguaje, a medida que se va ampliando, modifica el pensamiento. Si es así ya no se puede sostener la hipótesis racionalista de una gramática del pensamiento, universal y estable, a la que en cierto modo reflejan los lenguajes naturales. No existe ningún sistema de las ideas, postulado sobre la base de una razón abstracta, que pueda llegar a ser parámetro y criterio para la construcción de una lengua perfecta: la lengua no refleja un universo conceptual platónicamente preconcebido, sino que contribuye a formararlo.

RDF proporciona un mecanismo universal y distribuido para la convención de predicados y por lo tanto, al entrar éstos últimos dentro del juego convirtiéndose a su vez en recursos pueden ser creados dinámicamente y enlazados unos con otros a través de terceros predicados de modo que evitan el problema que supondría escoger como medio de representación del conocimiento un esquema basado en un *universo conceptual platónicamente preconcebido*. De hecho, RDF, se presenta como un mecanismo para plasmar sistemas de conocimiento que si bien siguen siendo preconcebidos; están provistos de los mecanismos necesarios para adaptarse al pensamiento con un nivel de expresividad equiparable al del lenguaje natural además de las ventajas derivadas del hecho de que pueden ser procesados de un modo automático y seguro mediante sistemas computacionales. Merece la pena citar otro de los fragmentos que Umberto Eco dedica a los sistemas de Inteligencia Artificial pues describe unos objetivos muy próximos a los que RDF pretende alcanzar:

El sueño de una lengua perfecta con la que se puedan definir todos los significados de los términos de un lenguaje natural, y que permita interacciones dialógicas *sensatas* entre hombre y máquina, o que permita a las máquinas elaborar inferencias propias de las lenguas naturales, aparece de nuevo en las investigaciones contemporáneas sobre Inteligencia Artificial. Por ejemplo, se intenta proporcionar a la máquina reglas de inferencia, a partir de las cuales pueda *juzgar* la coheren-

cia de una historia, o sea capaz de concluir, del hecho de que alguien esté enfermo que necesita consecuentemente ser curado, y así sucesivamente. La literatura que existe a este respecto es bastante extensa, y los sistemas son múltiples; desde los que todavía piensan en una semántica de componentes elementales hasta los que proporcionan a la máquina esquemas de acción, o hasta de situaciones (*frames, scripts, goals*).

No obstante, Umberto Eco, no parece demasiado convencido con dichos sistemas** y finaliza el apartado dedicado a la inteligencia artificial recurriendo de nuevo a la crítica expuesta anteriormente:

Todos los proyectos de Inteligencia Artificial heredan en cierto modo la problemática de las lenguas filosóficas a priori, y solamente consiguen resolver algunos de sus problemas

mediante soluciones ad hoc y para porciones muy locales respecto a todo el ámbito de acción de una lengua natural.

Puede ser interesante tratar de contraponer el punto de vista de Eco en 1993 con una afirmación reciente de R.V. Guha refiriéndose a los esfuerzos en el desarrollo de la Web Semántica:

There is nothing interesting in what we are doing. The only interesting thing is the scale upon which we are attempting to do it

RDF constituye un nuevo intento para tratar de obtener la solución de los problemas expuestos por el semiótico italiano escapando del ámbito de aquellas primeras soluciones ad hoc creadas para describir porciones muy locales respecto al ámbito de acción de una lengua natural. Y por lo tanto, pese a que Guha se refiera a una escala geográfica y/o cuantitativa (*número de usuarios, países implicados ...*), podemos efectuar una nueva lectura de su afirmación si consideramos que RDF además de resultar un sistema de aplicación global igual que la WWW tal y como hoy la conocemos, también y en consecuencia pueda llegar a conquistar **la escala que implica todo el ámbito de acción de una lengua natural**.

