

El mundo de la ontología (para proyecto MOMENT)



Felix Espina
felix.espina@unavarra.es

Índice

1. RDF
2. RDFS
3. RDF Query Languages
4. RDF Update Languages
5. Ontologías
 1. OWL
 2. DL Reasoner
 3. Frameworks
 4. Editores
 5. Buscadores
 6. Plataformas de publicación
6. Enlaces

RDF: Introducción

- RDF: Resource Description Framework
- Estándar W3C:
 - 1ª versión en 1999
 - 2º versión (la usada) en 2004
- Originalmente diseñado como modelo de datos (sintaxis) para metadatos (datos sobre datos)
- Se usa como método general para el modelado de información

RDF: representación

- RDF es un modelo abstracto con diferentes formas de serialización/representación:
 - RDF/XML
 - se suele llamar simplemente RDF porque salió junto con RDF, pero hay que diferenciarlo
 - Notation 3 (N3)
 - Formato no-XML
 - Más fácil de escribir a mano y seguir, porque está basado en notación tabular
 - RDFa (Resource Description Framework attributes)
 - Proveer una forma (grupo de atributos) para añadir metadatos a cualquier lenguaje XML
 - Pensado especialmente como valor añadido para XHTML
 - Turtle (Terse RDF Triple Language)
 - Subgrupo de N3
 - Muy extendido en la web semántica

RDF: representación

- La base: tripletas de recursos
 - sujeto // predicado // objeto
 - Ej: el cielo // es de color // azul
- Los recursos RDF normalmente se identifican/nombran mediante un URI (Uniform Resource Identifier)
- Si no tienen nombre: “blank nodes” o anónimos => no se pueden referenciar

RDF: URI

- Las URI's no tienen que ser enlaces reales accesibles por red
 - Aunque para algunas aplicaciones RDF (RSS o FOAF) si lo son

- Ej: **"New York has the postal abbreviation NY"**

- N3:

```
<urn:states:New%20York> <http://purl.org/doc/terms/alternative> "NY"
```

- XML:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:terms="http://purl.org/doc/terms/"
  <rdf:Description rdf:about="urn:states:New%20York">
    <terms:alternative>NY</terms:alternative>
  </rdf:Description>
</rdf:RDF>
```

RDF: URI

- Es necesario conocer la semántica de las URI's del RDF
 - Existen vocabularios controlados de amplio uso (ej. Dublin Core Metadata) parcialmente mapeados a URI's
 - URI's propias: ¿documentación?

RDF: Aplicaciones

- RSS (RDF Site Summary)
- FOAF (Friend of a Friend)
- DOAP (Description of a Project)
 - <http://trac.usefulinc.com/doap>
 - <http://pypi.python.org/pypi?:action=doap&name=infrac.plone.relations.schema&version=1.0>
- ...
- RDF Validator: <http://www.w3.org/RDF/Validator/>
- Automating the publication of Technical Reports of W3C: <http://www.w3.org/2002/01/tr-automation/tr.rdf>
- Semantic extractor: <http://www.w3.org/2003/12/semantic-extractor.html>

RDF: Visores

- Disco: http://www4.wiwiss.fu-berlin.de/rdf_browser/
- OpenLink RDF Browser:
<http://demo.openlinksw.com/DAV/JS/rdfbrowser/index.html>
- Tabulator:
<http://dig.csail.mit.edu/2005/ajar/release/tabulator/0.7/tab>
- Objectviewer: <http://objectviewer.semwebcentral.org/>

RDFS (RDF Schema)

- Estándar W3C desde 2004
- Provee elementos para poder estructurar recursos (vocabularios) RDF
 - Clases y subclases:
rdfs:Class, rdfs:subClassOf
 - Dominio y rango:
 - rdfs:domain : indica clase del sujeto usando esta propiedad como predicado
 - rdfs:range : indica clase o tipo de dato del objeto usando esta propiedad como predicado

RDFS (RDF Schema)

- `ex:employer` enlaza un sujeto `foaf:Person`

ex:employer rdfs:domain foaf:Person

- `ex:employer` enlaza con un objeto `foaf:Organization`

ex:employer rdfs:range foaf:Organization

- `ex:John` es necesariamente un `foaf:Person` y `ex:CompanyX` es necesariamente un `foaf:Organization`

ex:John ex:employer ex:CompanyX

RDF Query Languages

- DQL
- N3QL: basado en Notation 3
- R-Device
- RDQF: basado en XML
- RDQ: tipo SQL
- RDQL: tipo SQL, precursor de SPARQL
- RQL/RVL: tipo SQL
- **SPARQL**: tipo SQL, estándar W3C, el lenguaje RDF de-facto
- **SeRQL**: tipo SQL, similar a RQL/RVL, propio para Sesame
- Versa: sintaxis compacta (no SQL)
- Adenine: lenguaje de programación escrito en RDF

RDF Query Languages: SPARQL

- SPARQL: SPARQL **Protocol** and RDF **Query** Language

- Ejemplo: todas las capitales de África

PREFIX abc: <http://example.com/exampleOntology# >

SELECT ?capital ?country WHERE {

?x abc:cityname ?capital ;

abc:isCapitalOf ?y .

?y abc:countryname ?country ;

abc:isInContinent abc:Africa .

}

- Variables: prefijo "?" o "\$"
- Permite uso de prefijos y URIs para abreviar

RDF Query Languages: SPARQL

- Implementaciones:
 - Client Side
 - Query Engines
 - Grammars and Parsers
 - Endpoints

<http://esw.w3.org/topic/SparqlImplementations>

RDF Query Languages: SPARQL interfaces: Twinkle

- Una simple interfaz GUI que encapsula un SPARQL query engine (ARP de Jena)
- Sirve para practicar SPARQL
- Es Java (cargas localmente un .jar) y open source

- Ej: Find names and websites of contributors to PlanetRDF:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>  
SELECT ?name ?website  
FROM <http://planetrdf.com/bloggers.rdf>  
WHERE { ?person foaf:weblog ?website ;  
         foaf:name ?name .  
        ?website a foaf:Document  
}
```