Para ilustrar mejor éstas ideas podemos visitar el proyecto FOAF para destacar algunas de sus características en cuanto a establecimiento de predicados concretos acerca de relaciones personales. Para empezar, es necesario mencionar el hecho de que los sistemas RDF como cualquier otro tipo de recurso, tienen un URI asignado, en el caso de FOAF se trata de <http://xmlns.com/foaf/0.1/>,

**La primera edición del ensayo data de 1993 y supongo que se refiere a sistemas de aserción - inferencia similares a PROLOG

el URI de cada uno de los sujetos que describe se obtiene añadiendo una palabra clave al final del URI del recurso general FOAF. De modo que el concepto asociado a la clase Person de FOAF tiene un URI: <http://xmlns.com/foaf/0.1/Person>.

Igualmente, el predicado que en lenguaje natural expresaríamos al decir que una persona conoce a otra tiene una representación dentro del esquema de FOAF en el que el URI asignado es <http://xmlns.com/foaf/0.1/knows>, si recurrimos al comentario asociado en FOAF a dicho predicado, obtenemos el literal:

A person known by this person (indicating some level of reciprocated interaction between the parties).

De modo que en cualquier ocasión en la cual interese expresar una relación entre dos personas como la descrita anteriormente podemos elaborar una construcción del estilo de:

Sujeto	Predicado	Objeto
X	http://www.w3.org/22-rdf-syntax-ns#type	http://xmlns.com/foaf/0.1/Person
X	http://xmlns.com/foaf/0.1/mbox	sm@muebloa.es
Y	http://www.w3.org/22-rdf-syntax-ns#type	http://xmlns.com/foaf/0.1/Person
Y	http://xmlns.com/foaf/0.1/mbox	mp@muebloa.es
X	http://xmlns.com/foaf/0.1/knows	Y

Si analizamos la composición anterior teniendo en cuenta que <http://www.w3.org/22-rdf-syntax-ns#type> se trata del URI asociado al concepto de pertenencia a una clase establecido en el sistema RDF podemos apreciar que en lenguaje natural es equivalente a:

- i) existen dos recursos pertenecientes a la clase persona
- ii) uno de ellos tiene una dirección de correo electrónico sm@muebloa.es
- iii) la dirección de correo electrónico del otro es mp@muebloa.es
- iv) ambas personas se conocen, dónde la descripción del predicado "**se conocen**" se circunscribe dentro de las especificaciones de FOAF y por lo tanto, cualquier aplicación que pretenda la interpretación de dicha sentencia debe acogerse a dicha especificación

No obstante, las cosas se pueden complicar tal y como se aprecia en la descripción del predicado <http://xmlns.com/foaf/0.1/gender> contenida en FOAF:

The gender of this Agent (typically but not necessarily 'male' or 'female').
 (whatever one's gender is, and we are liberal in leaving room for more options than 'male' and 'female', we model this so that an agent has only one gender)

:)

El comentario anterior puede parecer curioso y anecdótico, pero nos puede servir para intuir las profundas implicaciones que puede tener el establecimiento de la web semántica como vía de comunicación universal. Es decir, los modeladores de FOAF, han sido especialmente cuidadosos con el colectivo de personas hermafroditas y por lo tanto dejan espacio en sus especificaciones para poder especificar el género más allá de la disyuntiva *Male Vs. Female*. En caso de que no hubiera sido así, dicho colectivo podría objetar que está al margen de las especificaciones de FOAF y por lo tanto no podrían expresar adecuadamente la naturaleza de su género dentro de FOAF; pese a todo, siempre hemos de tener en mente que la web semántica es de naturaleza distribuida y por lo tanto, dicho colectivo podría crear su propio esquema RDF con un URI asociado al género hermafrodita pudiéndolo hacer constar como objeto de sentencias FOAF o de cualquier otra aplicación RDF en la que fuera necesario.

Ésta última observación nos lleva a cuestionarnos una serie de detalles de orden práctico relacionados con la capacidad real de los individuos y colectivos de establecer esquemas RDF, popularizarlos entre los modeladores e integrarlos con las aplicaciones que deben procesarlos. Es decir, a la hora de comunicarse mediante la web semántica, las grandes organizaciones (*i.e. estructuras gubernamentales, compañías multinacionales, ...*) parten con ventaja en lo que se refiere a recursos tecnológicos, intelectuales y a capacidad de anticipación y difusión de sus esquemas para conseguir sus objetivos. Por lo tanto, seguramente se producirán situaciones que propiciarán actitudes monopolistas con respecto a la información y al conocimiento en las que la libertad de expresión se vea mermada de un modo implícito debido al hecho de que la recepción y la emisión de enunciados esté hábilmente manipulada por parte de grandes poderes capaces de difundir esquemas que ya no permitan la libre difusión del conocimiento sino el control de la forma en que deben expresarse las ideas con objeto de hacerlas sutil y exclusivamente *compatibles* con sus intereses a escala global.

Pese a todo, existen ya ejemplos de situaciones similares en las que se ha establecido un equilibrio. Por ejemplo la contraposición del Software Libre contra el Software Propietario. En éste caso, la existencia del mismo internet como medio natural de desarrollo de proyectores *Open Source* ha permitido la creación de sinergias gracias a las cuales, la unión de pequeños grupos de desarrolladores afines por un conjunto de motivaciones muy distintas ha permitido la elaboración de proyectos altamente competitivos como Linux.

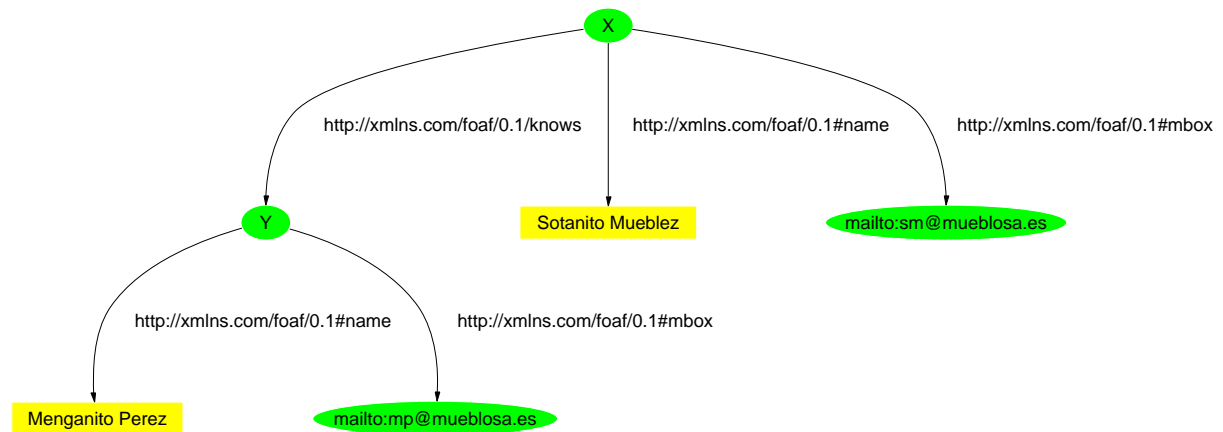


Figura 2: Representación en forma de grafo RDF correspondiente al ejemplo de que expresa que dos personas se conocen dentro del esquema de FOAF

3. Representaciones de RDF

3.1. Grafos RDF

Una forma de representar recursos RDF además de la empleada hasta ahora en modo tabular, se basa en la identificación de la terna sujeto - predicado - objeto como un grafo en el que se conectan los dos nodos correspondientes a los elementos sujeto y objeto a través de un arco que representa el predicado, de modo que el último ejemplo de la sección anterior podría representarse de una manera similar a la expuesta en la figura 2. Hay que notar que se ha seguido el estilo de las especificaciones relativas a RDF del World Wide Web Consortium representando los recursos como óvalos de fondo verde, los objetos literales como recuadros amarillos y los arcos como conexiones direccionales que van del sujeto al objeto de cada una de las sentencias. En el ejemplo, además, se ha vuelto a emplear el ardid de utilizar los denominados *nodos vacíos X e Y* (*desprovistos de URI pese a estar etiquetados*) que en sí mismos no pueden ser referenciados desde otro grafo a no ser que dicha referencia se establezca de modo indirecto a través de las propiedades del recurso (*en particular y según lo discutido anteriormente se trataría del mailbox de cada persona*). Un ejemplo más completo se ilustra a través de la figura 3 dónde se ha reflejado el conjunto de enunciados descrito acerca de Muebloa a partir del primer ejemplo expuesto en los apuntes.