RDF Query Languages: SPARQL interfaces: Twinkle

The screenshot shows the Twinkle SPARQL Tools application window. The main area contains a SPARQL query:

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?website
FROM <http://planetrdf.com/bloggers.rdf>
WHERE { ?person foaf:weblog ?website ;
        foaf:name ?name .
        ?website a foaf:Document
}
    
```

Below the query, the results are displayed in a table with two columns: 'name' and 'website'.

name	website
Talis	http://blogs.talis.com/nodalities/
Elias Torres	http://torrez.us/
John Breslin	http://www.johnbreslin.com/blog/
International Semantic Web Conference (ISWC 2008)	http://iswc2008.semanticweb.org/
Andrew Matthews	http://aabs.wordpress.com/
Web Semántica Hoy	http://www.wshoy.sidar.org/index.php
Orri Erling	http://www.openlinksw.com/weblogs/oerling/
Alexandre Passant	http://apassant.net/blog/
Leigh Dodds	http://ldodds.com/blog/
Ebiquity research group UMBC	http://ebiquity.umbc.edu/v2.1/blogger/
Danny Ayers	http://dannyayers.com/
Phil Dawes	http://www.phildawes.net/blog/
Jeen Broekstra	http://jeenbroekstra.blogspot.com/
W3C Semantic Web News	http://www.w3.org/2001/sw/
DOAP Project	http://usefulinc.com/doap
Jo Walsh	http://frot.org/devlog/
Alistair Miles	http://isegserv.itd.rl.ac.uk/blogs/alistair/
Bob DuCharme	http://www.snee.com/bobdc.blog/
Libby Miller	http://planb.nicecupoftea.org/
Benjamin Nowack	http://www.bnode.org/en-semweblog
Ivan Herman	http://ivanherman.wordpress.com/tag/work-related/semantic-web/
Dominique Hazaël-Massieux	http://people.w3.org/~dom/
Uldis Bojars	http://captsolo.net/info/
POWDER WG bloa	http://www.w3.org/bloa/powder

The interface also includes a sidebar with navigation options like 'General', 'In Memory', 'Inferencing', 'Persistent Stores', and 'Remote Services'. The status bar at the bottom indicates 'Ready'.

RDF Query Languages: SPARQL interfaces: Joseki

- Joseki es un motor HTTP que soporta el protocolo y el RDF Query language SPARQL:
 - Manejo de datos RDF en ficheros y BBDD
 - Implementación GET/POST del protocolo
 - Funcionamiento como servidor
 - Funcionamiento como servlet dentro de cualquier servidor java (Tomcat, Jetty)
- Es Java y open source
- Es parte de D2R
- Sparqler (online demo): <http://sparql.org/>

RDF Query Languages: SeRQL

- SeRQL: Sesame RDF Query Language
- Desarrollado por Aduna para usarlo en Sesame
- Combina lo mejor de otros lenguajes (RQL, RDQL, N3QL, ...) + características propias
- **La diferencia:** dos formas de hacer consultas:
 - SELECT: la devolución de tablas con los posibles valores que pueden tomar las variables en nuestra consulta
 - CONSTRUCT: devolviendo el resultado en forma de subgrafo para usar en otras consultas “Select”

RDF Query Languages: SeRQL

- Ejemplos
 - Devolver todas las URIs de clases:

```
SELECT C FROM {C} rdf:type {rdfs:Class}
```

- Devuelve un grafo de relaciones padre-hijo:

```
CONSTRUCT {Parent} ex:hasChild {Child}  
FROM {Child} ex:hasParent {Parent}  
USING NAMESPACE ex=<http://ex.org/things#>
```

RDF Update Languages

- Único encontrado: SPARUL o SPARQL/Update
- No es un estándar, desarrollado por HP
- Capacidades:
 - Insertar nuevas tripletas en RDF
 - Borrar tripletas del RDF
 - Realizar un grupo de operaciones de actualización como una única acción
- Sintaxis análoga a SPARQL

RDF Update Languages

- Ej: borrar una entrada, meter otra entrada

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
DELETE DATA FROM <http://example/bookStore> { <http://example/book3>  
  dc:title "Fundamentals of Compiler Desing" } INSERT DATA INTO  
<http://example/bookStore> { <http://example/book3> dc:title  
  "Fundamentals of Compiler Design" }
```

- Ej: borrar todas las entradas viejas (<año 2000)

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
```

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
DELETE { ?book ?p ?v } WHERE { ?book dc:date ?date . FILTER ( ?date <  
  "2000-01-01T00:00:00"^^xsd:dateTime ) ?book ?p ?v }
```

Ontologías: OWL

- Lenguaje estándar para ontologías
- Capa superior a RDF/RDFS
- Añade posibilidad de razonamiento sobre los contenidos
- 3 niveles de complejidad/restricciones:
 - OWL Lite
 - **OWL DL (Description Logics)**
 - OWL Full

Ontologías: OWL

- Algunos recursos interesantes:
 - OWL Ontology Patcher:
<http://www.mygrid.org.uk/OWL/Patcher>
 - OWL Ontology HTML Presentation:
<http://www.mygrid.org.uk/OWL/Presentation>
 - Listado de recursos:
http://en.wikipedia.org/wiki/Ontology_%28computer_sci
 - Tutorial OWL:
<http://www.cs.man.ac.uk/~horrocks/ISWC2003/Tutorial>,

Ontologías: DL Reasoner: Pellet

- Open source OWL DL Reasoner
- En Java
- Soporta la totalidad expresividad de OWL DL y casi la totalidad de OWL 1.1
- Tiene todos los servicios típicos de inferencia:
 - Consistency checking: comprueba que las ontologías no tienen hechos contradictorios
 - Concept satisfiability: determinar si una clase puede tener una instancia
 - Classification: crear la jerarquía completa de clases
 - Realization: encontrar la clase más específica a la que pertenece un individuo