3.2. RDF/XML

Debido al hecho de que ésta introducción a RDF pretende centrarse en aspectos más conceptuales no utilizaremos RDF/XML en lo que sigue. No obstante, es necesario citar que al igual que RDF puede ser representado mediante grafos, también puede representarse en ficheros XML con un formato específico sujeto al esquema RDF/XML. A modo de ejemplo, supongamos que deseamos expresar en RDF/XML que el 2 de Febrero de 2005 es la fecha para la cual tiene previsto

Mueblosa el lanzamiento de un modelo concreto de butaca . El código RDF/XML correspondiente se adjunta a continuación:

```

1. <?xml version="1.0"?>
2. <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.     xmlns:terminologia="http://www.mueblosa.es/terminos/"
4.     <rdf:Description rdf:about="http://www.mueblosa.es/cat#butaca3">
5.         <terminologia:lanzamiento>1/2/05</terminologia:lanzamiento>
6.     </rdf:Description>
7. </rdf:RDF>

```

En la primera línea, se adjunta una cabecera que identifica el tipo de fichero como XML, a continuación se abre el nodo principal del documento RDF/XML con el uso del tag `rdf:RDF` donde se declara el uso del espacio de nombres `xmlns:rdf` correspondiente al URI asociado desde la web del World Wide Web Consortium `http://www.w3.org` con la especificación de RDF. Igualmente se declara el uso de un espacio de nombres propio de Mueblosa bajo el identificador `xmlns:terminologia`. Por último se establece un nodo `rdf:Description` que como su propio nombre indica sirve para establecer una descripción en la que el sujeto se identifica mediante el URI asociado a `rdf:about`, el predicado mediante el tag propio de Mueblosa que en éste caso se trata de `terminologia:lanzamiento` y el objeto es un literal con la cadena de texto representativa de la fecha escogida.

Al margen de lo barroco de la representación, hemos de considerar que pese a que XML parezca complicar algo en principio sencillo, es un buen soporte para el intercambio, análisis y exploración de RDF por parte de programas informáticos. No obstante, existe una característica de XML que lo hace interesante para el consumo humano ... se trata de la posibilidad de declarar espacios nominales (*XML namespaces*) a modo de abreviatura de URIs que recogen terminologías de uso común. Entre otras, se utilizan de forma intensiva por parte de la comunidad de desarrolladores de la web semántica, las siguientes abreviaturas:

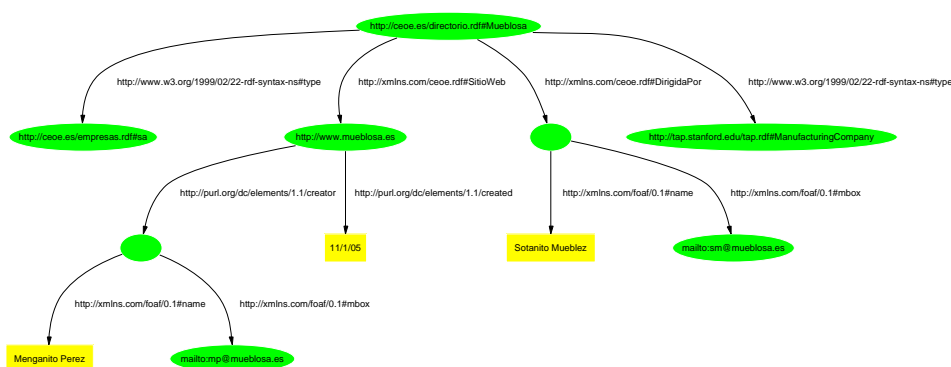


Figura 3: Grafo RDF correspondiente al ejemplo de Mueblosa

- i) **rdf** → <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
- ii) **rdfs** → <http://www.w3.org/2000/01/rdf-schema#>
- iii) **dc** → <http://purl.org/dc/elements/1.1/>
- iv) **owl** → <http://www.w3.org/2002/07/owl#>
- v) **foaf** → <http://xmlns.com/foaf/0.1/>

4. Aplicaciones, ontologías y/o esquemas RDF

A continuación, en éste capítulo, se describen de forma resumida algunos de los proyectos RDF más importantes existentes en la actualidad.

4.1. RDF

Las especificaciones del propio RDF en formato RDF:

- **rdf** → <http://www.w3.org/1999/02/22-rdf-syntax-ns>

En él se describe el vocabulario necesario para establecer sentencias RDF mediante el mismo RDF, se trata por tanto, de unas especificaciones que se autoreferencian constantemente tal y como se puede apreciar en el ejemplo seleccionado a continuación. En primer lugar, se describe la forma de expresar cuál es el sujeto de una sentencia RDF:

Sujeto	Predicado	Objeto
rdf:subject	rdf:type rdfs:isDefinedBy rdfs:label rdfs:comment rdfs:domain rdfs:range	rdf:Property rdf "subject" "The subject of the subject RDF statement" rdf:Statement rdfs:Resource

Tal y como se puede ver, no sólo se alude a `rdf` sino también a RDF Schema (ver sección 4.2 - `rdfs`). En concreto, se establecen los enunciados siguientes: el concepto `rdf:subject` es un tipo (según *type especificado en `rdf`*) de propiedad (según *lo especificado en el mismo `rdf` bajo la referencia a `rdf:Property`*). A continuación se expresa que `rdf:subject` se encuentra definido (según *la significación del predicado recogido en RDF Schema a través del URI `rdfs:isDefinedBy`*) en la misma especificación `rdf`. A lo que sigue el hecho de que la etiqueta establecida en RDF Schema como `rdf:label` asignada a `rdf:subject` es el literal "subject". Se añade un comentario literal a través del predicado `rdfs:comment` al respecto del sujeto de la sentencia (en éste caso *`rdf:subject`*). Y por último se especifica el dominio de aplicación del predicado dentro de la clase enunciado representada por `rdf:Statement` además del rango de valores sobre el que debe aplicarse (en éste caso *todos aquellos recursos descritos mediante `rdfs:Resource`*).

Es decir, se trata de un predicado que permite expresar el hecho de que un recurso concreto (objeto) debe ser identificado como el sujeto de la sentencia que actúa a modo de sujeto del enunciado. La explicación puede parecer similar a un trabalenguas, pero es un gaje común cuando se trata con la metalingüística.

4.2. RDFS - RDF Schema

Especificaciones de RDF Schema

- **rdf** → <http://www.w3.org/2000/01/rdf-schema>

El objetivo de RDF Schema, es el de proporcionar algunas capacidades expresivas enfocadas entre otras cosas en la conveniencia de establecer jerarquías de clases dentro de categorizaciones RDF. Como ejemplo, podemos adjuntar la descripción de la propiedad **rdfs:subClassOf** →

Sujeto	Predicado	Objeto
rdfs:subClassOf	rdf:type	rdf:Property
	rdfs:isDefinedBy	rdfs
	rdfs:label	"subClassOf"
	rdfs:comment	"The subject is a subclass of a class"
	rdfs:domain	rdfs:Class
	rdfs:range	rdfs:Class

4.3. DC - Dublin Core Metadata Initiative

Especificaciones del conjunto de elementos descriptivos Dublin (USA) Core.

- **dc** → <http://purl.org/dc/elements/1.1/>

En el vocabulario se recoge la terminología básica para describir información relacionada con recursos documentales de cualquier tipo. Por ejemplo, podemos destacar la propiedad RDF **publisher** →

Sujeto	Predicado	Objeto
dc:publisher	rdf:type rdfs:isDefinedBy rdfs:label rdfs:comment dc:description dc:issued dc:modified dc:type dc:hasVersion	rdf:Property dc "Publisher" An entity responsible for making the resource available Examples of a Publisher include a person, an organisation, or a service. Typically, the name of a Publisher should be used to indicate the entity. 1999-07-02 2002-10-04 http://dublincore.org/principles/#element http://dublincore.org/history/#publisher-004