Ontologías: DL Reasoner: Pellet

- Múltiples interfaces:
 - OWLSight: web demostrativa
<http://pellet.owldl.com/owlsight/>
<http://www.mindswap.org/2003/pellet/demo.shtml>
 - Línea de comandos
 - API de programación
 - Interfaces para Jena y para la librería OWL-API de la Universidad de Manchester
 - Un servidor DIG[*] para poder usar Pellet desde clientes (p.e. el editor Protégé)
 - Integración directa con el editor de ontologías SWOOP y con la próxima versión de Protégé (v4.0)

[*] DL Implementation Group (<http://dl.kr.org/dig/>): especificación de interfaz común para DL Reasoners

Ontologías: DL Reasoner: FaCT++

- Open source OWL DL Reasoner
- En C++
- Soporta la totalidad expresividad de OWL DL y parte de OWL 1.1
- Servicios parecidos a Pellet
- Múltiples interfaces (como Pellet), con diferencias:
 - API en C++, además de en Java
 - No interfaz para Jena
 - No integración directa con el editor de ontologías SWOOP

Ontologías: DL Reasoner: Racer

- RACER: Renamed ABox and Concept Expression Reasoner
- Análogo a los 2 anteriores, pero comercial (RacerPro, licencia miles \$)
- Algunas diferencias:
 - Incremental query answering for information retrieval tasks (retrieve the next n results of a query)
 - nRQL: new Racer Query Language
- Software relacionado:
 - RacerPorter: interfaz de usuario gráfica para RacerPro a través de TCP/IP
 - RacerPlus: paquete integrado RacerPro+RacerPorter

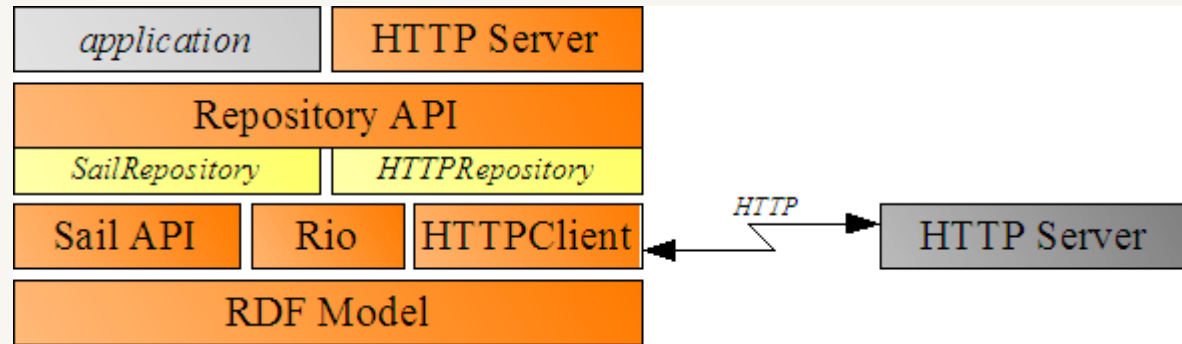
Ontologías: Frameworks: Sesame

- Framework open-source para inferencia y consulta de RDF Schema, NO OWL (ontología)
- Java 5
- Diversos lenguajes de query => SeRQL
- Almacenamiento de información: repositorios RDF => objeto en Java o como una base de datos común

Ontologías: Frameworks: Sesame

- Componentes del modelo:

<http://www.openrdf.org/doc/sesame2/2.1.2/users/ch03.html#figure-sesame-components>



- Storage And Inference Layer (Sail) API: low level System API. Abstracción de detalles de almacenamiento e inferencia. Ej:
 - MemoryStore: almacenamiento en memoria principal
 - NativeStore: estructuras de datos en disco dedicadas)

Ontologías: Frameworks: Sesame

- Rio (RDF I/O): parsers y writers para RDF. Ej:
 - read/write de RDF/XML
 - write de N3
- HTTP Client
- Repository API: high level API. Diversos métodos orientados al desarrollador para manejar datos RDF.
- HTTP Server: Java Servlets que implementan protocolo sobre HTTP para acceder a repositorios Sesame. No es una interfaz web de acceso al repositorio.

Ontologías: Frameworks: Sesame

- Relacionados:
 - OpenRDF Workbench: interface web para consulta de repositorio Sesame
 - Elmo: API sobre Sesame para aplicaciones web semánticas, permitiendo consultar ontologías conocidas (Dublin Core, RSS, FOAF)
 - OWLIM Semantic Repository: añade “capacidades” de ontología. Empaquetado como un SAIL. Co-financiado por varios proyectos FP6.
 - Protégé RDF(s)-DB Backend Plugin
 - RDF2GO: abstracción de almacenamiento de tripletas. Oculta sistema de almacenamiento. Soporta almacenamiento en Sesame y Jena
 - Adaptadores Jena-Sesame, Sesame-Jena
 - Wrappers, librerías, APIs en diversos lenguajes: PHP5, Python, Perl, Ruby, C#

Ontologías: Frameworks: Manchester OWL-API

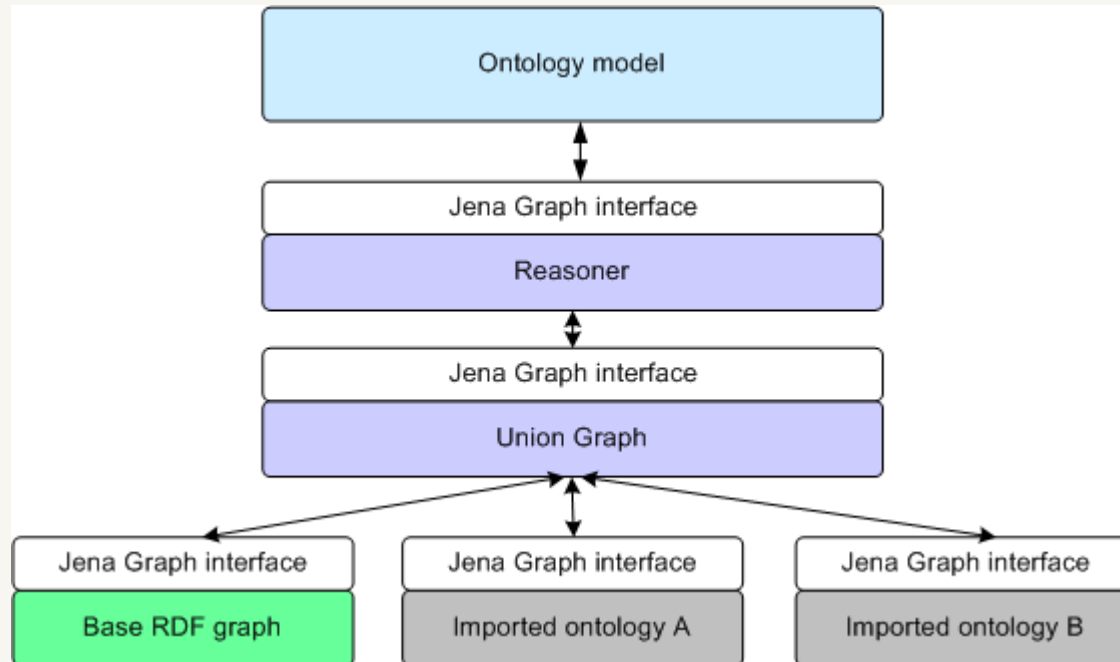
- Interfaz e implementación open source en Java de OWL 2.0 que incluye OWL-Lite, OWL-DL algunos elementos de OWL-Full
- También conocida como Wonderweb API
- Uso nativo de Pellet y FaCT++
- Algunas de las características:
 - API para OWL 2 e implementación en memoria eficiente
 - Parser/writer para RDF/XML, OWL/XML, OWL Functional Syntax, Turtle
 - Integración directa de Pellet y FaCT++
 - Soporte para “black-box debugging”
- Parte del proyecto CO-ODE
 - Desarrollan junto con Standford el editor Protégé 4.0 que re-implementa una interfaz familiar sobre OWL-API

Ontologías: Frameworks: Jena

- Framework (de HP) open source en Java para aplicaciones web semánticas, con soporte para el lenguaje OWL y ontologías
- Incluye:
 - RDF Parser => ARP
 - RDF API
 - Read/write de RDF en RDF/XML, N3 y N-Triples
 - Inference API => incluye Reasoner propio limitado (OWL/Lite). Mediante DIG se pueden usar los anteriores
 - OWL API
 - Almacenamiento en memoria y persistente => SDB
 - SPARQL query engine => ARQ

Ontologías: Frameworks: Jena

- OWL API + Inference API



Ontologías: Frameworks: Jena: ARP

- ARP : RDF/XML parser
- Características:
 - Command-line:
 - Filtro RDF/XML => N-Triple
 - Chequeo de errores RDF/XML
 - Procesa RDF/XML standalone y embedded
 - Streaming parser, perfecto para ficheros grandes
 - Soporta SAX y DOM para integración con fuentes XML no-ficheros
- Cumple completamente con las recomendaciones RDFS y RDF Test Cases

Ontologías: Frameworks: Jena: ARP

- Cumple con los siguientes estándares y recomendaciones:
 - daml:collection
 - `rdf:parseType='daml:collection'`
 - xml:lang
 - Los tags de lenguaje chequeados contra RFC1766, RFC3066, ISO639-1, ISO3166
 - xml:base
 - URI
 - Todas las referencias URI chequeadas contra RFC2396.
 - XML Names
 - Todos los `rdf:ID` chequeados contra la especificación XML Names.
 - Unicode Normal Form C
 - Para los "string literals".
 - XML Literals
 - `rdf:parseType='Literal'` se procesa respetando los namespaces, procesando las instrucciones y los comentarios XML.
 - Relative Namespace URI references
 - Checked in light of the W3C XML Plenary decision.

Ontologías: Frameworks: Jena: SDB

- SDB: componente para el almacenamiento y query RDF pensado para soportar SPARQL
- Se puede acceder al almacenamiento SDB via command line scripts o via SDB/Jena API.
- Almacenamiento mediante BBDD:
 - Oracle 10g (incluyendo OracleXE)
 - Microsoft SQL Server 2005 (incluyendo MS SQL Express)
 - DB2 9 (incluyendo DB2 9 Express)
 - PostgreSQL v8.2
 - MySQL v5.0.22
 - Apache Derby v10.2.2.0
 - H2 1.0.71
 - HSQLDB 1.8.0

Ontologías: Frameworks: Jena: ARQ

- ARQ: query engine de Jena
- Características:
 - Multiple query languages:
 - SPARQL
 - RDQL
 - ARQ, the engine's own language (access to experimental features)
 - General purpose engine
 - Remote access engines
 - Rewriter to SQL

RESTO SIN PREPARAR

Ontologías: Editores: SWOOP

- SWOOP is a tool for creating, editing, and debugging OWL ontologies. It was produced by the MIND lab at University of Maryland, College Park, but is now an open source project with contributors from all over.
- Uses Manchester OWL API
- Características:
 - * It is simple to load ontologies from the web and to navigate within and between them.
 - * Multiple ontologies may be loaded at the same time.
 - * Ontologies, classes, properties, and individuals are rendered in a high level, accessible manner.
 - * One can "view the source" of ontologies and their entities in a number of common syntaxes (e.g. RDF/XML, the OWL Abstract Syntax, Turtle).
 - * OWL reasoners can be integrated for subsumption, consistency checking etc. -- default reasoners include a RDFS-like simple reasoner and Pellet, a Description Logic Tableau Reasoner.
 - * Ontology change management with extensive rollback and undo mechanisms
 - * Share Annotations on Ontologies using the Annotea Protocol. Also attach and distribute Ontology Change sets with Annotations
 - * Search across multiple ontologies and 'find all references' of an OWL named entity
 - * Compare entities using a Resource Holder
 - * Export Ontologies directly to a remote WebDav store

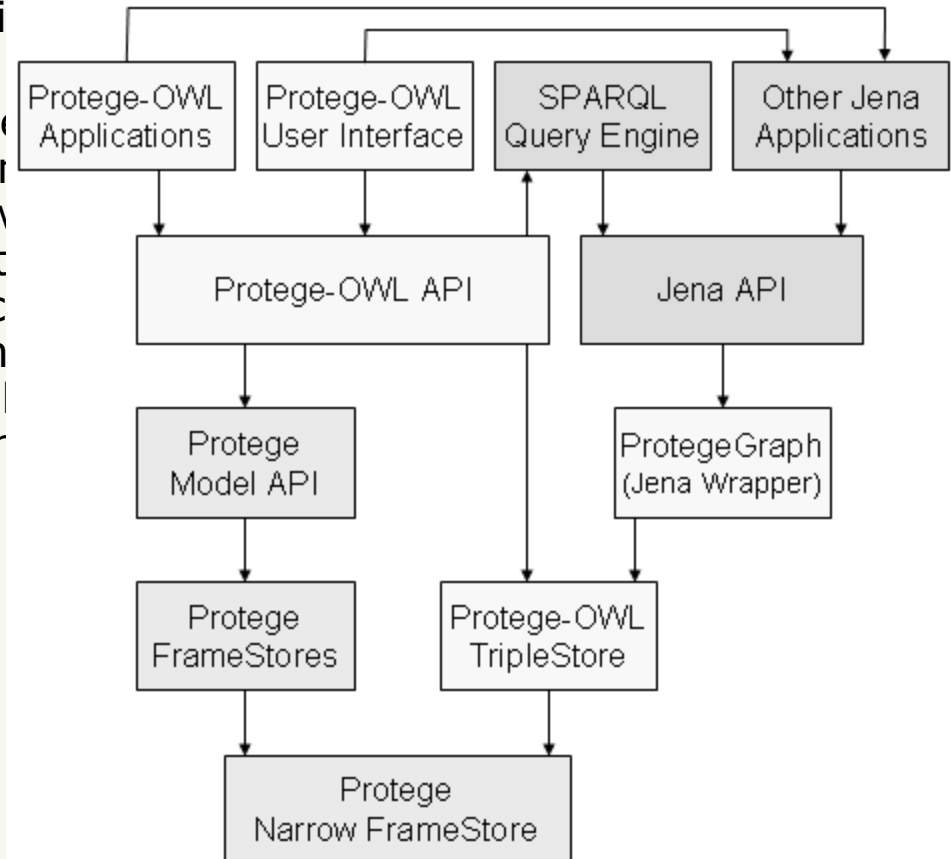
 - * Debug Ontologies using Pellet (explanations for unsatisfiable classes & inconsistent ontologies)

Ontologías: Editores: Protégé

- Protégé is a free, open-source platform that provides a growing user community with a suite of tools to construct domain models and knowledge-based applications with ontologies. At its core, Protégé implements a rich set of knowledge-modeling structures and actions that support the creation, visualization, and manipulation of ontologies in various representation formats. Protégé can be customized to provide domain-friendly support for creating knowledge models and entering data. Further, Protégé can be extended by way of a plug-in architecture and a Java-based Application Programming Interface (API) for building knowledge-based tools and applications.
- The Protégé platform supports two main ways of modeling ontologies:
 - * The Protégé-Frames editor enables users to build and populate ontologies that are frame-based, in accordance with the Open Knowledge Base Connectivity protocol (OKBC). In this model, an ontology consists of a set of classes organized in a subsumption hierarchy to represent a domain's salient concepts, a set of slots associated to classes to describe their properties and relationships, and a set of instances of those classes - individual exemplars of the concepts that hold specific values for their properties.
 - * The Protégé-OWL editor enables users to build ontologies for the Semantic Web, in particular in the W3C's Web Ontology Language (OWL). "An OWL ontology may include descriptions of classes, properties and their instances. Given such an ontology, the OWL formal semantics specifies how to derive its logical consequences, i.e. facts not literally present in the ontology, but entailed by the semantics. These entailments may be based on a single document or multiple distributed documents that have been combined using defined OWL mechanisms" (see the OWL Web Ontology Language Guide).

Ontologías: Editores: Protégé

- Protégé-OWL's flexible architecture makes it easy to configure and extend the tool. Protégé-OWL is tightly integrated with Jena and has an open-source Java API for the development of custom-tailored user interface components or arbitrary Semanti
- The key to this integration is the level "triple" representation of the store mechanism, which has been TripleStore classes. In the Jena world called Graph and Model. The Protégé Jena Graph, so that any read access the Protégé triples. In order to manipulate Protégé-OWL API must be used. Use of Jena methods for querying, with



Ontologías: Buscadores

- <http://motoresrecuperacionxmlrdf.iespana.es/swoogle.html>
- <http://motoresderecuperacion.50webs.com/motoresdebusqueda.html>
- <http://buscadoresweb.iespana.es/>
- <http://www.schemaweb.info/default.aspx>
- <http://www.daml.org/ontologies/>
- <http://swoogle.umbc.edu/>

Ontologías: Plataformas de publicación: D2R

- <http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/>
- <http://www4.wiwiss.fu-berlin.de/bizer/d2r-server/publishing/>
- <http://www4.wiwiss.fu-berlin.de/bizer/d2rq/>
- D2RQ

Ontologías: Plataformas de publicación: Virtuoso

- http://en.wikipedia.org/wiki/Virtuoso_Universal_Server
- <http://www.openlinksw.com/>
- <http://virtuoso.openlinksw.com/main/featurematrix/index.htm>
- <http://virtuoso.openlinksw.com/wiki/main/Main/>

Ontologías: Plataformas de publicación: ARC

- <http://arc.semsol.org/>

Enlaces

- RDF:
 - http://en.wikipedia.org/wiki/Resource_Description_Framework
 - <http://www.w3.org/RDF/>
 - <http://www.rdfabout.net/>
- N3: http://en.wikipedia.org/wiki/Notation_3
- RDFa: <http://en.wikipedia.org/wiki/RDFa>
- Turtle: http://en.wikipedia.org/wiki/Turtle_%28syntax%29
- URI: <http://en.wikipedia.org/wiki/URI>
- PURL: <http://purl.org>: Persistent Uniform Resource Locator (servicio de redireccionamiento HTTP)
- RDFS: <http://www.w3.org/TR/rdf-schema/>
- SPARQL: <http://en.wikipedia.org/wiki/SPARQL>
- SPARQL Query Language: <http://www.w3.org/TR/rdf-sparql-query/>
- SPARQL Query Results XML format: <http://www.w3.org/TR/rdf-sparql-XMLres/>
- SPARQL Protocol: <http://www.w3.org/TR/rdf-sparql-protocol/>
- Twinkle: <http://www.ldodds.com/projects/twinkle/>
- Joseki: <http://www.joseki.org/>
- SeRQL: <http://www.openrdf.org/doc/SeRQLmanual.html>
- SPARUL: <http://jena.hpl.hp.com/~afs/SPARQL-Update.html>

Enlaces

- OWL:
 - http://en.wikipedia.org/wiki/Web_Ontology_Language
 - <http://www.w3.org/2004/OWL/>
 - [http://en.wikipedia.org/wiki/Ontology_\(computer_science\)](http://en.wikipedia.org/wiki/Ontology_(computer_science))
- Pellet: <http://pellet.owldl.com/>
- FaCT++: <http://code.google.com/p/factplusplus/>
- Racer: <http://agraph.franz.com/racer/>
- Sesame:
 - <http://www.aduna-software.com/technologies/sesame/overview.view>
 - <http://www.openrdf.org/>
- OWLIM Semantic Repository: <http://ontotext.com/owlim/>
- RDF2GO: <http://rdf2go.semweb4j.org/>
- OWL-API de Manchester: <http://owlapi.sourceforge.net/>
- Jena: <http://jena.sourceforge.net/>
- ARP: <http://jena.sourceforge.net/ARP/>
- RDF API: http://jena.sourceforge.net/tutorial/RDF_API/index.html
- SDB: <http://jena.hpl.hp.com/wiki/SDB>
- ARQ: <http://jena.sourceforge.net/ARQ/>
- Lista de Editores de ontología: http://en.wikipedia.org/wiki/Ontology_editor
- SWOOP: <http://code.google.com/p/swoop/>
- Protégé: <http://protege.stanford.edu/>