4.4. OWL

Web Ontology Language

- **owl** → <http://www.w3.org/2002/07/owl>

Especial atención merecen las especificaciones de OWL, puesto que en ellas se recoge funcionalidad más sofisticada que la expuesta hasta ahora en `rdf`, `rdfs` y `dc`. Para empezar, es necesario decir que OWL se subdivide en tres sublenguajes cada uno incremental respecto al anterior en capacidad expresiva:

- **OWL Lite** → Permite establecer clasificaciones jerárquicas de conceptos de forma similar a RDFS además de proveer de mecanismos restrictivos básicos con respecto a la naturaleza de relaciones como el hecho de asignar cardinalidades (*p. e. enunciar el hecho de que una persona sólo puede tener un padre o una madre biológicos*).
- **OWL DL** → Proporciona una mayor riqueza expresiva con respecto a OWL Lite garantizando la computabilidad de cualquier conclusión en un tiempo finito. Debe su nombre a la Lógica Descriptiva (*Description Logics*), un campo de investigación de reciente actualidad.
- **OWL Full** → Pensado para usuarios que requieran la máxima expresividad dentro de la libertad sintáctica de RDF prescindiendo de cualquier garantía computacional en lo que respecta a la extracción de conclusiones.

Para tratar de centrar las ideas y a modo de ejemplo ilustrativo consideraremos algunas de las propiedades de OWL Lite. Por ejemplo →

Sujeto	Predicado	Objeto
owl:equivalentClass	rdf:type rdfs:label rdfs:comment rdfs:subPropertyOf rdfs:domain rdfs:range	rdf:Property . ^{equivalentClass} ” Two classes may be stated to be equivalent. Eq. classes have the same instances. Equality can be used to create synonymous classes. For example, Car can be stated to be <i>equivalentClass</i> to Automobile. From this, a reasoner can deduce that any individual that is an instance of Car is also an instance of Automobile and vice versa. rdfs:subClassOf rdfs:class rdfs:class

4.5. FOAF - Friend Of A Friend

En secciones anteriores, ya se han discutido algunas de las propiedades de FOAF, no obstante hay que destacar que se trata de una de las primeras especificaciones RDF no centradas en la misma metalingüística. Tal y como se ha expuesto se trata de un vocabulario apropiado para describir relaciones entre personas. Tiene igualmente muchas posibilidades de convertirse en un convenio universalmente aceptado para la identificación de personas en cualquier entorno ya sea el de las relaciones informales, laborales y/o comerciales entre otros; de hecho en su última versión incluye términos como **OnlineEcommerceAccount**, **currentProject** o el enigmático **dnaChecksum** cuyo comentario asociado es:

A checksum for the DNA of some thing. Joke.

5. Software RDF

A día de hoy, ya existe gran cantidad de software para poder trabajar con RDF de forma bastante amigable y cómoda. No obstante, aún no se ha alcanzado la masa crítica de desarrollo para que RDF se consolide como una herramienta de expresión universal. A continuación destacamos algunos ejemplos a nivel de APIs para desarrolladores, o de aplicaciones para usuarios finales que actualmente se encuentran disponibles a modo de proyectos propietarios o libres propuestos indistintamente por colectivos de usuarios y/o programadores o grandes compañías como IBM o Microsoft.

5.1. RDF y Perl → RDF::Simple

En primer lugar, podríamos estudiar el API proporcionado por el módulo Perl de Jo Walsh `RDF::Simple::Parser`. La descripción de dicho módulo es la siguiente:

A simple RDF/XML parser - reads a string containing RDF in XML returning a 'bucket-o-triples' (array of arrays)

exponiendo en su documentación ejemplos del estilo de:

```
use RDF::Simple::Parser;
use LWP::Simple;

my $uri = http://www.zooleika.org.uk/bio/foaf.rdf;
my $rdf = LWP::Simple::get($uri);
my $parser = RDF::Simple::Parser->new(base => $uri)
my @triples = $parser->parse_rdf($rdf);
# returns an array of array references which are triples
```

Para tratar de ver con más detalle como trabajar con el módulo de J. Walsh en Perl, en la sección 5.1.1 se adjunta un script realizado con el propósito de explorar cómodamente vocabularios RDF. El script no hace más que invocar al módulo `RDF::Simple::Parser` para cargar en memoria las estructuras de datos contenidas en un documento RDF cuyo nombre debe pasarse como primer argumento con la extensión `*.rdf` y devuelve un fichero HTML listo para visualizar en un navegador convencional en el que se agrupan las sentencias por sujeto a modo de tablas de propiedades de recursos (ver *Figura 4*). Resulta interesante notar que dicho script se ha utilizado de forma intensiva a la hora de confeccionar estos apuntes.

5.1.1. Código fuente del ejemplo de uso de `RDF::Simple::Parser`

```
1 #!/usr/bin/perl
2 use strict;
3 use RDF::Simple::Parser;
4 my $ns = {
```

```

5  'http://nets.ii.uam.es/iptc' => 'iptc',
6  'http://xmlns.com/foaf/0.1' => 'xmlns:foaf',
7  'http://xmlns.com/wordnet/1.6' => 'xmlns:wordnet',
8  'http://www.w3.org/1999/02/22-rdf-syntax-ns' => 'RDF:rdf',
9  'http://purl.org/dc/elements/1.1' => 'xmlns:dc',
10 'http://www.w3.org/2002/07/owl' => 'xmlns:owl',
11 'http://www.w3.org/2000/01/rdf-schema' => 'xmlns:rdfs',
12 'http://www.w3.org/TR/1999/PR-rdf-schema-19990303' => 'xmlns:rdfs',
13 'http://xmlns.com/wot/0.1' => 'xmlns:wot',
14 'http://www.w3.org/2003/06/sw-vocab-status/ns' => 'xmlns:vs'
15 };
16 my $uri = shift;
17 my ($name) = $uri =~ m/([^\/*]*)\.rdf\Z/;
18 print "$uri -> $name.html\n";
19 open(RDF,"$uri");
20 my @rdf = <RDF>;
21 close(RDF);
22 my $rdf = join(" ",@rdf);
23 my $parser = RDF::Simple::Parser->new(base => $uri);
24 my @triples = $parser->parse_rdf($rdf);
25 open(HTML,">${name}.html");
26 print HTML <<EOF;
27 <html>
28 <head>
29 <title>
30   RDF de $uri
31 </title>
32 <body>
33 <center>
34 <table border="0" width="600">
35 <tr>
36 <td>
37 <font face="arial" size="-1">
38 EOF
39 my $sentence = {};
40 foreach my $triple ( @triples ) {
41   my ( $subj, $pred, $obj) = @{$triple};
42   foreach my $key ( sort keys %{$ns} ) {
43     $subj =~ s/$key/$ns->{$key}/gi;
44     $obj =~ s/$key/$ns->{$key}/gi;
45     $pred =~ s/$key/$ns->{$key}/gi;
46   }
47   push(@{$sentence->{$subj}}, [$pred,$obj]);

```

```

48 }
49 foreach my $subject ( sort keys %{$sentence} ) {
50 print HTML "<table border=\"1\">\n";
51   my $scalar = scalar @{$sentence->{$subject}};
52   print HTML "<tr><th bgcolor=\"EEEEFF\" rowspan=\"$scalar\">
53     $subject</th>\n";
54   print HTML "<td>$sentence->{$subject}[0][0]</td>\n";
55   print HTML "<td>$sentence->{$subject}[0][1]</td></tr>\n";
56   foreach my $i ( 1 .. $#{$sentence->{$subject}} ) {
57     print HTML "<tr><td>$sentence->{$subject}[$i][0]</td>\n";
58     if ( $sentence->{$subject}[$i][0] =~
59       m/comment|description|explanation/ ) {
60       print HTML "<td bgcolor=\"FFEE55\"><strong>
61         $sentence->{$subject}[$i][1]</strong></td></tr>\n";
62     } elsif ( $sentence->{$subject}[$i][0] =~ m/title/ ) {
63       print HTML "<td bgcolor=\"EEFF77\"><strong>
64         $sentence->{$subject}[$i][1]</strong></td></tr>\n";
65     } else {
66       print HTML "<td>$sentence->{$subject}[$i][1]</td></tr>\n";
67     }
68   }
69   print HTML "</table>\n";
70   print HTML "<br></br>\n";
71 }
72 print HTML <<EOF;
73 </font>
74 </td>
75 </tr>
76 </table>
77 </center>
78 </body>
79 </html>
80 EOF
81 close(HTML);

```

5.1.2. Comentarios acerca del ejemplo

En primer lugar, se define el hash `$ns` donde se almacenan un conjunto de abreviaturas típicas de espacios nominales para su posterior sustitución (*líneas 4 → 15*). A continuación se recupera el nombre del fichero donde se encuentra la estructura RDF/XML en la variable `$uri` separando el nombre sin la extensión `*.rdf` en la variable `$name`. Después de emitir algún mensaje por pantalla se procede a la lectura del fichero RDF/XML (*líneas 19 → 22*).

Una vez leído el fichero, se procede a construir una instancia de `RDF::Simple::Parser` y acto seguido se invoca al método `parse_rdf` del objeto construido sobre la cadena de texto que contiene el código recuperado del fichero RDF/XML obteniendo un array de predicados llamado `@triples` (línea 24).

xmlns:foaf/	RDF:rdf#type	xmlns:owl#Ontology
	xmlns:dc/date	\$Date: 2004/09/01 15:37:56 \$
	xmlns:dc/description	The Friend of a Friend (FOAF) RDF vocabulary, described using W3C RDF Schema and the Web Ontology Language.
	xmlns:dc/title	Friend of a Friend (FOAF) vocabulary
	xmlns:rdfs#seeAlso	http://www.w3.org/2001/08/rdfweb/foaf
	xmlns:owl#imports	xmlns:rdfs
	xmlns:owl#imports	xmlns:owl
	xmlns:wot/assurance	foaf.rdf../foafsig
	xmlns:wot/src_assurance	foaf.rdf../htmlfoafsig

xmlns:foaf/Agent	RDF:rdf#type	xmlns:rdfs#Class
	xmlns:vs#term_status	unstable
	xmlns:rdfs#comment	An agent (eg. person, group, software or physical artifact).
	xmlns:rdfs#label	Agent
	xmlns:rdfs#subClassOf	xmlns:wordnet/Agent-3
	xmlns:owl#disjointWith	xmlns:foaf/Document

Figura 4: Vista HTML proporcionada por el explorador de vocabularios RDF implementado en Perl gracias a `RDF::Simple::Parser`

Una vez, tenemos la estructura en forma de array de arrays (*i. e.* `@triples`), abrimos un fichero HTML para volcar la estructura de tablas e imprimimos las cabeceras (líneas 26 → 38). Seguidamente, se declara una estructura tipo referencia a hash `$sentence` que albergará las sentencias *parseadas* agrupadas adecuadamente. El agrupamiento se realiza mediante el bucle de las líneas 40 → 48 en el que se separan sujeto, objeto y predicado, se realizan las sustituciones correspondientes a las abreviaturas contenidas en la tabla `$ns` definida al principio y por último se *empujan* sucesivamente predicados y objetos bajo cada una de las entradas (*i. e.* los distintos sujetos) de la tabla asociativa `$sentence` definida a tal efecto (línea 47).

Con la estructura `$sentence` rellena, se inicia el volcado de las tablas (*bucle de las líneas 49 → 71*). En él se pone especial cuidado en resaltar con colores específicos cualquier comentario, explicación o título para facilitar la lectura del documento final (*líneas 58 → 67*).

Por último se cierra el documento HTML dando fin al volcado y a la ejecución del script (*líneas 72 en adelante*).

6. The GNU General Public License

Version 2, June 1991

Copyright © 1989, 1991 Free Software Foundation, Inc.

59 Temple Place - Suite 330, Boston, MA 02111-1307, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software—to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the

original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
 - a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
 - b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

- c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
 - a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
 - c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source

code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.
6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and “any later version”, you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS