

Comparativa entre dues BD semàntiques: Owlim-Lite i Virtuoso

Pello Olascoaga Figuerola
Enginyeria en Informàtica

Consultor
Joan Anton Pérez Braña

11/06/2013



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

© Pello Olascoaga Figuerola

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel•lectual.

FITXA DEL TREBALL FINAL

Títol del treball:	Comparativa entre dues BD semàntiques: Owlim-Lite i Virtuoso
Nom de l'autor:	Pello Olascoaga Figuerola
Nom del consultor:	Joan Pérez Braña
Data de lliurament (mm/aaaa):	06/2013
Àrea del Treball Final:	XML i Web Semàntica
Titulació:	Enginyeria Tècnica Informàtica de Gestió
Resum del Treball (màxim 250 paraules):	
<p>L'objectiu fonamental d'aquest treball és estudiar els conceptes bàsics de la web semàntica mitjançant l'estudi de dos sistemes de gestió de bases de dades (SGBD) utilitzats en el context de la web semàntica, Owlim-Lite i Virtuoso, que ens permetran dissenyar i implementar una ontologia.</p> <p>D'una banda, és realitzarà una comparativa rigorosa d'aquests SGBD, s'identificaran les diferències entre ells, els avantatges i desavantatges que tenen, limitacions, rendiment, com emmagatzemen la informació, com realitzen consultes, etc.</p> <p>D'altra banda, un cop realitzat l'estudi comparatiu entre tots dos SGDBs, escollirem un dels SGBD estudiats i realitzarem un exemple d'ús, mitjançant la creació d'una ontologia amb Protégé.</p>	

Abstract (in English, 250 words or less):

The main goal of this project is to study the main concepts of semantic web through the analysis of Virtuoso and Owlim, two semantic Database Management Systems, that will help us to design and implement a new ontology.

On the one hand, an exhaustive comparative between those two DBMS will be presented, identifying the main differences, benefits and drawbacks, limitations, performance issues, storing capabilities, statement execution, etc.

On the other hand, once the analysis will be completed, one of DBMS under analysis will be selected, to evaluate how it works using the data provided by the ontology recently created.

Paraules clau (entre 4 i 8):

Web Semàntica
SGDB Semàntic
Virtuoso
Owlim-Lite
Ontologia
SPARQL

Índex

1.	Introducció.....	1
1.1	Context i justificació del Treball.....	1
1.2	Objectius del Treball	1
1.3	Enfocament i mètode seguit.....	2
1.4	Planificació del projecte	3
1.4.1	Tasques	3
1.4.2	Fites	5
1.4.3	Calendari	5
1.4.4	Diagrames de Gannt.....	6
1.5	Breu sumari de productes obtinguts	7
1.6	Breu descripció dels altres capítols de la memòria.....	8
2.	Web Semàntica.....	9
3.	Ontologies	13
3.1	Introducció.....	13
3.2	Disseny i implementació	13
3.2.1	Domini i abast de l' ontologia	14
3.2.2	Reutilització d'ontologies existents	14
3.2.3	Enumeració dels termes importants de l' ontologia	14
3.2.4	Definició de classes i jerarquia.....	15
3.2.5	Definició de les propietats	16
3.2.6	Definició de les característiques de les propietats	18
3.2.7	Creació de les instàncies	20
3.2.8	Transformació de l' ontologia a format RDF	23
4.	SGDB utilitzats en el context de la web semàntica.....	24
4.1	Què és un SGDB?	24
4.2	SGDB Web Semàntica.....	25
4.3	Avantatges	26
4.4	SGDBs disponibles	29
5.	Llenguatges de consulta en magatzems de dades semàntiques	31
5.1	Llenguatges de consulta XML.....	31
5.2	Llenguatges de consultes RDF	31
5.3	SPARQL.....	32
6.	Virtuoso	33
6.1	Característiques tècniques.....	33
6.2	Plataformes suportades	34
6.3	Política de llicències.....	34
6.4	Arquitectura	35
6.5	Extensions.....	36
6.6	Emmagatzemat	37
6.7	Càrrega de dades	37
6.8	Consultes	39
6.9	Instal·lació	40
7.	Owlim-Lite	42
7.1	Característiques tècniques.....	42
7.2	Plataformes suportades	43

7.3	Política de llicències	43
7.4	Arquitectura	44
7.5	Extensions	45
7.6	Emmagatzemat	45
7.7	Càrrega de dades	46
7.8	Consultes	49
7.9	Instal·lació	50
8.	Proves de rendiment	52
8.1	Joc de dades	52
8.2	Equip de proves	52
8.3	Càrrega de dades	53
8.4	Consultes	57
8.4.1	Consulta 1	57
8.4.2	Consulta 2	59
8.4.3	Consulta 3	61
8.4.4	Consulta 4	62
8.4.5	Consulta 5	64
9.	Conclusions	66
10.	Glossari	69
11.	Bibliografia i Referències	70

Llista de figures

Figura 1. Arquitectura per capes de la web semàntica	10
Figura 2. Relació entre les classes i els object properties	17
Figura 3. Arquitectura web semàntica	25
Figura 4. Diferències entre el model relacional, jeràrquic i semàntic	27
Figura 5. Arquitectura de Virtuoso	35
Figura 6. Arquitectura d'Owlim-Lite	44

1. Introducció

1.1 Context i justificació del Treball

El Treball fi de carrera (TFC) té com a objectiu principal mostrar l'assoliment de l'aprenentatge que s'ha dut a terme al llarg dels estudis d'Enginyeria Tècnica en Informàtica mitjançant la realització d'un treball de síntesi dels coneixements adquirits en altres assignatures de la carrera i que requereixi posar-los en pràctica conjuntament en un treball concret.

Aquest objectiu es desglossa en els següents:

- Analitzar un problema complex de tipus pràctic transformant-lo en un projecte informàtic.
- Planificar i estructurar el desenvolupament del projecte mitjançant l'elaboració d'un pla de treball aplicant una metodologia adient.
- Treballar a fons els aspectes formals del desenvolupament de projectes.
- Sintetitzar una solució viable i realista al problema proposat.
- Elaborar una memòria del projecte segons una estructura prefixada.
- Elaborar una presentació del desenvolupament i resultats finals del projecte.

1.2 Objectius del Treball

Actualment les bases de dades estan evolucionant davant l'aparició de nous reptes en la gestió de la informació, com per exemple els que es relacionen amb l'estàndard XML i la Web Semàntica. La proliferació de la Web comporta la necessitat de tractar dades semi-estructurades. XML i derivats com RDF i OWL són els llenguatges que, avui dia, s'utilitzen per intercanviar informació entre diferents sistemes d'informació donada la flexibilitat que proporcionen.

En els últims anys estan apareixent nous sistemes gestors de bases de dades per recolzar a aquest nou paradigma semàntic, oferint un nou ventall de possibilitats per a poder tractar, emmagatzemar i gestionar aquest nou tipus d'informació semi-estructurada, alhora que permeten la comunicació entre sistemes de manera transparent a l'usuari (com per exemple usant una filosofia de dades enllaçades - linked data). D'aquestes bases de dades, una de les més conegudes de codi obert és OpenLink Virtuoso.

L'objectiu fonamental d'aquest treball és estudiar els conceptes bàsics de la web semàntica mitjançant l'estudi de dos sistemes de gestió de bases de dades (SGBD) utilitzats en el context de la web semàntica, Owlím-Lite i Virtuoso. És a dir, d'una banda l'estudiant haurà de realitzar una comparativa rigorosa

d'aquests SGBD, identificar les diferències entre ells, els avantatges i desavantatges que tenen, limitacions, rendiment, com emmagatzemen la informació, com realitzen consultes, etc. D'altra banda, l'estudiant haurà de seleccionar un dels SGBD estudiats i realitzar un exemple d'ús, mitjançant la creació d'una ontologia i, si el temps ho permet, una petita aplicació.

Aquest objectiu es concreta en els següents:

- Estudiar els conceptes bàsics de la web semàntica.
- Conèixer l'estructura i representació d'alguns llenguatges de representació d'informació a la Web.
- Conèixer l'organització i estructura de les BD:XML natives així com l'estat actual d'utilització d'alguns sistemes concrets.
- Analitzar els llenguatges de consulta actuals que incorporen els Sistemes de BD:XML natives.

En el cas d'XML (relacionant-lo amb les bases de dades) s'ofereix la possibilitat de treballar amb llenguatges de consultes i llenguatges de definició d'esquemes basats en XML, com per exemple XQuery i SPARQL. Tot això inclou l'estudi dels orígens d'XML, els objectius de disseny directament relacionats amb les bases de dades i el paper que hi juga el W3C (World Wide Web Consortium), així com les seves diferències amb el model relacional.

De forma resumida podríem dir que l'objectiu del projecte és fer un anàlisi entre dos SGDB de web semàntica (Virtuoso i Owlím-Lite). Aquest anàlisi és força ambiciós, tenint en compte el desconeixement tant de les aplicacions sota anàlisi com de les tecnologies associades a elles. Per tant, l'objectiu fonamental serà adquirir el coneixement necessari sobre la web semàntica que ens permeti dur a terme un anàlisi de les aplicacions seleccionades rigorós i cert.

1.3 Enfocament i mètode seguit

Un cop analitzats els reptes proposats en el TFC, i per tal d'assolir les fites requerides, he decidit dividir el treball en cinc grans àrees:

- Planificació i gestió de dependències
- Recerca d'informació
- Disseny i implementació d'una ontologia
- Estudi de les aplicacions seleccionades (Virtuoso i Owlím-Lite)
- Vídeo de presentació

La temporització de les tasques sorgides d'aquestes àrees de treball, ha estat alineada amb l'entrega de les diferents PACs del projecte.

1.4 Planificació del projecte

En aquesta secció es fa un estudi dels lliuraments del projecte, d' on ha de sortir les tasques necessàries per dur-los a terme. Aquestes tasques, un cop definides, ens permetran planificar amb molta més precisió, els esforços necessaris per completar cadascuna de les PACs previstes tenint en compte les fites definides i la nostra disponibilitat.

1.4.1 Tasques

Com ha resultat d'aquest anàlisi, obtenim les següents tasques:

		Tasca	Precedents
1		Inici (tasques preliminars)	
	1.1	Descarregar documentació. Enunciat, exemples i recomanacions	
	1.2	Instal·lació de programari	
	1.3	Llegir documentació	1.2
	1.4	Selecció opció projecte. Escollir una de les dues opcions disponibles	1.3
2		PAC1 – Pla de treball	
	2.1	Recerca informació	1.3
	2.2	Redacció del pla de treball (excepte seccions Anàlisi i Planificació)	2.1
	2.3	Anàlisi. Estudi d' on sortiran les tasques necessàries	2.2
	2.4	Planificació	2.3
	2.5	Lliurament	2.4
	2.6	Revisió entrega (a partir dels comentaris sorgits de la correcció realitzada pel consultor)	2.5
3		PAC2	
	3.1	Estudi dels sistemes de gestió de bases de dades (SGBD) utilitzats en el context de la web semàntica en general	2.5
	3.2	Instal·lació i configuració del SGDB seleccionats (Owlím-Lite i Virtuoso)	3.1
	3.3	Identificació de les característiques i	3.2

		funcionalitats específiques dels SGBD (comparativa entre Owlím-Lite i Virtuoso)	
	3.4	Definició de la instància (Ontologia)	3.3
	3.5	Lliurament	3.4
	3.6	Revisió entrega (a partir dels comentaris sorgits de la correcció realitzada pel consultor)	3.5
4	PAC3		
	4.1	Definició del conjunt de proves	3.5
	4.2	Proves Funcionals i de rendiment	4.1
	4.3	Redacció resum resultats	4.2
	4.4	Conclusions	4.3
	4.5	Lliurament	4.4
	4.6	Revisió entrega (a partir dels comentaris sorgits de la correcció realitzada pel consultor)	4.5
5	Memòria i presentació		
	5.1	Síntesi. Selecció de les parts del treball que ens seran útils per la presentació	4.5
	5.2	Presentació. Desenvolupament de la presentació	5.1
	5.3	Lliurament	5.2
6	Debat		
	6.1	Preguntes i respostes	5.3

1.4.2 Fites





Per a la realització del TFC, tenim establertes una sèrie de fites, que han de permetre realitzar un seguiment del desenvolupament del projecte. Les fites establertes són:

Fita	Data
Lliurament PAC1	11/03/13
Lliurament PAC2	15/04/13
Lliurament PAC3	21/05/13
Lliurament Presentació i Memòria	11/06/13
Inici debat	25/06/13
Fi debat	27/06/13

1.4.3 Calendari

Per tal de realitzar una planificació amb garanties, és necessari conèixer la quantitat de temps disponible per a realitzar cada una de les tasques. En termes generals, durant la setmana laboral, la dedicació serà de 2 hores diàries excepte els divendres. Els dissabtes dedicarem 2 hores de forma regular, i els diumenges (o festius) podrem dedicar fins a 4 hores de forma regular. En els períodes de vacances la dedicació serà de 2 hores diàries, de dilluns a diumenge. Amb aquesta informació obtenim el següent calendari de disponibilitat (en blau dedicació de 2 hores i en vermell 4 hores).

Març							Abril							Maig						
				1	2	3	1	2	3	4	5	6	7			1	2	3	4	5
4	5	6	7	8	9	10	8	9	10	11	12	13	14	6	7	8	9	10	11	12
11	12	13	14	15	16	17	15	16	17	18	19	20	21	13	14	15	16	17	18	19
18	19	20	21	22	23	24	22	23	24	25	26	27	28	20	21	22	23	24	25	26
25	26	27	28	29	30	31	29	30						27	28	29	30	31		
Juny																				
					1	2														
3	4	5	6	7	8	9														
10	11	12	13	14	15	16														
17	18	19	20	21	22	23														
24	25	26	27	28	29	30														

	Disponibilitat 2 hr/dia
	Disponibilitat 4 hrs/dia
	No disponible
	Lliurament (*)

(*) En les dates de lliurament la disponibilitat serà 2 de hores en cas de jornada laborable i de 4 hores en cas de festiu.

(**) La setmana de debat hem comptabilitzat 2 hores de dedicació diària durant tot el període.

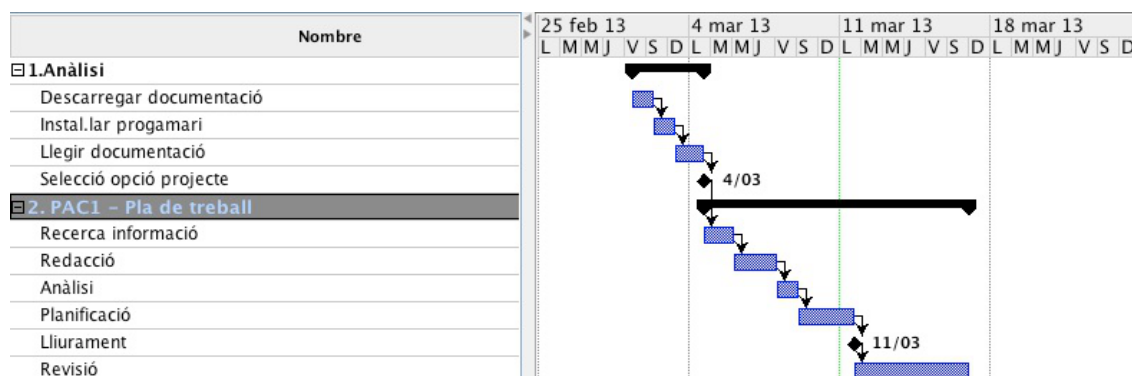
Per tant, la nostra disponibilitat és:

Març	59 hores
Abril	64 hores
Maig	66 hores
Juny	56 hores
Total	245 hores

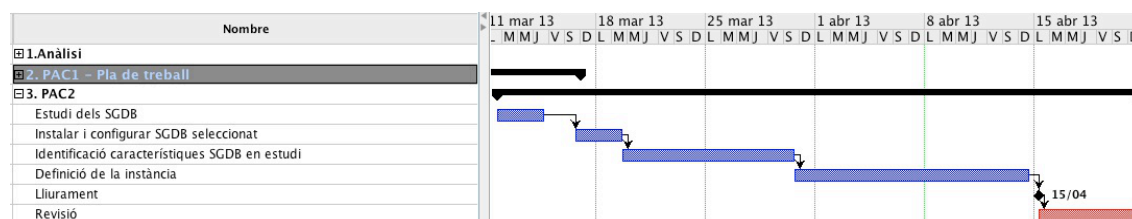
1.4.4 Diagrames de Gannt

Les següents imatges són el resultat de la generació d'un diagrama de Gannt amb l'aplicació OpenProject, a partir de la planificació del projecte.

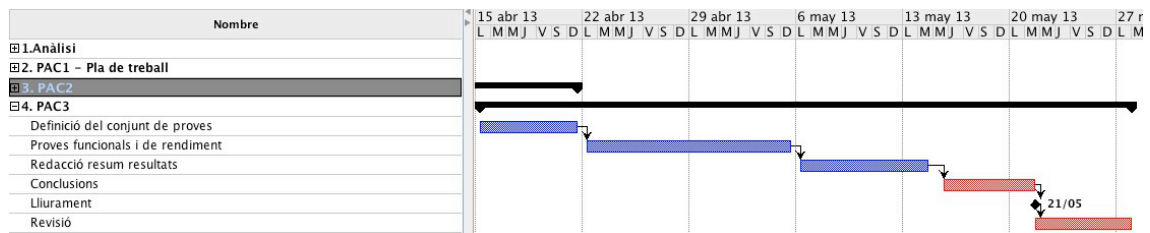
Fase preliminar (anàlisi) i Pla de treball



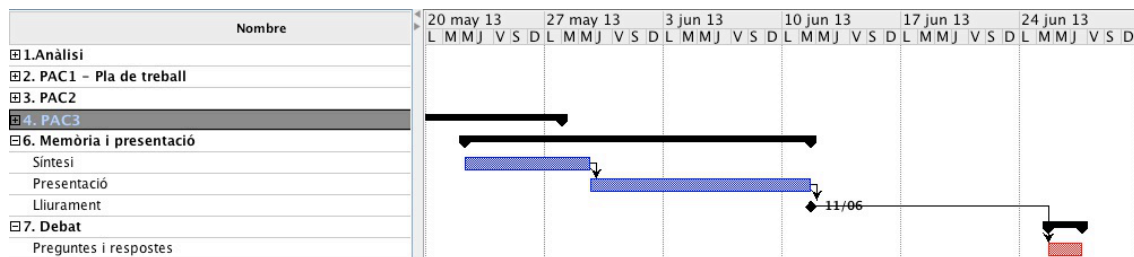
PAC2



PAC3 i Conclusions



Memòria, presentació i debat



1.5 Breu sumari de productes obtinguts

Un cop completat el TFC, obtindrem els següents lliuraments:

- Memòria
- Vídeo presentació TFC
- Disseny i implementació d'una ontologia

1.6 Breu descripció dels altres capítols de la memòria

En aquest apartat farem una breu descripció del contingut de la resta de capítols que formen aquest treball:

- Introducció dels elements i tecnologies que formen part de la Web Semàntica.

- Definició del concepte d'ontologia. Quina és la seva funció dintre de la Web Semàntica? Per què són tan importants?

Descripció del procés realitzat i de les característiques, per a la implementació de l' Ontologia entregada com a lliurament d'aquest TFC.

- Definició del concepte de SGDB en la web semàntica, què són? , quan i per què van aparèixer? quina és la seva finalitat? Hi trobarem un anàlisi de les seves característiques així com avantatges i desavantatges respecte a altres models. Finalment farem un recorregut per les aplicacions existents.

- Breu recorregut sobre els llenguatges de consulta més importants disponibles, amb una menció especial sobre SPARQL, llenguatge que emprarem majoritàriament per analitzar el rendiment de les dues aplicacions sota estudi.

- Estat de l'art dels SGDB analitzats en aquest treball (Virtuoso i Owlím-Lite), tenint en compte els següents paràmetres:

- Característiques tècniques
- Plataformes suportades
- Polítiques de llicència
- Arquitectura
- Emmagatzemat
- Consultes
- Instal·lació
- Rendiment

- Per completar l'anàlisi de Virtuoso i Owlím-Lite, executarem una sèrie de proves amb l'objectiu de determinar el rendiment de les dues aplicacions en diferents escenaris.

- Un cop enllestides les tasques sorgides de la planificació del TFC, analitzarem tota la informació generada així com l'aprenentatge sorgit de les proves pràctiques per a redactar les conclusions del projecte.

2. Web Semàntica

Per tal de poder definir correctament que són els sistemes gestors de bases de dades en el context de la web semàntica, introduïrem els conceptes de web semàntica i analitzarem les tecnologies sobre les quals es construeix.

Actualment, a la web podem trobar-hi milions de web sites, on cadascun d'ells conté pàgines web i documents que en un gran majoria, no són més que versions electròniques de documents en paper (factures, catàlegs, menús ...). Quan fem una cerca a google, obtenim una llista de web sites que poden contenir (o no) la informació que estem cercant, i per tant, es el propi usuari qui ha de validar la seva correcció. Això és degut a la tecnologia emprada per tal d'identificar la informació. Actualment els cercadors es basen en les paraules clau. Cerquen tots els documents on apareixen determinades paraules clau, processen tots aquests documents, els ordenen i els presenten a l'usuari. Aquest enfoc no és el més òptim, primer per què l'usuari no té la certesa que els resultats corresponguin a allò que ell ha cercat, i segon, per què aquest model no és escalable tenint en compte que tant la informació com el nombre d'usuaris a la web s'incrementaran de forma exponencial en els pròxims anys.

Per tal de solucionar tots aquests problemes, neix la web semàntica.

Segons Tim Berners-Lee “la Web Semàntica no és una web alternativa, sinó una extensió de l'actual, en la qual la informació té un significat ben definit, millorant el treball col·laboratori entre ordinadors i persones” ^[1].

Per tant, la web semàntica no és més que un revisió de l'actual estructura de la informació, basada en els següents principis ^[2]:

- La informació electrònica ha de ser inequívoca
- La informació s'ha de poder trobar
- La informació ha de ser reutilitzable
- La informació ha de ser inter operable (No s'ha de traduir la informació per passar d'un sistema a un altre)
- Els sistemes han de ser flexibles
- Temps real (Els processos i els productes generaran la informació per si mateixos, sense esperes ni processos offline)

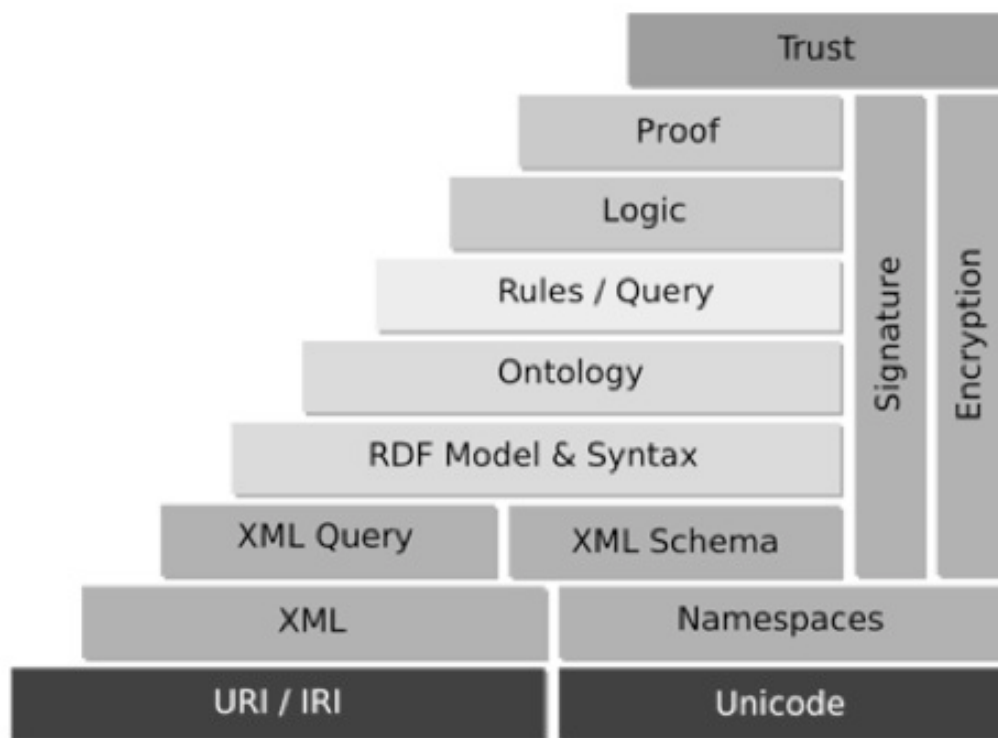


Figura 1. Arquitectura per capes de la web semàntica

Tot seguit descriurem, breument, els elements i les accions que es realitzen en cadascuna de les capes mostrades en la figura anterior.

- En les primeres capes podem trobar els elements bàsics que ens serviran de base per descriure els models i el llenguatge que ens permetran aplicar les sintaxis relacionades amb el RDF.

Una **URI**^[3] és simplement un identificador únic d'un recurs web. Cadascun dels objectes i models que podem trobar a la Web Semàntica ha de tenir per força un URI únic que l'identifiqui.

Unicode^[4], és un estàndard que ens permet proporcionar el mitjà pel qual un text, en qualsevol forma o idioma, pugui ser codificat per l'ús informàtic, amb la seguretat que les seves propietats originals es mantindran (no apareixeran caràcters o símbols estranys).

XML (Extensible Markup Language)^[5] no és més que un conjunt de regles per definir etiquetes semàntiques que ens ajuden a organitzar un document en diferents parts. Es tracta d'un meta-llenguatge que defineix la sintaxis utilitzada per definir altres llenguatges d'etiquetes estructurats.

El **XML Schema**^[6] és un llenguatge d'esquema desenvolupat per el World Wide Web Consortium (W3C), utilitzat per descriure l'estructura i les restriccions dels continguts dels documents XML d'una forma molt precisa, més enllà de les normes sintàctiques imposades pel propi XML. Aquesta precisió ens permet proporcionar una major potència expressiva que els DTD.

RDF (*Resource Description Framework*)^[7] és el model estàndard per representar tot tipus d'informació (metadades) present a la web, desenvolupat per el World Wide Web Consortium (W3C). Aquest model es basa en la traducció dels recursos de la Web a expressions amb la forma subjecte-predicat-objecte (estructura coneguda amb el nom de triplet), la qual ens permet barrejar, exposar i compartir informació estructurada o semi-estructurada entre diferents tipus d'aplicacions.

- En la següent capa trobarem les ontologies, que ens han de permetre descriure formalment el significat de la terminologia utilitzada en els documents web.

El **RDF Schema** (RDF's vocabulary description language)^[8] és una extensió semàntica del RDF desenvolupat per el World Wide Web Consortium (W3C) que té com a objectiu proporcionar els mecanismes necessaris per a descriure grups de recursos, les relacions entre aquests recursos així com les restriccions en les seves relacions.

L'**OWL** (Web Ontology Language)^[9] és un llenguatge de marcatge desenvolupat a partir del DAML+OIL i recomanat pel World Wide Web Consortium (W3C), que té com a objectiu facilitar un model de marcatge construït sobre RDF i codificat en XML, que ens ha de permetre descriure formalment el significat de la terminologia utilitzada en els documents web. L'OWL constitueix el primer nivell sobre el RDF requerit per la web semàntica.

- En la capa superior trobarem les regles i les consultes. Aquesta capa ens permetrà localitzar assertions sobre conjunts de dades inferibles, i per tant, obtenir informació semàntica correcta.

SPARQL (SPARQL Protocol and RDF Query Language)^[10] és un llenguatge estandarditzat per la consulta de grafs RDF, normalitzat pel RDF Data Access Working Group (DAWG) i el World Wide Web Consortium (W3C). Està basat en anteriors llenguatges de consulta per RDF, com ara dFDB, RDQA i SeRQL, aportant noves característiques molt importants. SPARQL és en aquests moments el llenguatge de consulta més potent per RDF/OWL.

- A la següent capa hi trobem la lògica i la demostració, que ens permetran efectuar els raonaments necessaris per a l'obtenció de les dades, aplicant les regles descrites en l'ontologia i comprovant la seva validesa.
- La darrera capa (anomenada confiança) ens ha de permetre donar confiança (qualificar) a les transaccions realitzades, tant a nivell usuari – programa com entre programes.

La capa de seguretat (Encryption i Signature), és una capa transversal que proporciona un suport específic per a garantir que una font d'informació és de confiança.

3. Ontologies

3.1 Introducció

Podem definir ontologia^[11] com a un vocabulari compartit que descriu un determinat domini i que es defineix en termes d'un llenguatge formal de forma que pugui ser manipulat de forma automàtica. Per tant, podem dir que, les ontologies proporcionen una comprensió compartida i consensuada del coneixement d'un domini que pot ser comunicada entre persones i sistemes heterogenis.

Les ontologies es representen mitjançant classes, propietats, atributs de les classes, relacions entre les classes i restriccions, que poden produir-se tant a nivell d'atributs com en les propietats.

Les ontologies han de permetre que els usuaris organitzin la informació de manera que els agents de software puguin interpretar-ne el significat, i per tant, puguin cercar i integrar les dades d'una forma més eficient que l' actual. Gràcies al coneixement emmagatzemat en les ontologies, les aplicacions podran extreure, de forma automàtica, dades de les pàgines web, processar-les, extreure'n conclusions, i finalment prendre decisions.

3.2 Disseny i implementació

Un dels objectius del TFC és dissenyar i implementar una ontologia per tal de poder posar en pràctica alguns dels coneixements adquirits sobre web semàntica.

A l'hora de dissenyar la nostra ontologia, hem seguit la metodologia proposada per la Universitat de Stanford^[12], que consisteix en les següents fases:

- Domini i abast de l' ontologia
- Considerar reutilitzar ontologies existents
- Enumerar els termes importants de l' ontologia
- Definir les classes i jerarquia
- Definir les propietats de les classes
- Definir les característiques de les propietats
- Crear les instàncies

L'aplicació utilitzada per a realitzar tant el disseny com la instanciació ha estat Protégé^[13] 3.4.8.

3.2.1 Domini i abast de l' ontologia

Com a domini de la nostra ontologia hem seleccionat avions.

Tenint en compte el temps limitat per a la realització d'aquesta ontologia l'abast de la mateixa ha estat acotat voluntàriament, per tal de cobrir, únicament, les característiques més generals del domini seleccionat.

Hem deixat fora de l'abast elements i classificacions com ara:

- Tipus d'estela
- Ales giratòries
- Aeròstats
- Aerodines

L' ontologia ha de servir per a organitzar de forma eficient la informació relativa al món dels avions, així com les relacions establertes entre elles, i ha de permetre respondre a les preguntes més freqüents que els usuaris i sistemes puguin formular-se.

3.2.2 Reutilització d'ontologies existents

A l'hora de realitzar el disseny i implementació d'aquesta ontologia no hem tingut en compte la possibilitat de reutilitzar alguna ontologia ja existent. Entenim que l'objectiu d'aquest treball era crear-ne una des de zero.

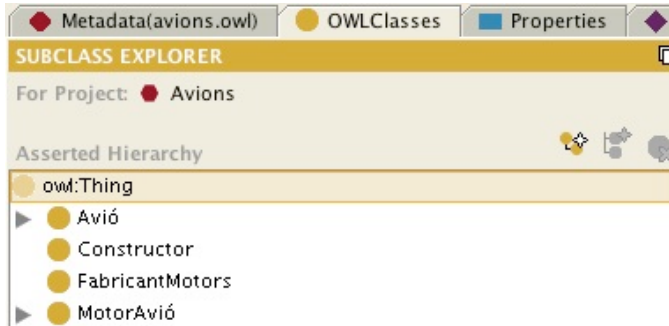
3.2.3 Enumeració dels termes importants de l' ontologia

L'objectiu és generar una llista amb tots els termes amb els que podríem fer afirmacions sobre un domini o que serien utilitzats per explicar aquest domini als usuaris. En el nostre cas, aquests serien alguns dels termes a enumerar:

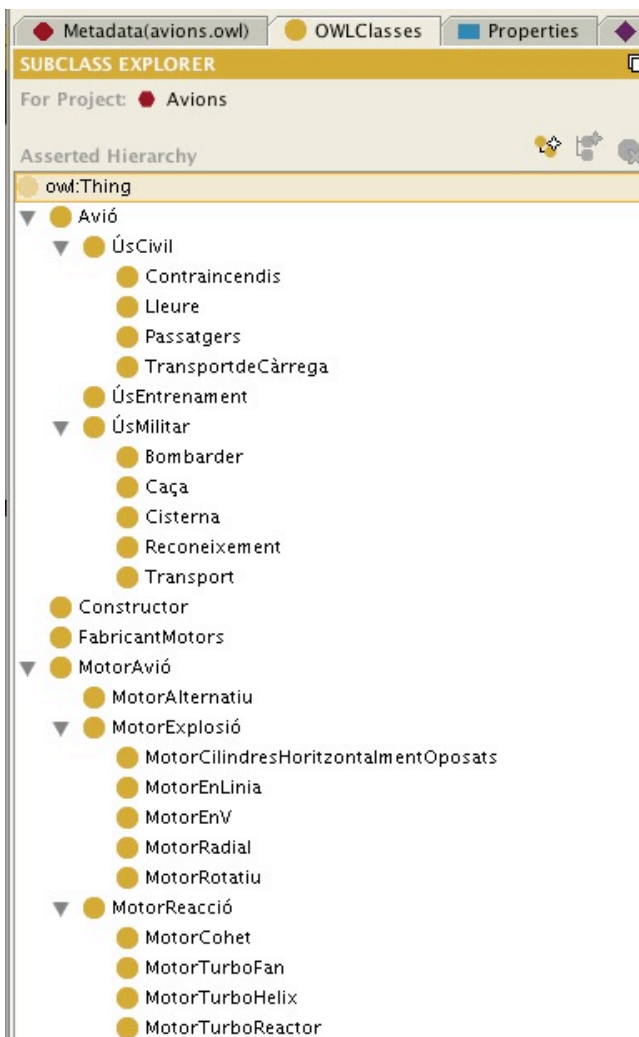
- Avions, motors, fabricants, constructors, turboreactors, motors d'explosió, hidroavions, caça, transport, passatgers ...

3.2.4 Definició de classes i jerarquia

Un cop analitzada la llista de termes de l'apartat anterior, hem seleccionat els termes que considerem poden ser independents per tal de constituir les classes.

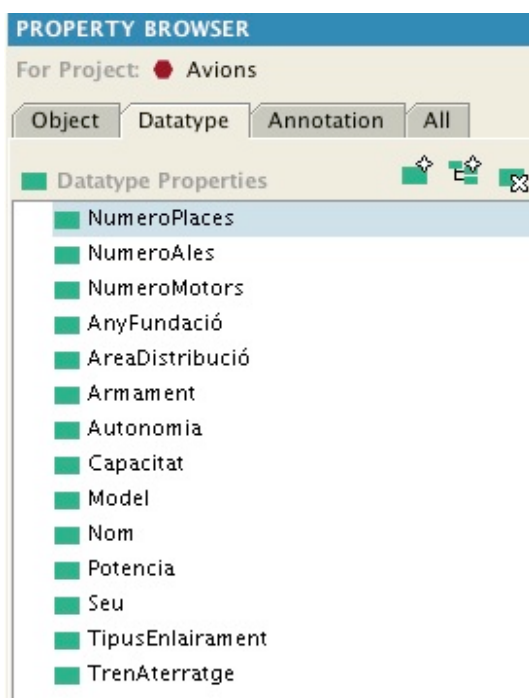


Un cop identificades les classes hem definit la jerarquia de l' ontologia. El resultat ha estat el següent:



3.2.5 Definició de les propietats

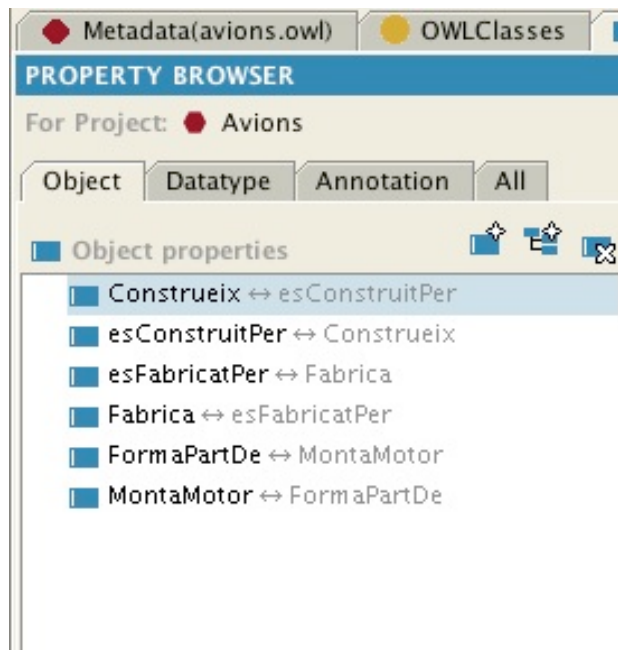
Hem definit les següents propietats (intrínseca):



Les propietats no tenen per què pertànyer a tots els dominis. En la següent taula defineixo a quins dominis pertanyen cadascuna de les propietats definides anteriorment.

Propietat	Domini
Nombre de places	Passatgers
Nombre d'ales	Avió
Nombre de motors	Avió
Any de fundació	Fabricant de motors
Àrea de distribució	Fabricant de motors
Armament	Bombarder, Caça
Autonomia	Avió
Capacitat	Passatgers, Transport de càrrega, Transport
Model	Avió, Motor avió
Nom	Constructor, Fabricant de motors
Potencia	Motor avió
Seu	Constructor, Fabricant de motors
Tipus d'enlairament	Avió
Tren d'aterratge	Avió

Com a part de la definició de les propietats, també hem inclòs les relacions entre els diferents objectes. Hem establert les següents relacions:



Tot seguit descriu les característiques de les relacions implementades:

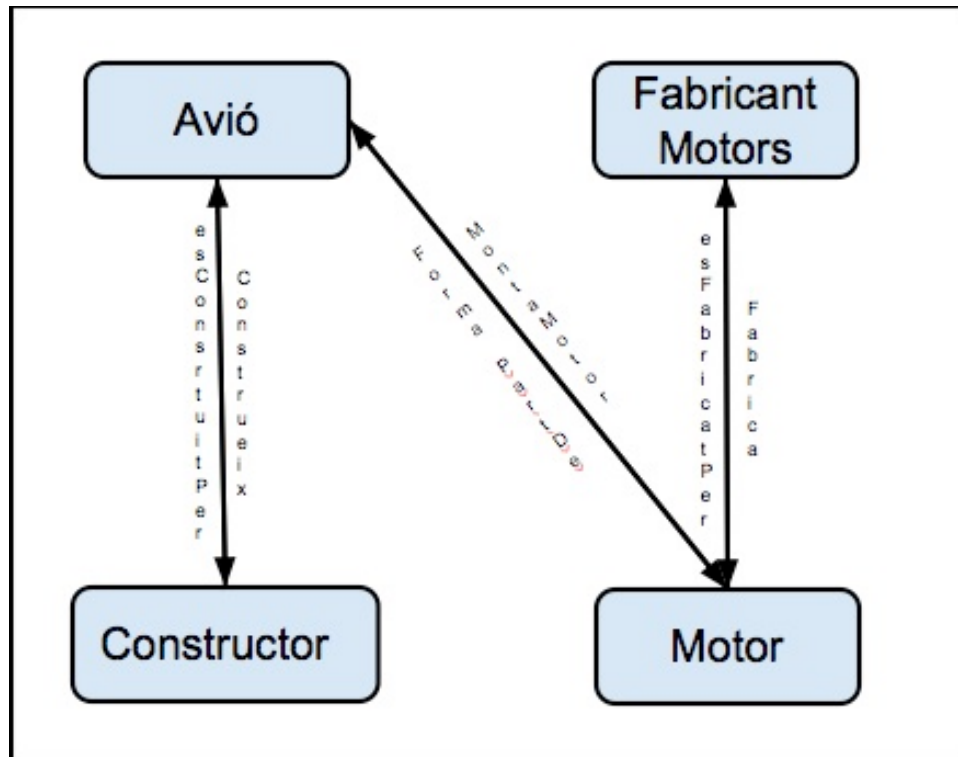


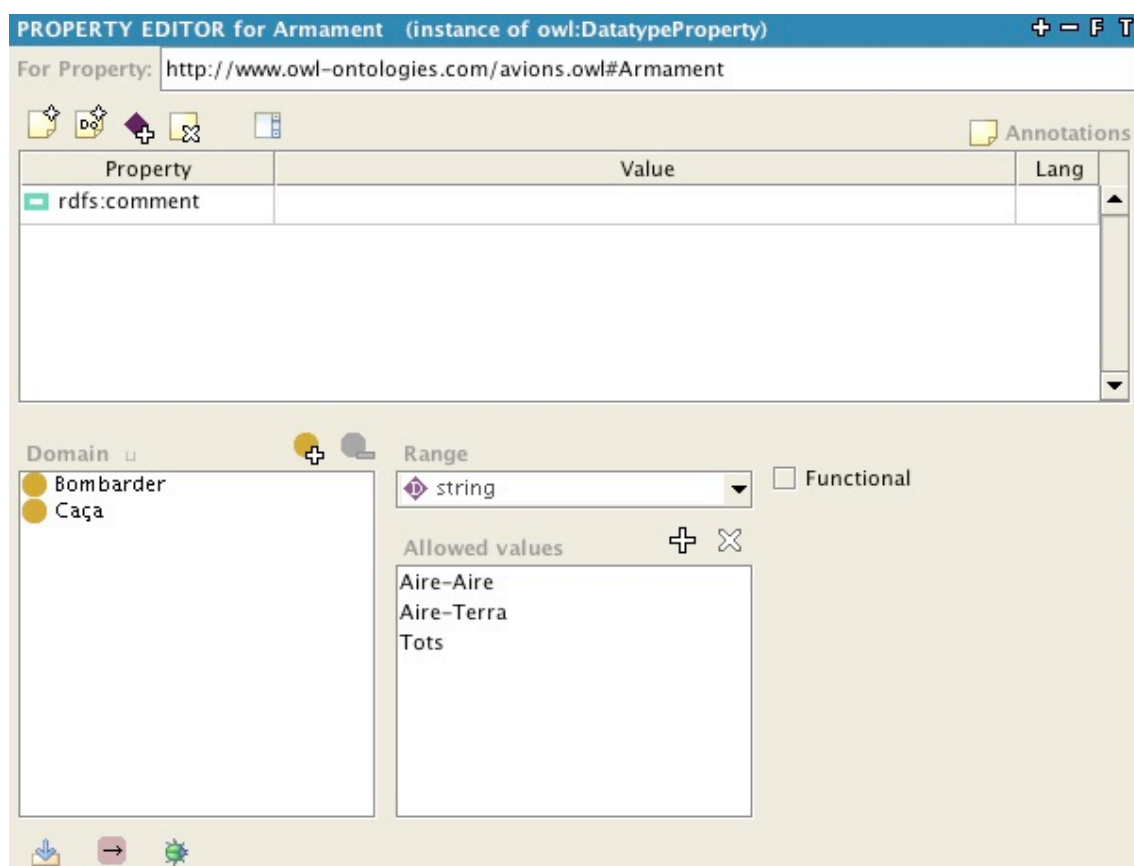
Figura 2. Relació entre les classes i els object properties

Propietat/Relació	Domini	Rang	Inversa
Construeix	Constructor	Avió	esConstruitPer
esConstruitPer	Avió	Constructor	Construeix
esFabricatPer	Motor avió	Fabricant motors	Fabrica
Fabrica	Fabricant motors	Motor avió	esFabricatPer
FormaPartDe	Motor avió	Avió	MontaMotor
MontaMotor	Avió	Motor avió	FormaPartDe

3.2.6 Definició de les característiques de les propietats

Les propietats tenen diferents característiques que serveixen per definir el tipus de valor acceptat, els valors acceptats, la cardinalitat ...

Tot seguit definiré aquestes característiques per a cadascuna de les propietats definides. Com a exemple, he inclòs una captura de pantalla de les característiques d'una de les propietats definides a Protégé. Les característiques de la resta de propietats les he inclòs en una taula.



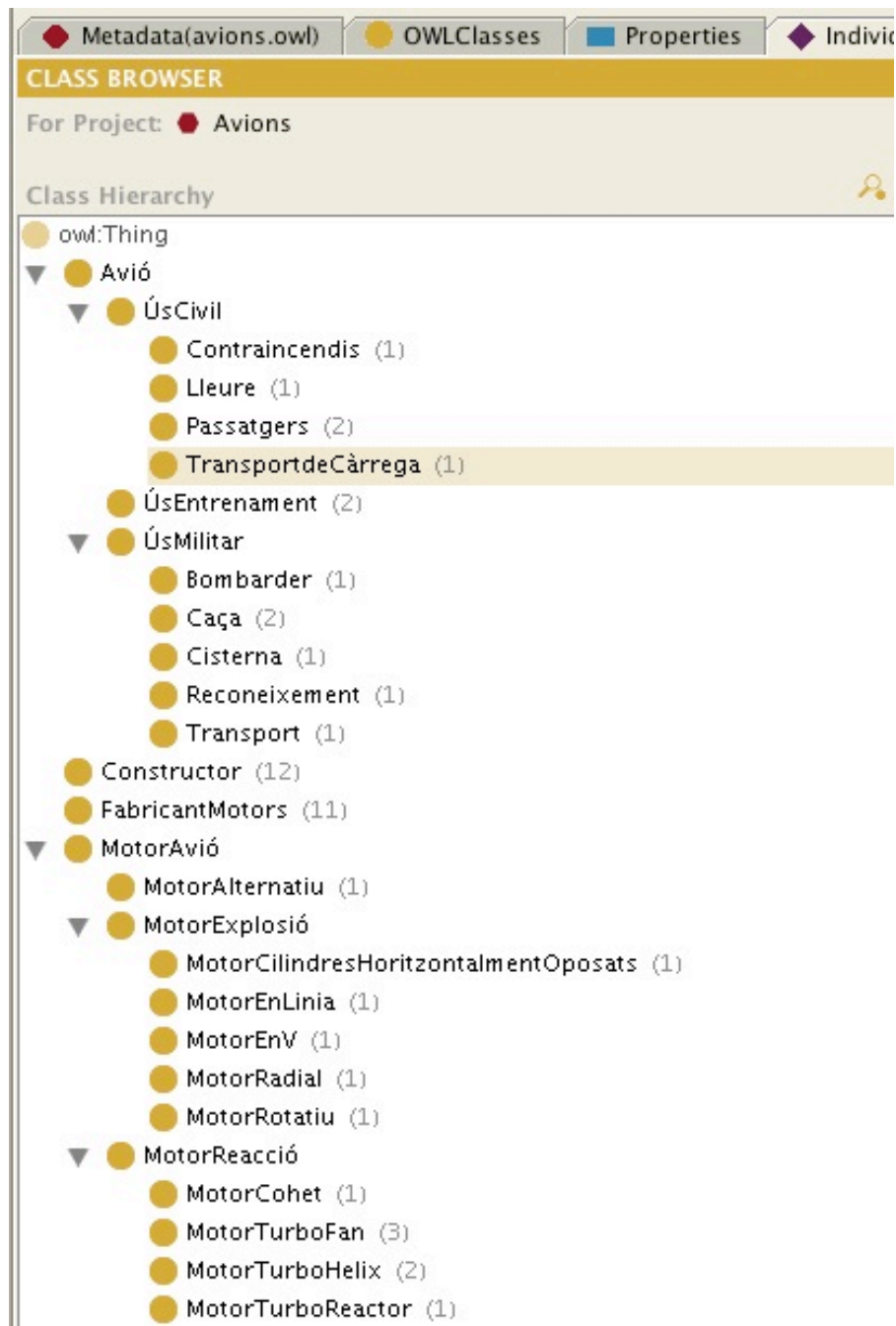
Propietat	Rang	Valors acceptats
Nombre de places	Enters	
Nombre d'ales	Enters	
Nombre de motors	Enters	
Any de fundació	Enters	
Àrea de distribució	Cadena de text	Europa Amèrica Àfrica Oceania Àsia Tot el món
Armament	Cadena de text	Aire-Aire Aire-Terra Tots
Autonomia	Cadena de text	Curta Mitja Gran
Capacitat	Cadena de text	Petita Mitja Gran
Model	Cadena de text	
Nom	Cadena de text	
Potència	Cadena de text	
Seu	Cadena de text	
Tipus d'enlairament	Cadena de text	Vertical Horitzontal
Tren d'aterratge	Cadena de text	Fix Retràctil Flotadors

3.2.7 Creació de les instàncies

La creació d'una instància és l'últim pas del procés. Per tal de poder definir una instància necessitem:

- Escollir una classe
- Crear una instància individual de la classe
- Omplir les propietats amb valors correctes

En la nostra ontologia hem creat una instància per a cada tipus de classe:



Tot seguit mostrem el procés realitzat a Protegé per la creació d'una instància:

- 1) Seleccionem una classe i creem una instància nova
- 2) Indiquem un nom per a la instància
- 3) Omplim els valors de les propietats simples.

INDIVIDUAL EDITOR for Antonov_An-225 (instance of TransportdeCàrrega) + - F T

For Individual: http://www.owl-ontologies.com/avions.owl#Antonov_An-225

Annotations

Property	Value	Lang
rdfs:comment		

Autonomia

Value	Type
Gran	string

NumeroAles

Value	Type
2	int

TrenAterratge

Value	Type
Retràtil	string

Capacitat

Value	Type
Gran	string

NumeroMotors

Value	Type
6	int

esConstruitPer

◆ Antonov

Model

Value	Lang
An-225	es

TipusEnlairament

Value	Type
Vertical	string

MontaMotor

◆ MotorSich_Progress_D-18T

4) Omplim els valors de les relacions (per exemple, munta motor):

INDIVIDUAL EDITOR for Antonov_An-225 (instance of TransportdeCàrrega)

For Individual: http://www.owl-ontologies.com/avions.owl#Antonov_An-225

Annotations

Property	Value	Lang
rdfs:comment		

Autonomia

Value	Type
Gran	string

NumeroAles

Value	Type
2	int

TrenAterratge

Value	Type
Retràctil	string

Capacitat

Value	Type
Gran	string

NumeroMotors

Value	Type
6	int

esConstruitPer

Antonov

Model

Value	Lang
An-225	es

TipusEnlairament

MontaMotor

Select Resources..

Allowed Classes

- MotorÀvió
 - MotorAlternatiu (1)
 - MotorExplosió
 - MotorCilindresHoritzontalmentOposats (1)
 - MotorEnLinia (1)
 - MotorEnV (1)
 - MotorRadial (1)
 - MotorRotatiu (1)
- MotorReacció
 - MotorCohet (1)
 - MotorTurboFan (3)
 - MotorTurboHelix (2)
 - MotorTurboReactor (1)

Direct Asserted Instances

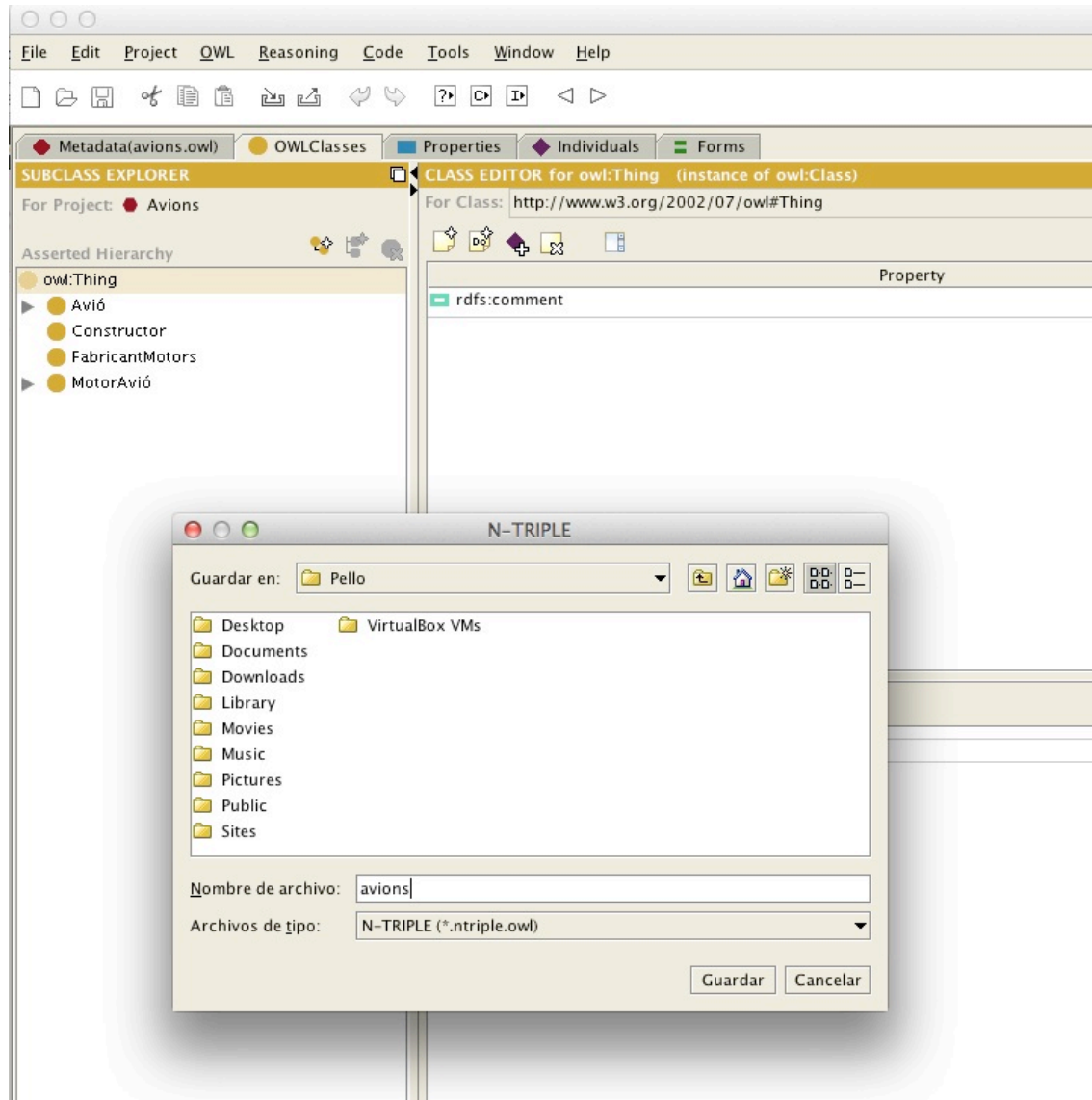
- Kuznetsov_NK-12MA
- Pratt_And_Whitney_PW123AF

OK Cancel

Hem de tenir en compte que, per completar el procés de creació d'una instància, hauríem de crear una altra instància per tal de poder completar el valor d'una propietat de relació. En l'exemple anterior, la relació munta motor fa referència a instàncies de la classe motor avió.

3.2.8 Transformació de l' ontologia a format RDF

Per tal de poder treballar l'ontologia amb les aplicacions Virtuoso i Owlim-Lite haurem de transformar-la al format RDF. Per fer-ho, haurem de seleccionar la opció arxiu -> exportar a format -> Triplet RDF del menú principal de Protégé.



Només ens quedarà indicar el nom i el camí on volem guardar l' ontologia.

4. SGDB utilitzats en el context de la web semàntica

4.1 Què és un SGDB?

Els SGDB^[14] (de l'anglès database management system) són un tipus de software dedicats a servir d'interfície entre la base de dades, l'usuari i les aplicacions que l'utilitzen que té com a propòsit general manejar de forma clara, senzilla i ordenada un conjunt de dades que posteriorment es convertiran en informació rellevant.

Una Base de dades és un conjunt de dades que pertanyen a un mateix context i s'emmagatzemen de forma sistemàtica per al seu posterior ús.

Un SGDB ha de complir els següents objectius:

- Abstracció d' informació
- Independència
- Consistència
- Seguretat
- Manegar les transaccions
- Temps de resposta

Com a tot sistema, els SGDB tenen avantatges i inconvenients. Tot seguit, descriurem breument els més importants.

Avantatges

- Permeten manipular grans volums de dades
- Simplifiquen la programació dels equips de consistència
- Garanteixen la correcció de les dades (sempre i quan s'apliquin polítiques adequades)
- Organitzen les dades de forma que es redueix l' impacte en el codi dels programes
- Disminueixen els temps de desenvolupament i augmenten la qualitat del sistema desenvolupat
- Proveeixen interfícies i llenguatges de consulta que simplifiquen la recuperació de dades

Inconvenients

- El manteniment es costós
- Complexitat
- Mida
- Cost econòmic

4.2 SGDB Web Semàntica

L'origen del model semàntic data dels anys 70^{[15][16]}. Inicialment va ser introduït com a eina de disseny, degut a la seva exactitud. El model semàntic no és més que una representació del món real.

No ha estat, però, fins fa pocs anys que hem pogut veure eines comercials que l'implementin.

El model semàntic redueix la complexitat lògica representativa de les bases de dades, la qual cosa la fa més usable pels usuaris, ja que aquests no necessiten saber-ne l'estructura per poder explotar-la satisfactòriament.

L'abstracció de dades i l'aïllament dels components lògics^[17] són dos conceptes molt importants per a les bases de dades semàntiques. Juntament amb l'adaptació de dades derivades conforme tres de les característiques fonamentals d'aquests sistemes.

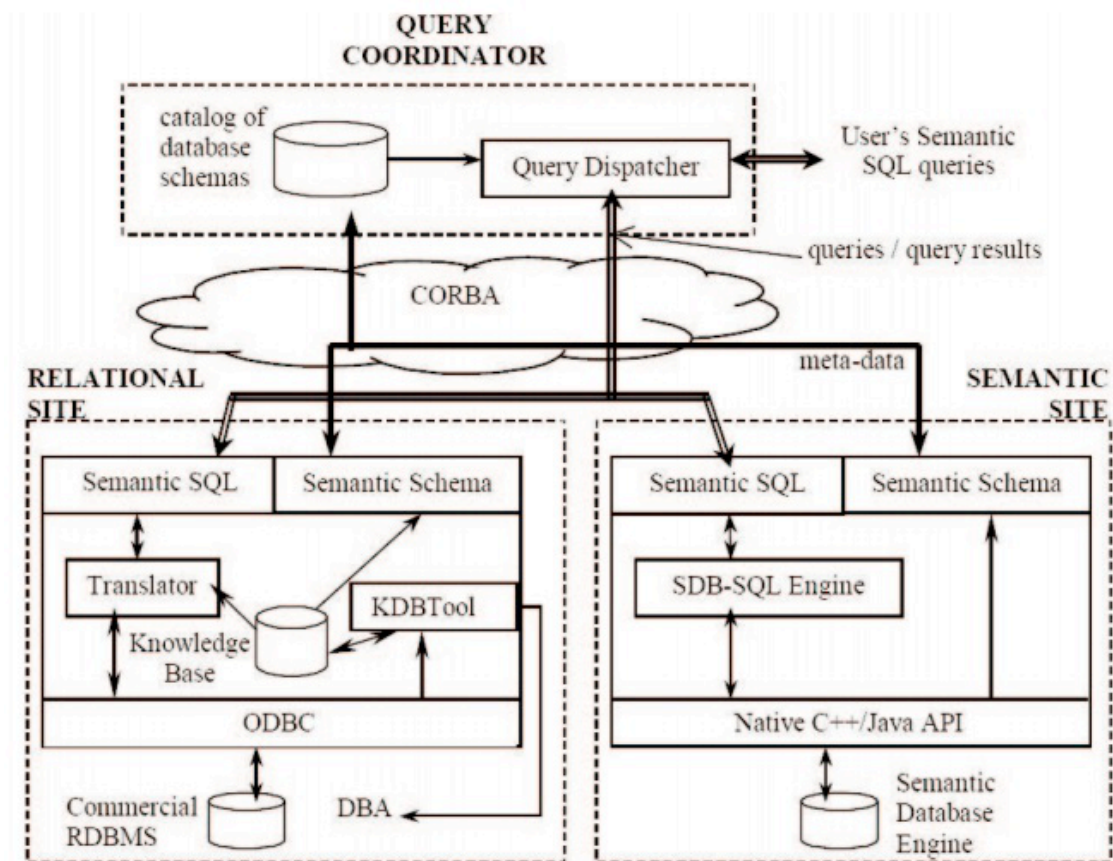


Figure 1. Overall architecture of SemanticAccess

Figura 3. Arquitectura web semàntica

La majoria de SGDB representen la informació com a simples registres. Les bases de dades semàntiques emmagatzemen la informació en forma de triplets RDF, i per tant, ens proporcionen estructures més complexes i riques que ens permeten construir mecanismes per a representar no només les dades, sinó també informació sobre els atributs i les relacions entre aquestes dades.

4.3 Avantatges

Tal i com hem descrit anteriorment, els models de dades semàntics són més complexes que els models de dades relacionals^[18] i, emfatitzen el modelatge de les relacions entre les dades. Aquest enfocament ens proporciona una sèrie d'avantatges sobre la resta de models tradicionals.

Aquests avantatges els podem resumir en:

- Separació entre dades i aplicacions
- Reducció de la complexitat en les relacions
- Millora en la disponibilitat de mecanismes d'abstracció

Separació entre dades i aplicacions

En una base de dades relacional, qualsevol canvi en les relacions establertes entre les dades significa canviar l'estructura de dades i possiblement dur a terme una migració de les dades o inclús haver de fer una nova instal·lació. Aquests canvis seran necessaris si volem mantenir la integritat de la informació, i per tant, signifiquen temps i diners. Això es degut, bàsicament, a que les aplicacions estan íntimament lligades a les dades i a l'estructura de les seves relacions.

En el model semàntic, les relacions formen part de les dades. Afegir una nova relació significa, simplement, afegir una nova entrada al model existent. No tenim dades per a una aplicació específica, sinó que tenim aplicacions que entenen un determinat model de dades. Com a resultat, tenim sistemes molt més fàcils de mantenir i que s'adapten millor a qualsevol tipologia de canvi.

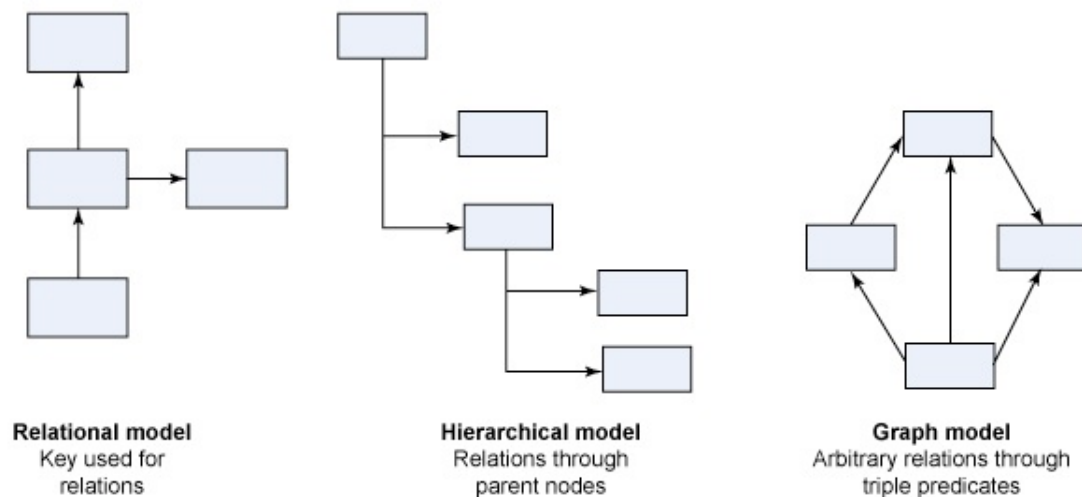


Figura 4. Diferències entre el model relacional, jeràrquic i semàntic

Reducció de la complexitat en les relacions

Els models orientats a registre ens proporcionen, únicament, dos o tres tipus de constructors per a representar les relacions entre dades. Aquesta, representa una diferència significativa respecte el model semàntic que ens proporciona moltíssimes més possibilitats a l'hora de representar les relacions entre dades.

Aquesta limitació, en el tipus de constructors disponibles en els models orientats a registre, provoca que les relacions entre dades estiguin semànticament sobrecarregades, que genera un alt grau de complexitat, i per tant, provoca ineficiència en els sistemes.

Aquesta ineficiència s'accentua en sistemes que maneguen un alt volum de dades. Per aquest motiu, avui en dia, creem sistemes que repliquen les dades ja existents de forma sumariada, reduint la complexitat en les seves relacions per tal de poder explotar-les de forma eficient.

Aquesta complexitat en les relacions es millora substancialment en el model semàntic. En aquest model, la informació s'emmagatzema en forma de triplets RDF, i la seva integració és tan senzilla com concatenar un graf darrera l'altre.

Millora en els mecanismes d'abstracció

Els models semàntics proporcionen uns mecanismes d'abstracció més rics que la resta de models de dades.

Aquests mecanismes d'abstracció ens permeten definir esquemes navegables i treballar amb esquemes derivats, la qual cosa facilita el coneixement de com estan organitzades les dades, i permet explotar-les d'una forma molt més eficient.

Els models semàntics proporcionen una rica varietat de mecanismes per visualitzar i accedir a l'esquema de dades amb diferents nivells d'abstracció. Un primer nivell d'abstracció, fa referència al nivell de detall amb què podem visualitzar una determinada part de l'esquema.

Un segon nivell d'abstracció, és el grau de modularitat que ens proporcionen. És fàcil aïllar informació basant-nos en el seu tipus, subtipus o pels seus atributs. Aquest nivell d'abstracció ens permet definir esquemes navegables i això ens permet determinar quines dades tenim i com estan modelades. Aquest tipus d'esquemes encara no han estat possibles en altres models

Un tercer nivell d'abstracció, ens proporciona la capacitat de crear nous esquemes a partir d'esquemes ja existents. L'esquema derivat permetrà als usuaris accedir-hi i operar-hi de forma transparent, ignorant els càlculs generats per obtenir-lo.

És cert que, aquesta característica, també la podem trobar en alguns models relacionals, no obstant, en el model relacional, el nou esquema derivat s'haurà de reflectir en forma de noves relacions o noves columnes.

4.4 SGDBs disponibles

Els SGDB de web semàntica són una realitat que cada vegada te més adeptes, es per això que tenim una gran varietat d'eines disponibles en el mercat. Tot seguit en farem un breu recorregut, classificant les eines segons la seva habilitat per treballar amb grans volums de dades (escalabilitat).

Grans magatzems de RDF ^[19]

Nom	Número de triplets emmagatzemats	Llicència	Comentaris
AllegroGraph	1+ trillions	Comercial	
Openlink Virtuoso	15.4+ bilions	Comercial	
Garlik 4Store	15 bilions	GPLv3	
BigOwl	12 bilions	Comercial	
BigData	12.7 bilions	Lliure	
YARS2	7 bilions	Comercial	
Jena TDB	1.7 bilions	BSD	
Jena SDB	650 milions	Lliure	
Mulgara	500 milions	Lliure	
RDF Gateway	262 milions	Comercial	
Jena with PostgreSQL	200 milions	Lliure	
Kowari	160 milions	Lliure	Sense manteniment
3Store with MySQL 3	100 milions	GNU	
Sesame	70 milions	Lliure	

Magatzems de RDF^[20]

Nom	Comentaris
Aduna Metadata Server	Basat en el servidor Sesame
ARC's RDF store	Programat en PHP, suporta SPARQL i té llicència de programari lliure
Boca	Programat en Java, forma part de la IBM Semantic Layered Research Platform
disKover	Producte comercial
D2RQ	Llibreria Java que proporciona accés a la BBDD relacionals a través de SPARQL i les APIs de Jena i Sesame
Dojo.data	Mòdul javascript que inclou suport per emmagatzemar RDF
Oracle Spatial 11g	Mòdul que inclou suport per treballar amb RDF
Pubby	És un servidor RDF que afegeix una interfície a magatzems RDF ja existents
RDFStore	És un magatzem RDF amb suport per Perl, C and SPARQL
RDF Server	Servidor pel paquet PHP RAP, que permet manipular RDFs.
Semantic Server	Implementat per Semantic Soft, proporciona un processador visual de SPARQL
SemWeb	Sense manteniment
SDB	És una BBDD SPARQL per Jena
SWI-Prolog Semantic Web Server	Implementa SPARQL i SeRQL per sobre de SWI-Prolog Semantic Web Library

5. Llenguatges de consulta en magatzems de dades semàntiques

5.1 Llenguatges de consulta XML

Amb l'aparició de les bases de dades XML es fa necessària la disponibilitat de llenguatges que permetin la consulta i recuperació d'informació emmagatzemada amb aquest tipus d'estructura.

Aquest llenguatge ha de permetre combinar les característiques dels llenguatges tradicionals de consulta. Per tant, ha de permetre una selecció de dades de l'estructura del document, la cerca d'informació en text lliure i consultes mixtes de continguts i estructura. Pel que fa a la sortida generada, una consulta XML retornarà un document XML.

Els llenguatges de consulta més estesos i estandarditzats són:

- **XPath**^[21]: És el llenguatge més popular per navegar, seleccionar i filtrar contingut d'un document XML. Permet l'especificació d'expressions de ruta d'accés i s'utilitza com a constructor d'altres llenguatges de consulta (XSLT i XQuery).
- **XQuery**^[22]: Llenguatge de consulta de propòsit general pel XML estandarditzat per la W3C (2005). La seva principal funció és extreure informació d'un conjunt de dades organitzades en forma d'arbre d'ordre N d'etiquetes XML.

5.2 Llenguatges de consultes RDF

A nivell sintàctic, els documents RDF són a la vegada documents XML. Per tant, qualsevol document RDF podrà ser consultat mitjançant un llenguatge de consulta XML. No obstant, les relacions que no són advertides en l'estructura del XML són molt difícils de consultar. Per a evitar aquesta limitació, i basant-se en l'estructura dels documents RDF, com a conjunt de triplets, han aparegut varis llenguatges de consulta que busquen i recuperen els triplets, com ara:

- RQL
- RDQL
- SeRQL
- SPARQL
- ...

5.3 SPARQL

SPARQL^[10] (SPARQL Protocol and RDF Query Language) és un llenguatge estandarditzat per a la consulta de grafs RDF, normalitzat pel RDF Data Access Working Group (DAWG) i el World Wide Web Consortium (W3C). Està basat en anteriors llenguatges de consulta per RDF, com ara dRDB, RDQA i SeRQL, aportant noves característiques molt importants.

SPARQL pot ser utilitzat per a expressar consultes que permetin interrogar diverses fonts de dades, sempre que les dades s'emmagatzemin de forma nativa en format RDF o bé siguin definides mitjançant vistes RDF.

SPARQL té capacitat per a la consulta de patrons obligatoris i opcionals de graf, juntament amb les seves conjuncions i disjuncions. També suporta l'ampliació o restriccions de l'àmbit de les consultes indicant els grafs sobre els que s'opera. Els resultats de les consultes SPARQL poden ser resultats o grafs RDF.

És necessari distingir entre el llenguatge de consulta i el motor d'emmagatzemament i recuperació de dades. Per aquest motiu, existeixen múltiples implementacions de SPARQL.

SPARQL permet quatre tipus diferents de consultes, que són:

- SELECT que identifica les variables que apareixen en els resultats de la consulta.
- ASK, ens retorna un booleà indicant si el patró de cerca existeix o no.
- CONSTRUCT, ens retorna un graf RDF, construït substituint variables en un conjunt de triplets.
- DESCRIBE, en retorna un graf RDF que descriu el tipus de dades que formen part dels recursos trobats.

Mostrarem un exemple de cadascuna de les consultes més endavant en l'anàlisi d'Owlím-Lite i Virtuoso.

SPARQL és, en aquest moment, el llenguatge de consulta més potent per RDF/OWL.

6. Virtuoso

OpenLink Virtuoso^[23] és una plataforma híbrida dissenyada com a solució única (Universal server). El seu disseny, li permet combinar les següents funcionalitats en un únic sistema:

- Servidor Web
- Servidor Fitxers
- SGDB (SQL, XML, RDF, objectes)
- Free-text search engine

L'origen de Virtuoso el trobem en Kubl RDBMS, una base de dades lleugera d'alt rendiment que va arribar al mercat a finals del 1996. Uns anys més tard, OpenLink i Kubl van fusionar-se donant lloc a l'actual Virtuoso.

Virtuoso, s'ha convertit en un dels SGDB comercials més madurs del mercat, i passa per tenir a producció algunes de les bases de dades semàntiques més grans que existeixen actualment.

6.1 Característiques tècniques

Protocols implementats

- HTTP
- HTTPS
- WebDAV
- CalDAV
- CardDAV
- SOAP
- UDDI
- WSDL
- WS-Policy
- WS-Security
- WS-ReliableMessaging
- WS-Routing
- WS-Referral
- WS-Attachment
- WS-BPEL
- SyncML
- Gdata
- SPARQL
- SPARUL
- NNTP

API suportades

- ODBC
- JDBC
- OLE DB
- ADO.NET
- ADO.NET Entity Framework
- XMLS

Altres

- Atom
- RSS 2.0
- RSS 1.0
- OPML
- XBEL
- FOAF
- SIOC

6.2 Plataformes suportades

Virtuoso està disponible pels següents sistemes operatius:

- Windows
- Linux
- Unix (AIX, HP-UX, Solaris ...)
- Mac OS X

6.3 Política de llicències

Virtuoso és un producte comercial de OpenLink. La seva política de llicències es basa en el número de processadors (o cores) i el nombre de fils en paral·lel que permet executar. Basant-nos en aquesta política trobem diferents versions:

- Servidor únic / Equip de treball
- Site
- Empresa
- Cluster

També hi ha una versió preparada per a treballar amb instàncies d' Amazon EC2 (cloud).

L'any 2006 OpenLink va llançar una versió open source anomenada també OpenLink Virtuoso.

6.4 Arquitectura

L'arquitectura de Virtuoso està dividida en diferents components, que són:

- Motor de bases de dades virtual
- Web & Internet
- Web Services

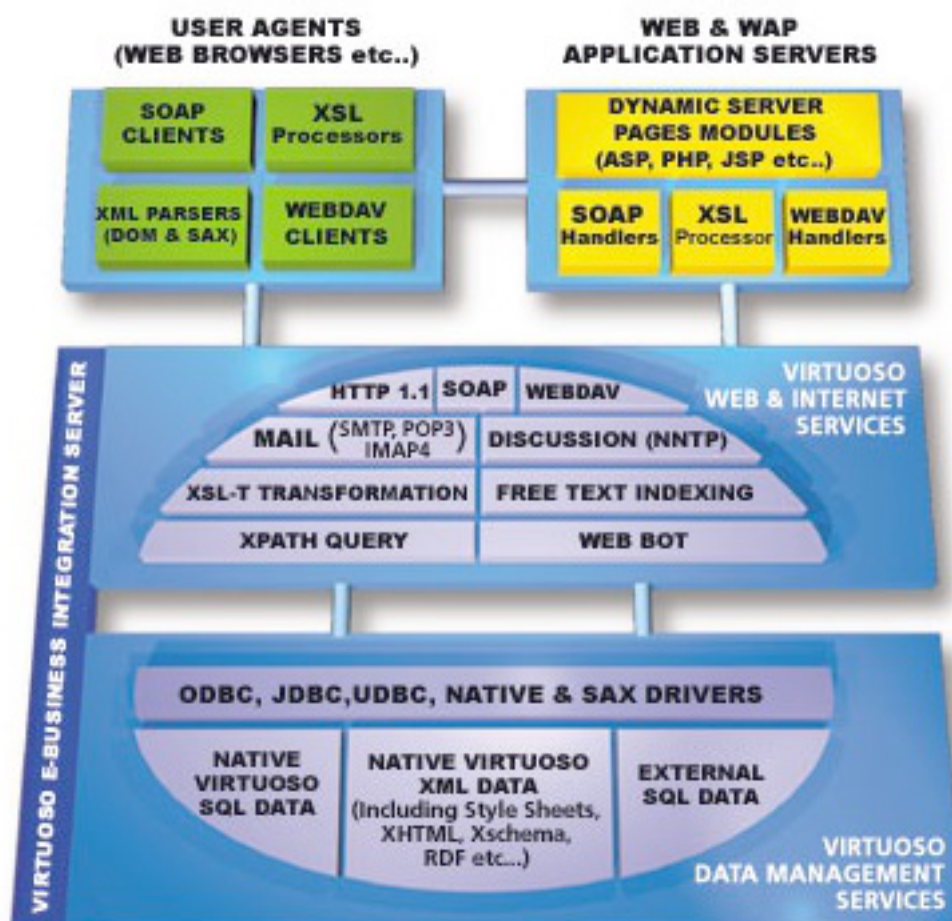


Figura 5. Arquitectura de Virtuoso

La base de Virtuoso és el seu motor de base de dades virtual. Aquest manega les tasques de la base de dades i les bases de dades virtuals, emprant tant recursos locals com remots (SQL i XML). El mòdul de Web i Internet es situa just damunt i proporciona les interfícies necessàries per a interactuar amb la web i els seus usuaris.

El motor de bases de dades virtuals està format pels següents components:

Data Access Drivers : Porta d'entrada als serveis del motor de bases de dades virtuals. Suporta els estàndards del mercat.

Security Manager : Responsable de mantenir les dades protegides i també de gestionar els privilegis dels usuaris de l'aplicació.

Query Manager : Proporciona anàlisi lèxic i sintàctic de les consultes, plans d'execució, compilació i servei de cerca.

Meta Data Manager : És l'encarregat de proporcionar al Query Manager informació sobre les entitats en les que treballarà el pla d'execució i també s'encarrega d'enllaçar les fonts de dades externes i dirigir-les al processador de consultes i als gestors d'entrada/sortida que pertoquin.

Transaction Manager : És l'encarregat de garantir l' atomicitat, consistència, aïllament i persistència de les transaccions.

Concurrency Manager : Gestiona que clients i serveis puguin obrir múltiples sessions al mateix temps, sense posar en compromís el rendiment i l'eficiència de les operacions realitzades.

Local Data I/O Manager : Proporciona l' emmagatzemament local i les operacions associades.

External Data I/O Manager : Gestiona les operacions d'escriptura i lectura de fonts de dades externes.

Replication Manager : Gestiona la migració, transformació i sincronització de les dades en un entorn Virtuoso distribuït.

6.5 Extensions

Virtuoso proporciona una sèrie de funcions i procediments emmagatzemats que poden ser invocats mitjançant estàndards com ara ODBC, JDBC, OLE-DB, ADO.NET, XMLA i d'altres que afavoreixen la integració amb d'altres sistemes. No obstant, no ofereix un API que faciliti aquesta integració de forma nativa. En la versió GPLv2 (no així en les versions comercials), és possible accedir al codi font de l'aplicació i, per tant, és possible implementar noves interfícies o millorar-ne les actuals facilitant així possibles integracions amb d'altres aplicacions.

6.6 Emmagatzemat

Degut a la seva arquitectura única i multiprocés, Virtuoso ofereix la possibilitat d'emmagatzemar les dades tant en el sistema de fitxers com en la seva base de dades. Les dades es guarden en forma de triplet, seguint el format estàndard genèric des-normalitzat.

Virtuoso suporta RDF, RDFS y OWL, però el nivell d'inferència és molt limitat. La càrrega de dades es pot realitzar en forma de sentències individuals, o bé, en format massiu a través de procediments emmagatzemats com ara el `RDF_LOAD_RDFXML`.

La serialització també està suportada (entrada i sortida) en RDF/XML i Turtle. Totes les operacions anteriorment descrites poden realitzar-se dintre de transaccions i permeten concurrència.

6.7 Càrrega de dades

La càrrega de dades es pot realitzar en forma de sentències individuals o bé en format massiu a través de procediments emmagatzemats com ara el `RDF_LOAD_RDFXML`.

La serialització, també està suportada (entrada i sortida) en RDF/XML i Turtle. Totes les operacions anteriorment descrites poden realitzar-se mitjançant transaccions i permeten concurrència.

Per tal de comprovar com funciona la càrrega massiva de dades, utilitzarem l'ontologia "Avions", creada durant el projecte, prèviament transformada com un conjunt de triplets RDF.

Per tal de carregar els triplets RDF a Virtuoso utilitzarem l'eina `isql` (línia de comandes). Primer de tot, afegirem el directori on tenim les dades a carregar com a directori vàlid en el fitxer de configuració de Virtuoso "virtuoso.ini".

Tot seguit, obrim `isql-vt` (és l'equivalent a `isql` si hem instal·lat el Virtuoso des de el gestor de paquets de ubuntu).

```
$ isql-vt
```


Indiquem el fitxer a carregar amb la següent comanda:

```
$ ld_dir ('/home/pello/Descargas','avions.nt','http://dbpedia.org');
```

```
pello@pello-vb-virtuosodb:~$ isql-vt
OpenLink Interactive SQL (Virtuoso), version 0.9849b.
Type HELP; for help and EXIT; to exit.
SQL> ld_dir ('/home/pello/Descargas/', 'avions.nt', 'http://dbpedia.org');

*** Error 28000: [Virtuoso Driver]CL034: Bad login
at line 1 of Top-Level:
ld_dir ('/home/pello/Descargas/', 'avions.nt', 'http://dbpedia.org')

Enter password for dba :
Connected to OpenLink Virtuoso
Driver: 06.01.3127 OpenLink Virtuoso ODBC Driver

Done. -- 47 msec.
SQL> █
```

Procedim a la càrrega de dades:

```
$ rdf_loader_run();
```

```
SQL> rdf_loader_run();

Done. -- 627 msec.
SQL> █
```

Tot seguit, validarem que les dades s'hagin carregat correctament. Per fer-ho, executarem una consulta que ens retorni el numero de triplets carregats en el sistema:

```
$ sparql select count(*) where { ?s ?p ?o };
```

```
SQL> sparql select count (*) where { ?s ?p ?o };
callret-0
INTEGER
-----
4837
1 Rows. -- 1 msec.
```

6.8 Consultes

Virtuoso implementa SPARQL i una extensió pròpia d'aquest anomenat SPARUL. No obstant, algunes funcionalitats de SPARQL encara no estan suportades:

- Caràcters unicode en els noms
- Comentaris dintre de les consultes

Com hem dit anteriorment Virtuoso ha implementat una extensió de SPARQL anomenada SPARUL que millora certes funcionalitats i n'afegeix algunes noves.

El motor de virtuoso també ofereix suport per a les cerques Free-Text així com consultes XML.

En la secció, proves de rendiment, mostrarem un exemple per a cadascun dels següents tipus de consulta sota Virtuoso:

- SELECT
- ASK
- DESCRIBE
- CONSTRUCT

L'execució de les consultes les realitzarem a través de la interfície gràfica Conductor.

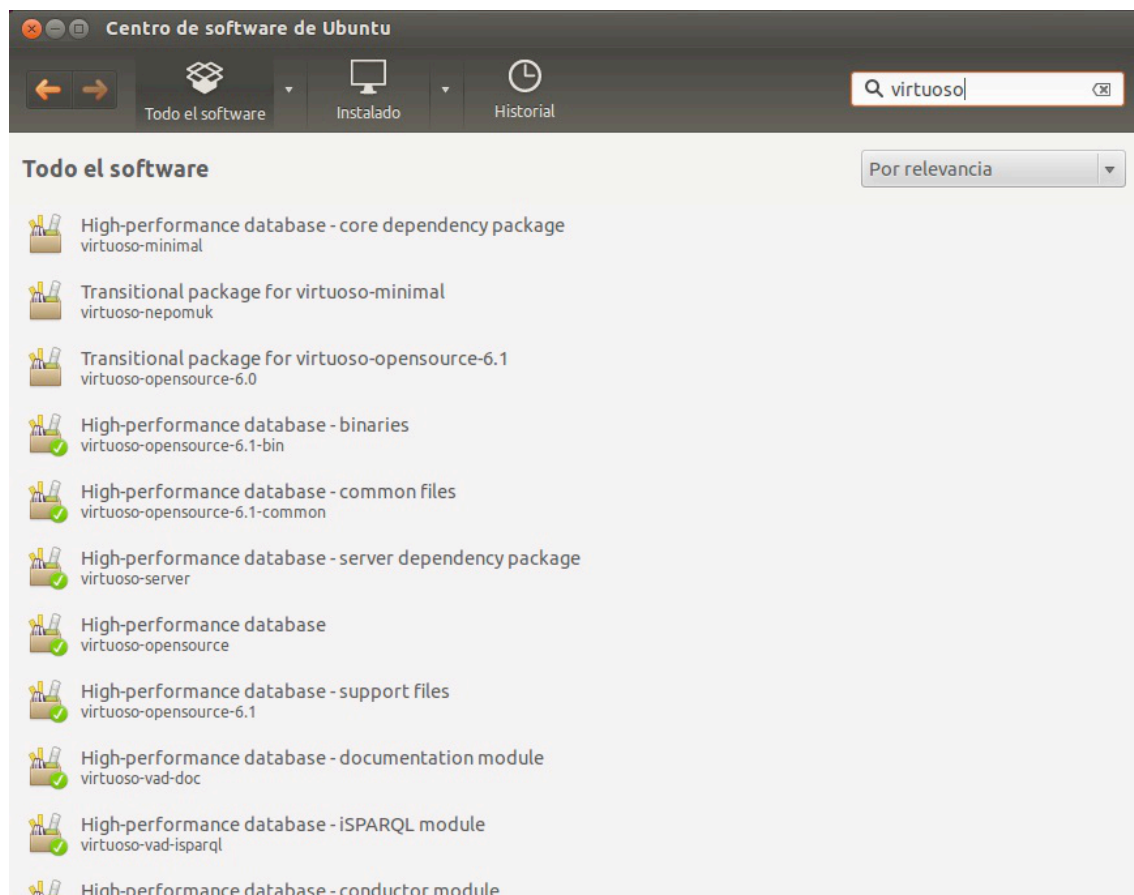
6.9 Instal·lació

OpenLink ens ofereix dos possibles modes d'instal·lar l'aplicació sota Ubuntu:

- Mitjançant Ubuntu Packages
- Baixant el codi font i compilant-lo

Hem decidit optar per la primera opció. Tot seguit, farem un breu resum dels passos necessaris per a dur a terme la instal·lació en el nostre equip.

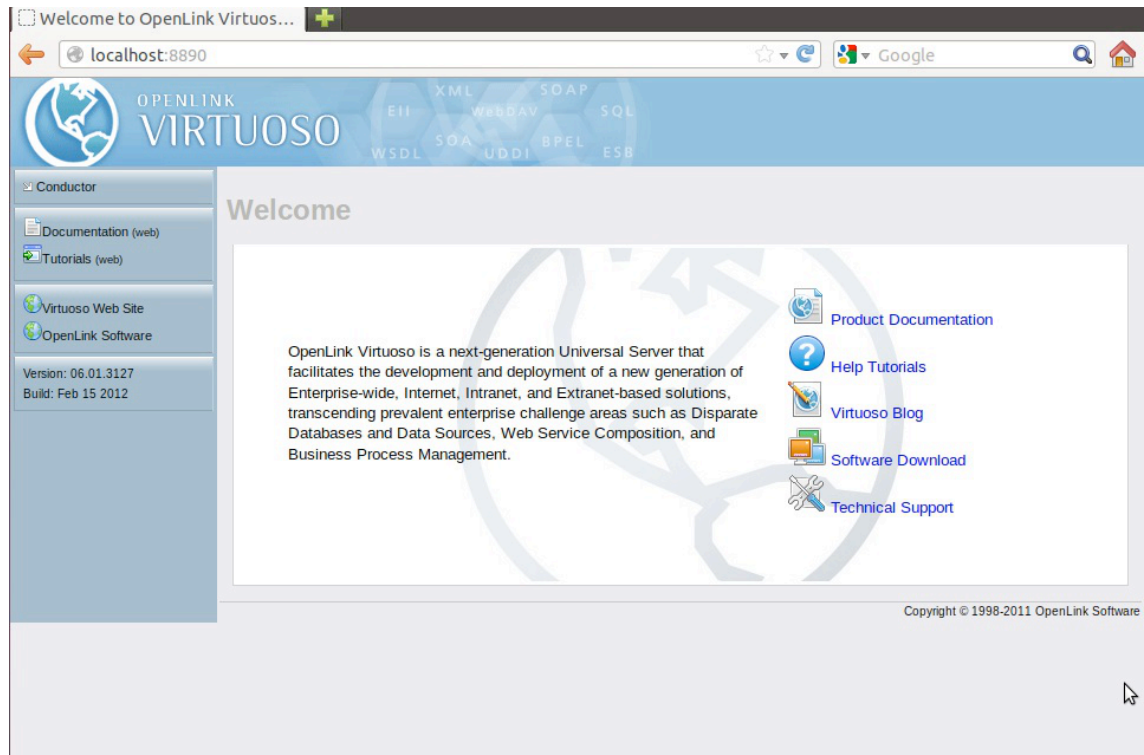
La instal·lació per paquets pot realitzar-se a través de la línia de comandes, o bé, mitjançant el centre de software. La segona opció es molt intuïtiva i fàcil de seguir, només hem de fer una cerca per “Virtuoso” en el centre de software i ens apareixeran tots els paquets disponibles.



Seleccionem els paquets que volem instal·lar, i fem clic a instal·lar. Durant el procés d'instal·lació ens demanarà definir una contrasenya d'administració per la Base de dades i pel sistema de fitxers Dav.

Un cop el procés d'instal·lació ha acabat, Virtuoso ja està en funcionament. Per a poder veure la seva consola d'administració només hem d'obrir un navegador i carregar la següent URL:

<http://localhost:8890/>



7. Oqlim-Lite

Oqlim-Lite^[24] és la base de dades dissenyada per Ontotext per emmagatzemar i consultar dades estructurades en format RDF. Es tracta d'un producte comercial, està implementat en Java i es distribueix com a mitjà d'emmagatzemat i capa d'inferència pel framework Sesame OpenRDF.

Aquestes són algunes de les seves característiques més importants:

- Segons els seus desenvolupadors Oqlim-Lite és el magatzem OWL més ràpid del mercat.
- Implementat totalment en Java, la qual cosa en facilita el desenvolupament així com la seva portabilitat.
- Compatible amb Sesame 2 i, per tant, amb la majoria de sintaxis i llenguatges de consulta RDF.

El nom d'Oqlim ve de "OWL in Memory". Els seus creadors (Ontotext) són una empresa Búlgara creada a l'any 2000, especialista en desenvolupar solucions propietàries per la web semàntica.

7.1 Característiques tècniques

Degut a l'estreta relació amb Sesame, Oqlim-Lite suporta tots els estàndards de consulta i tipus de dades suportats per Sesame.

Llenguatges de consulta:

- SeRQL
- SPARQL
- RQL
- RDQL

Sintaxis suportades:

- XML
- N3
- N-Triples
- N-Quads
- Turtle
- TriG
- Trix

Semàntiques:

- RDFS
- OWL Horst
- OWL Max
- OWL2 QL
 - OWL2 RL
 - Custom Rules

7.2 Plataformes suportades

Owlim-Lite està disponible per a qualsevol plataforma que suporti Java 1.6 o posterior.

7.3 Política de llicències

Owlim-Lite és un producte comercial d' Ontotext. Owlim-Lite es troba disponible en tres versions diferents (amb polítiques de llicències diferents):

- Owlim-Lite. Dissenyada per a volums de càrrega mitjos (per sota de 100 milions de registres). Es distribueix sota llicència LGPL sense cost per ús.
- Owlim-SE. Dissenyada per a grans volums de càrrega. Es distribueix sota llicència comercial.
- Owlim-Enterprise. Dissenyada per a un alt rendiment, permet consultes en paral·lel i grans volums de càrrega. Es distribueix sota llicència comercial.

També hi ha una versió preparada per treballar amb instàncies de Amazon EC2 (cloud).

7.4 Arquitectura

L'arquitectura d'Owlim-Lite es basa en dos grans components:

- Owlim-Lite SAIL. Interfície que permet la integració amb Sesame
- Motor TRREE (Triple Reasoning and Rule Entailment Engine). Encarregat de l'emmagatzemat i consulta de les dades

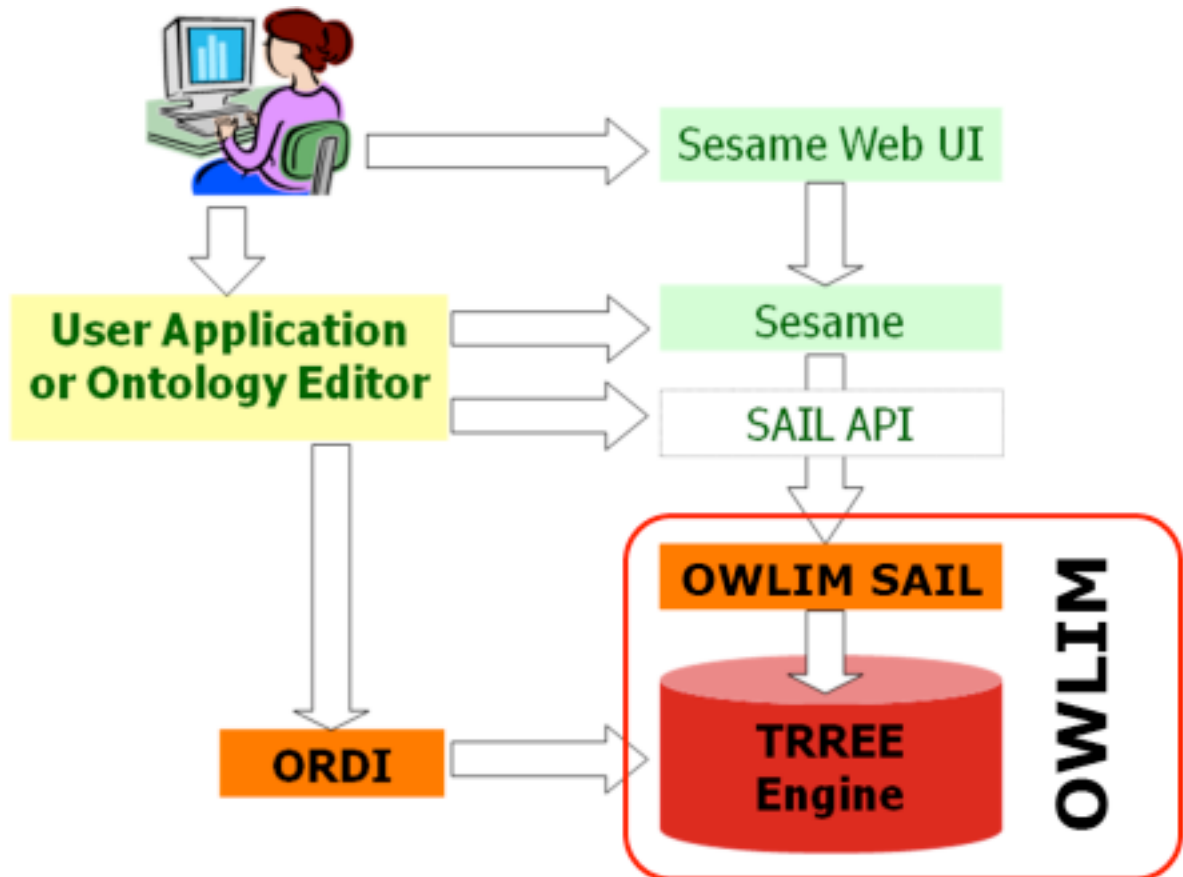


Figura 6. Arquitectura d'Owlim-Lite

Com hem mencionat anteriorment Owlim-Lite es distribueix com a mitjà d'emmagatzemat i capa d'inferència pel framework Sesame. La integració entre Owlim-Lite i Sesame es realitza a través de la interfície Sesame SAIL que permet la comunicació amb la resta de components del framework Sesame.

També podem treballar amb un magatzem Owlim-Lite a través d'interfícies funcionals d'alt nivell (obviant Sesame), però Sesame és la forma més senzilla d'integració.

Owlim-Lite es construeix sobre el motor TRREE (Triple Reasoning and Rule Entailment Engine). Aquest motor és capaç de realitzar raonaments basats en

encadenaments progressius de regles de vinculació sobre triplets RDF amb variables.

Ontotext té disponibles diferents versions d'aquest motor. La diferència principal entre ells és la capacitat de processament, inferència i indexació. La versió d' Owlím-Lite que estem avaluant (Owlím-Lite), empra el motor anomenat "SwiftTRREE", que té com a característica principal que els processos de consulta i raonament s'avaluen en memòria.

7.5 Extensions

Owlím-Lite implementa Sesame SAIL, el que el permet, no únicament comunicar-se amb Sesame, sinó treure avantatge de totes les seves funcionalitats i possibilitats d'extensió.

7.6 Emmagatzemat

El sistema de emmagatzemat d'Owlím-Lite es basa en fitxers. Aquests fitxers s'emmagatzemen en format binari i, permeten carregar a memòria de forma ràpida totes les dades un cop arrencat Owlím-Lite. Donat que Owlím-Lite carrega les dades a memòria, la sincronització entre el contingut de memòria i el fitxer d'origen només es realitza a l'arrencar i aturar l'aplicació.

La càrrega de dades es pot realitzar en forma de sentències individuals o bé, en formatiu massiu, a través d'una URL o d'un fitxer.

Owlím-Lite (a través de Sesame), és capaç de llegir aquestes dades en els següents formats:

- TriX
- N-Triples
- N3
- RDF/XML
- BinaryRDF
- Turtle
- TriG

7.7 Càrrega de dades

La càrrega de dades es pot realitzar en forma de sentències individuals o bé en format massiu, ja sigui des d'una URL, un fitxer de dades o bé, com a un conjunt de triplets en qualsevol dels formats mencionats en l'apartat anterior.

Per tal de comprovar com funciona la càrrega massiva de dades, utilitzarem l'ontologia "Avions" creada durant el projecte, prèviament transformada com un conjunt de triplets RDF.

Per tal de carregar els triplets RDF a Owlim-Lite utilitzarem la interfície gràfica Sesame. Primer de tot, crearem un nou magatzem.

The screenshot shows the OpenRDF Workbench interface in a web browser. The browser's address bar displays the URL `localhost:8080/openrdf-workbench/repositories/NONE/create`. The page title is "OpenRDF Workbench". On the left, a sidebar menu contains sections: "Sesame server", "Repositories" (with sub-items "New repository" and "Delete repository"), "Explore" (with sub-items "Summary", "Namespaces", "Contexts", "Types", "Explore", "Query", and "Export"), "Modify" (with sub-items "SPARQL Update", "Add", "Remove", and "Clear"), and "System" (with sub-item "Information"). The main content area is titled "New Repository". It shows "Current Selections:" with "Sesame server: http://localhost:8080/openrdf-sesame" and "Repository: - none -". Below this, the form fields are: "Type:" set to "OWLIM-Lite", "ID:" set to "DSS", and "Title:" set to "Avions-TFC". There are "Next" and "Cancel" buttons at the bottom of the form. The footer of the page reads "Copyright © Aduna 1997-2011" and "Aduna - Semantic Power".

Tot seguit, carregarem les dades del nostre fitxer de triplets al magatzem que tot just acabem de crear.

OpenRDF Workbench - Add ... Gmail: Email from Google

localhost:8080/openrdf-workbench/repositories/DSS/add

Workbench OpenRDF

Sesame server

Repositories

New repository

Delete repository

Explore

Summary

Namespaces

Contexts

Types

Explore

Query

Export

Modify

SPARQL Update

Add

Remove

Clear

System

Information

Current Selections:

Sesame server: <http://localhost:8080/openrdf-sesame> [change]

Repository: Avions-TFC (DSS) [change]

Add RDF

Base URI:

☒ use base URI as context identifier

Context:

Data format:

☐ Location of the RDF data you wish to upload

RDF Data URL:

☒ Select the file containing the RDF data you wish to upload

RDF Data File:

☐ Enter the RDF data you wish to upload

RDF Content:

Copyright © Aduna 1997-2011
Aduna - Semantic Power

Posteriorment, validarem que les dades s'hagin carregat correctament. Per fer-ho, executarem una consulta que ens retorni el numero de triplets carregats en el sistema:

```
select (count(*) as ?count) where { ?s ?p ?o };
```

OpenRDF Workbench - Que... Gmail: Email from Google

localhost:8080/openrdf-workbench/repositories/DS5/query

Workbench

OpenRDF

Sesame server

Repositories

- New repository
- Delete repository

Explore

- Summary
- Namespaces
- Contexts
- Types
- Explore
- Query
- Export

Modify

- SPARQL Update
- Add
- Remove
- Clear

System

- Information

Current Selections:
Sesame server: <http://localhost:8080/openrdf-sesame> [\[change\]](#)
Repository: Avions-TFC (DS5) [\[change\]](#)

Query Repository

Query Language: SPARQL

```
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX psys:<http://proton.semanticweb.org/protonsys#>
PREFIX owl:<http://www.w3.org/2002/07/owl#>
PREFIX xsd:<http://www.w3.org/2001/XMLSchema#>
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX pext:<http://proton.semanticweb.org/protonext#>

select (count(*) as ?count) where {?s ?p ?o };
```

Limit results: None

☐ Include inferred statements

Execute

Copyright © Aduna 1997-2011
Aduna - Semantic Power

L'execució de la consulta ens informa que 558 triplets han estat carregats satisfactòriament en el nou magatzem.

OpenRDF Workbench - Que... Gmail: Email from Google

localhost:8080/openrdf-workbench/repositories/DS5/query?queryLn=SPARQL8

Workbench

OpenRDF

Sesame server

Repositories

- New repository
- Delete repository

Explore

- Summary
- Namespaces
- Contexts
- Types
- Explore
- Query
- Export

Modify

- SPARQL Update
- Add
- Remove
- Clear

System

- Information

Current Selections:
Sesame server: <http://localhost:8080/openrdf-sesame> [\[change\]](#)
Repository: Avions-TFC (DS5) [\[change\]](#)

Query Result (1)

Limit results: None

Count

558

Copyright © Aduna 1997-2011
Aduna - Semantic Power

7.8 Consultes

Owlim-Lite (a través de Sesame) implementa SPARQL 1.1. Qualsevol aplicació dissenyada per a treballar amb SPARQL, funcionarà amb Owlim-Lite.

En la secció, proves de rendiment, mostrarem un exemple per a cadascun dels següents tipus de consulta sota Owlim-Lite:

- SELECT
- ASK
- DESCRIBE
- CONSTRUCT

L'execució de les consultes les realitzarem a través de la interfície gràfica Sesame.

7.9 Instal·lació

Per tal de poder descarregar Owlím-Lite de la web d' Ontotext, haurem de sol·licitar una llicència a la següent URL:

<http://www.ontotext.com/Owlim/Owlim-Lite-lite-registration>

Un cop rebuda la llicència, podrem descarregar el software i començar amb la instal·lació del producte.

Owlim-Lite pot utilitzar-se de dues maneres diferents:

- Com a llibreria (integrada en un altre aplicació)
- Com a servidor (via servidor de Sesame)

La instal·lació que tot seguit descriu és per a utilitzar Owlím-Lite via Sesame.

Per tal de poder completar la instal·lació d'aquesta opció necessitarem tenir instal·lat, en el nostre PC, Tomcat.

Instal·lació de Tomcat

Instal·larem Tomcat utilitzant apt-get:

```
sudo apt-get install tomcat7
```

Tot seguit, instal·larem Java (en cas que no tinguem la jdk instal·lada en el nostre PC)

```
sudo apt-get install default-jdk
```

Un cop tinguem Java instal·lat, afegirem Tomcat al fitxer `/.bashrc`

```
sudo nano ~/.bashrc
export JAVA_HOME=/usr/lib/jvm/default-java
export CATALINA_HOME=/usr/share/tomcat7
```

Tot seguit, afegirem un nou usuari amb permisos sobre el gestor d'aplicacions. Per fer-ho, editarem el fitxer `tomcat-users.xml`, afegint la següent línia:

```
<user username="tomcat" password="tomcat" roles="manager-gui,admin" />
```

Ja tenim Tomcat instal·lat, ara només falta aixecar-lo:

```
$. CATALINA_HOME/bin/startup.sh
```

Instal·lació Sesame

Instal·lar Sesame serà una operació molt senzilla. Només hem de copiar els següents fitxers (inclosos en la distribució de Owlím-Lite) al directori webapps de Tomcat:

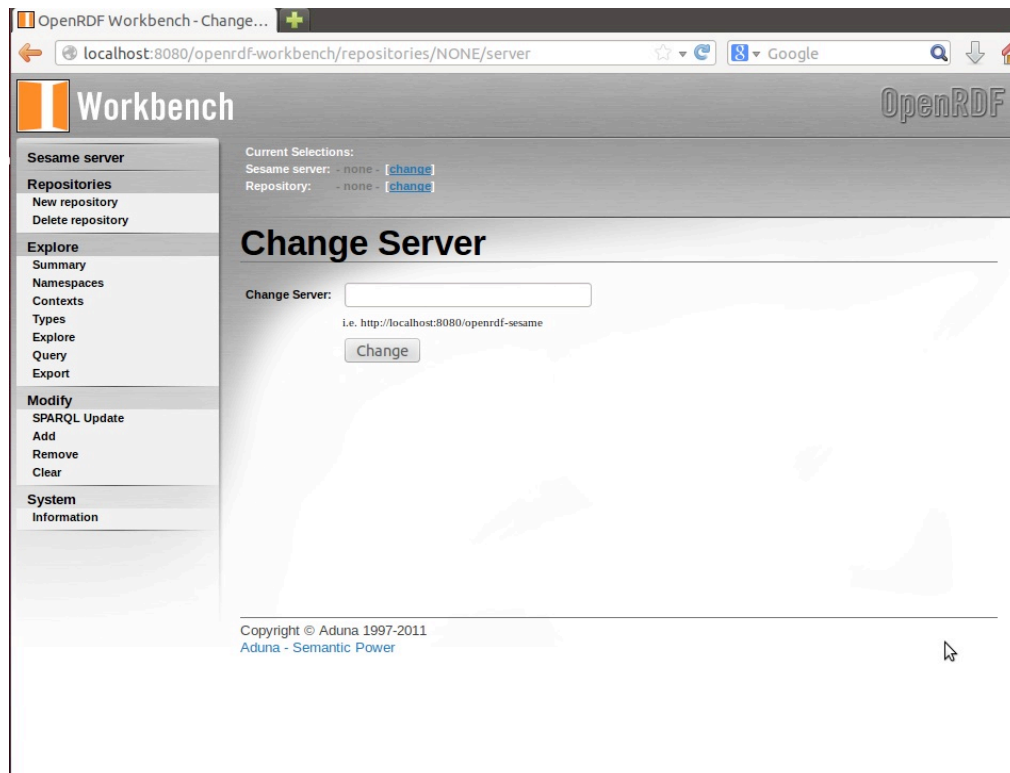
```
cp /camí_Owlím-Lite/openrdf-sesame.war /var/lib/tomcat7/webapps/  
cp /camí_Owlím-Lite/openrdf-workbench.war /var/lib/tomcat7/webapps/
```

Abans d'accedir a l'aplicació haurem de donar accés a Sesame sobre els directoris de log amb les següents comandes:

```
sudo mkdir -p /usr/share/tomcat7/.aduna  
sudo chown -R tomcat7:tomcat7 /usr/share/tomcat7
```

Per accedir a la aplicació, només haurem de carregar la següent URL en el nostre navegador:

<http://localhost:8080/openrdf-workbench/>



8. Proves de rendiment

Un dels objectius d'aquest treball és realitzar una comparativa rigorosa entre Virtuoso i Owlím-Lite. En les dues darreres seccions, hem analitzat les característiques tècniques d'ambdues aplicacions des del punt de vista teòric i pràctic. Per acabar amb la comparativa, realitzarem un conjunt de proves amb l'objectiu de comparar-ne el seu rendiment.

Per tal de mesurar el rendiment d'ambdues aplicacions hem executat dos tipus de proves:

- Càrrega de dades. L'objectiu ha estat mesurar el rendiment de les aplicacions a l'hora de carregar diferents conjunts de dades.
- Execució de consultes. L'objectiu ha estat mesurar el rendiment de les aplicacions a l'hora d'executar consultes sobre els conjunts de dades anteriorment mencionats.

8.1 Joc de dades

Per tal de realitzar les proves de rendiment, hem utilitzat cinc conjunts de dades. El primer d'ells és el fitxer amb les dades de la nostra ontologia. Degut al número limitat de triplets que conté la nostra ontologia he decidit utilitzar quatre fitxers més (disponibles a la dbpedia^[25]) de diferents mides i amb diferent nombre de triplets.

Fitxer	Descripció	Triplets	Mida
Avions.nt	Dades de l'ontologia creada durant el TFC	558	90Kb
Homepages.nt	Propietats de les homepages emprades per la wikipedia	200.036	24Mb
Geoocordinates.nt	Coordenades i GEO localitzacions emprades per la wikipedia	447.517	64Mb
Infoboxes-reduit.nt	Propietats dels infoboxes de la wikipedia	725.712	97.5Mb
Infoboxes.nt	Propietats dels infoboxes de la wikipedia	14.553.905	2.1Gb

8.2 Equip de proves

Processador: Intel Core i7 2.8Ghz (1 nucli habilitat)
Memòria: 1 Gb
Disc dur: 40 Gb
Sistema Operatiu: Ubuntu 12.04 LTS

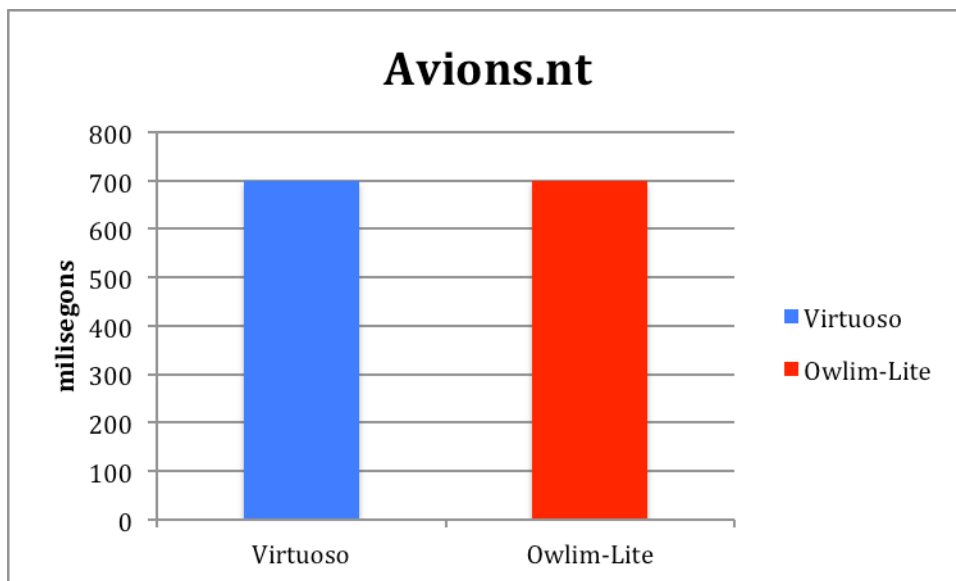
Les proves s'han realitzat amb instàncies de VirtualBox corrent sota OSX 10.8.3.

8.3 Càrrega de dades

La càrrega massiva de dades l'hem realitzat utilitzant les eines disponibles en cadascuna de les aplicacions sota anàlisi. El procediment seguit, per a la càrrega de dades, és el mateix que hem descrit en els anteriors apartats “Càrrega de dades” de les seccions Virtuoso i Owlím-Lite.

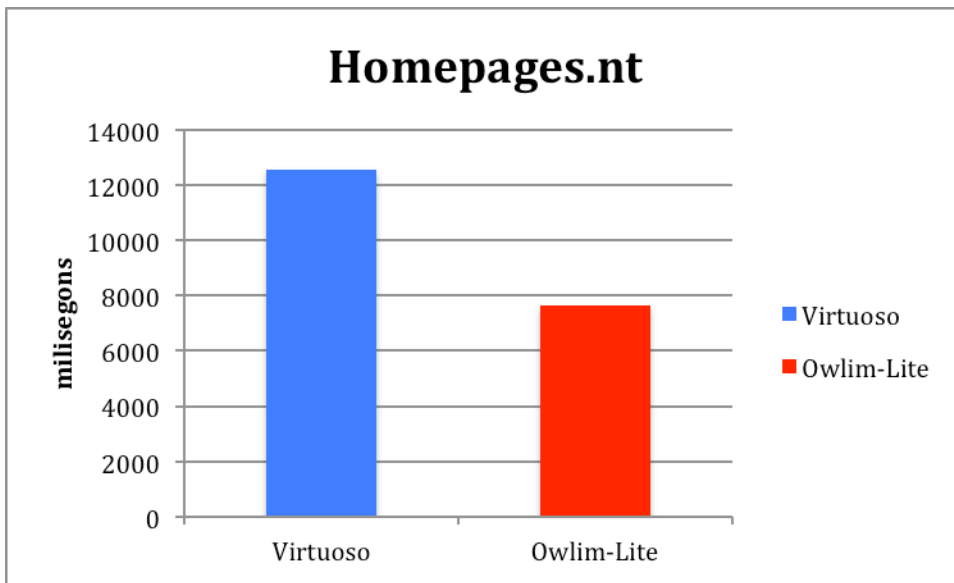
8.3.1 Resultats

Avions.nt (558 triplets. 90Kb)



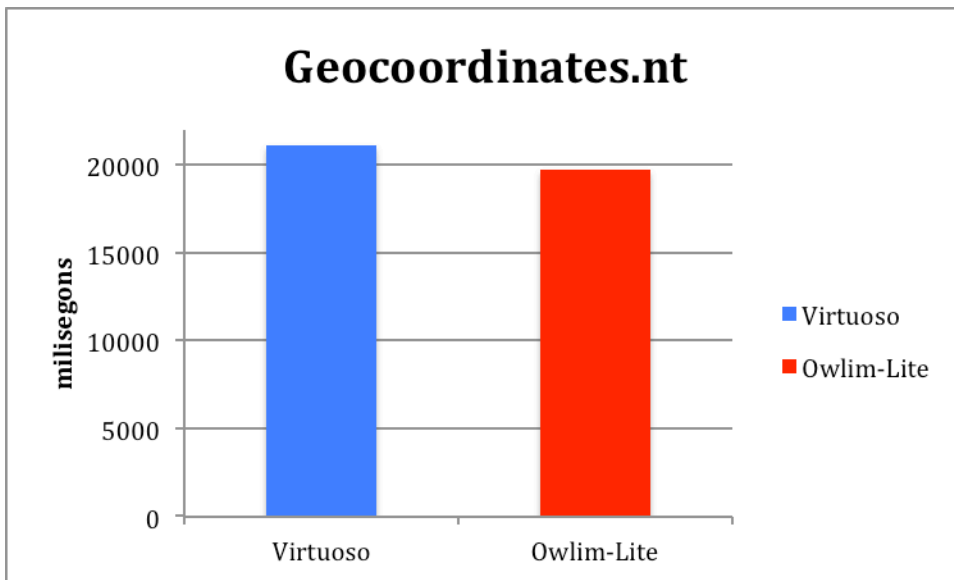
RDF Store	Temps de càrrega
Virtuoso	< 1 segon
Owlím-Lite	< 1 segon

Homepages.nt (200.036 triplets. 24Mb)



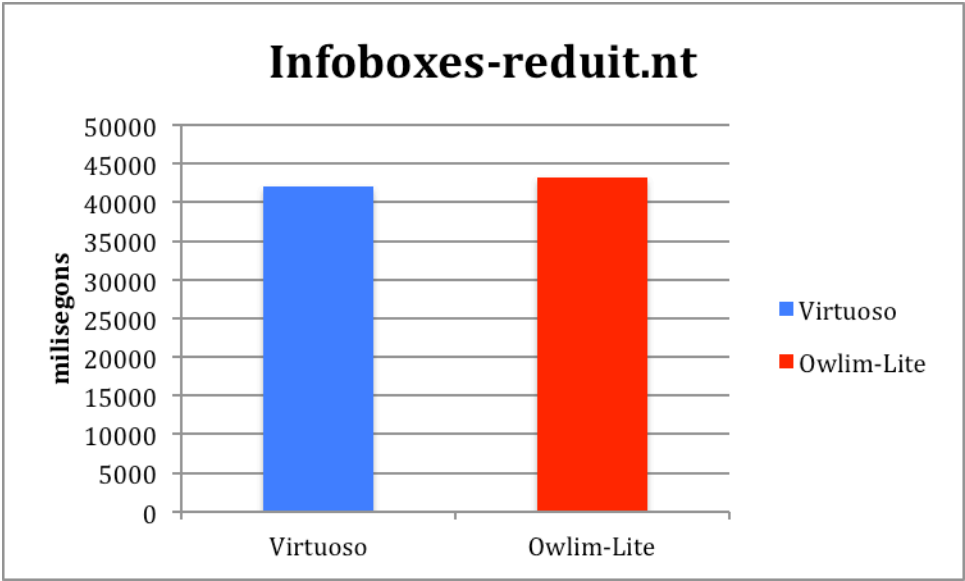
RDF Store	Temps de càrrega
Virtuoso	12573 ms
Owl-Im-Lite	7652 ms

Geocoordinates.nt (447.517 triplets. 64Mb)



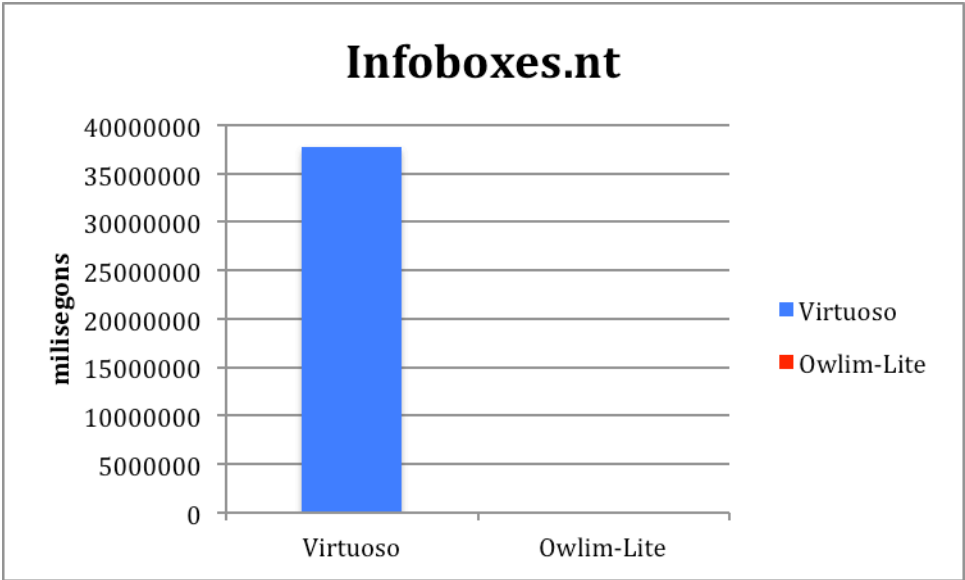
RDF Store	Temps de càrrega
Virtuoso	21095 ms
Owl-Im-Lite	19713 ms

Infoboxes-reduit.nt (725.712 triplets. 97.5 Mb)



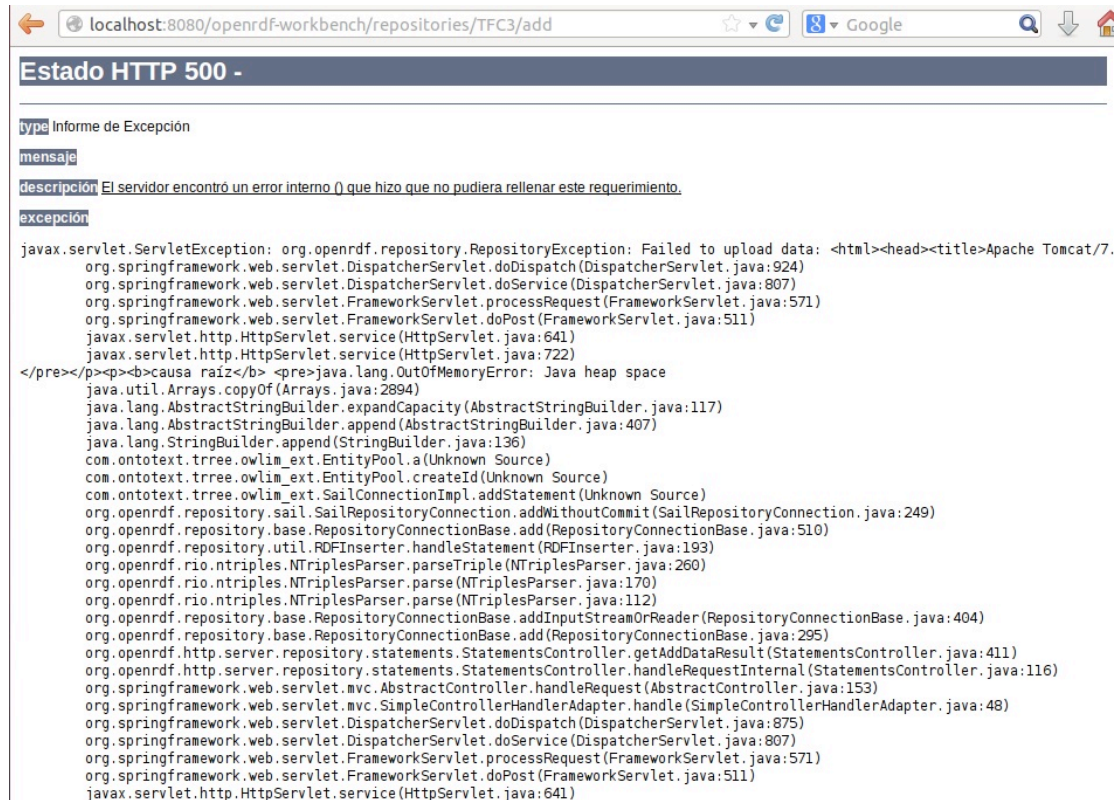
RDF Store	Temps de càrrega
Virtuoso	42014 ms
Owlim-Lite	43213 ms

Infoboxes.nt (14.553.905 triplets. 2.1 Gb)



RDF Store	Temps de càrrega
Virtuoso	3777985 ms
Owlim-Lite	Error

Com es pot observar, a la gràfica de temps de càrrega del fitxer infoboxes.nt, ens ha estat impossible carregar aquest fitxer a Owlím-Lite. Malgrat haver jugat amb diferents parametritzacions del sistema, el resultat ha estat sempre el mateix:



```
Estado HTTP 500 -
type Informe de Excepción
mensaje
descripción El servidor encontró un error interno () que hizo que no pudiera rellenar este requerimiento.
excepción
javax.servlet.ServletException: org.openrdf.repository.RepositoryException: Failed to upload data: <html><head><title>Apache Tomcat/7.
org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:924)
org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:807)
org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:571)
org.springframework.web.servlet.FrameworkServlet.doPost(FrameworkServlet.java:511)
javax.servlet.http.HttpServlet.service(HttpServlet.java:641)
javax.servlet.http.HttpServlet.service(HttpServlet.java:722)
</pre><p><b>causa raíz</b></p><pre>java.lang.OutOfMemoryError: Java heap space
java.util.Arrays.copyOf(Arrays.java:2894)
java.lang.AbstractStringBuilder.expandCapacity(AbstractStringBuilder.java:117)
java.lang.AbstractStringBuilder.append(AbstractStringBuilder.java:407)
java.lang.StringBuilder.append(StringBuilder.java:136)
com.ontotext.trree.owlím_ext.EntityPool.a(Unknown Source)
com.ontotext.trree.owlím_ext.EntityPool.createId(Unknown Source)
com.ontotext.trree.owlím_ext.SailConnectionImpl.addStatement(Unknown Source)
org.openrdf.repository.sail.SailRepositoryConnection.addWithoutCommit(SailRepositoryConnection.java:249)
org.openrdf.repository.base.RepositoryConnectionBase.add(RepositoryConnectionBase.java:510)
org.openrdf.repository.util.RDFInserter.handleStatement(RDFInserter.java:193)
org.openrdf.rio.ntriples.NTriplesParser.parseTriple(NTriplesParser.java:260)
org.openrdf.rio.ntriples.NTriplesParser.parse(NTriplesParser.java:170)
org.openrdf.rio.ntriples.NTriplesParser.parse(NTriplesParser.java:112)
org.openrdf.repository.base.RepositoryConnectionBase.addInputStreamOrReader(RepositoryConnectionBase.java:404)
org.openrdf.repository.base.RepositoryConnectionBase.add(RepositoryConnectionBase.java:295)
org.openrdf.http.server.repository.statements.StatementsController.getAddDataResult(StatementsController.java:411)
org.openrdf.http.server.repository.statements.StatementsController.handleRequestInternal(StatementsController.java:116)
org.springframework.web.servlet.mvc.AbstractController.handleRequest(AbstractController.java:153)
org.springframework.web.servlet.mvc.SimpleControllerHandlerAdapter.handle(SimpleControllerHandlerAdapter.java:48)
org.springframework.web.servlet.DispatcherServlet.doDispatch(DispatcherServlet.java:875)
org.springframework.web.servlet.DispatcherServlet.doService(DispatcherServlet.java:807)
org.springframework.web.servlet.FrameworkServlet.processRequest(FrameworkServlet.java:571)
org.springframework.web.servlet.FrameworkServlet.doPost(FrameworkServlet.java:511)
javax.servlet.http.HttpServlet.service(HttpServlet.java:641)
```

Aquest error es deu, probablement, a les especificacions de la màquina on hem corregut els tests. No obstant, Virtuoso ha estat capaç de carregar tots els fitxers sota les mateixes condicions.

Degut a no poder carregar el fitxer de infoboxes en la seva totalitat a Owlím-Lite, he decidit carregar una versió reduïda (menys triplets) del mateix fitxer que ens permeti executar les proves de consulta que teníem previstes.

8.4 Consultes

Per tal de comprovar el rendiment en l'execució de consultes SPARQL hem generat un total de cinc consultes que avaluen tant el rendiment obtingut amb en l'execució de diferents tipus de consulta (SELECT, ASK, DESCRIBE i CONSTRUCT) com el comportament obtingut treballant amb diferents volums de dades i integrant diferents objectes en un únic conjunt de resultats.

8.4.1 Consulta 1

La consulta 1 ens permetrà comprovar el rendiment de l'execució d'una consulta SELECT on integrem resultats de diferents instàncies de la nostra ontologia.

```
SELECT ?fabricant ?construeix ?motor ?model
where {
?fabricant <http://www.owl-ontologies.com/avions.owl#Nom> ?a.
?fabricant          <http://www.owl-ontologies.com/avions.owl#Construeix>
?construeix.
?construeix <http://www.owl-ontologies.com/avions.owl#MontaMotor> ?motor.
?motor <http://www.owl-ontologies.com/avions.owl#Model> ?model.
}
```

Virtuoso Interactive SQL

localhost:8890/conductor/isql_main.vsp

Database | Replication | Web Application Server | XML | Web Services | Linked Data | NNTP

Interactive SQL

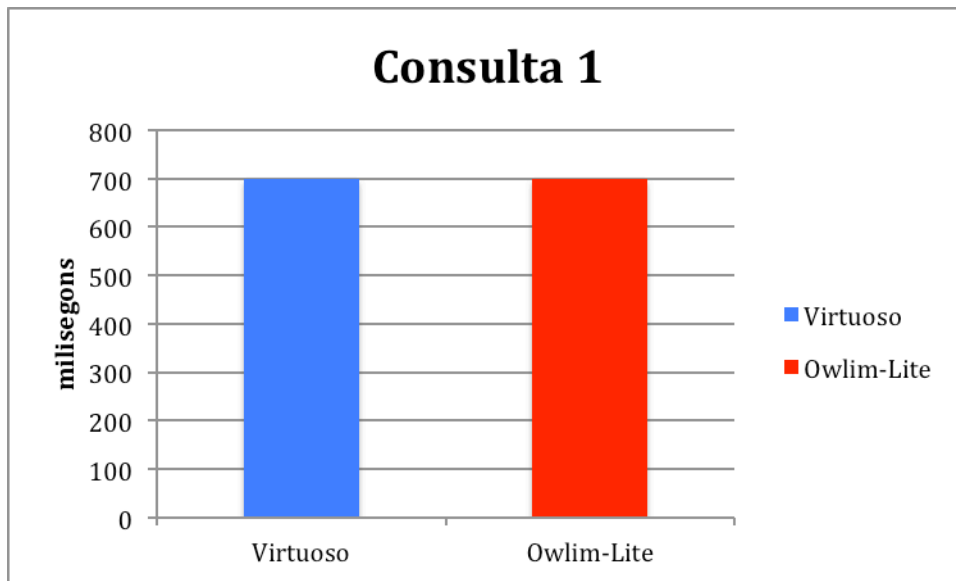
Basic | Advanced

Return

construeix	motor
ANY	ANY
http://www.owl-ontologies.com/avions.owl#Antonov	http://www.owl-ontologies.com/avions.owl#Antonov_An-225
http://www.owl-ontologies.com/avions.owl#Boeing_Commercial_Airplanes	http://www.owl-ontologies.com/avions.owl#Boeing_777
http://www.owl-ontologies.com/avions.owl#Northrop	http://www.owl-ontologies.com/avions.owl#Northrop_F-5
http://www.owl-ontologies.com/avions.owl#Kuznetsov	http://www.owl-ontologies.com/avions.owl#N-1F
http://www.owl-ontologies.com/avions.owl#Tupolev	http://www.owl-ontologies.com/avions.owl#Tupolev_TU-95
http://www.owl-ontologies.com/avions.owl#Fairchild_Aircraft	http://www.owl-ontologies.com/avions.owl#Fairchild_PT-19
http://www.owl-ontologies.com/avions.owl#Nieuport	http://www.owl-ontologies.com/avions.owl#Nieuport_21
http://www.owl-ontologies.com/avions.owl#Bombardier_Aerospace	http://www.owl-ontologies.com/avions.owl#Bombardier_415_SuperScooper
http://www.owl-ontologies.com/avions.owl#Zenair	http://www.owl-ontologies.com/avions.owl#Zenair_Zodiac_CH601
http://www.owl-ontologies.com/avions.owl#Boeing_Commercial_Airplanes	http://www.owl-ontologies.com/avions.owl#Boeing_KC-135_Stratotanker
http://www.owl-ontologies.com/avions.owl#Supermarine_Aviation_Works	http://www.owl-ontologies.com/avions.owl#Supermarine_Spitfire_MKI
http://www.owl-ontologies.com/avions.owl#Hughes_Aircraft	http://www.owl-ontologies.com/avions.owl#Hughes_Hercules_H4
http://www.owl-ontologies.com/avions.owl#Cessna	http://www.owl-ontologies.com/avions.owl#Cessna_Turbo_182_NXT

No. of rows in result: 13

Return



RDF Store	Temps d'execució
Virtuoso	< 1segon
Owlim-Lite	< 1segon

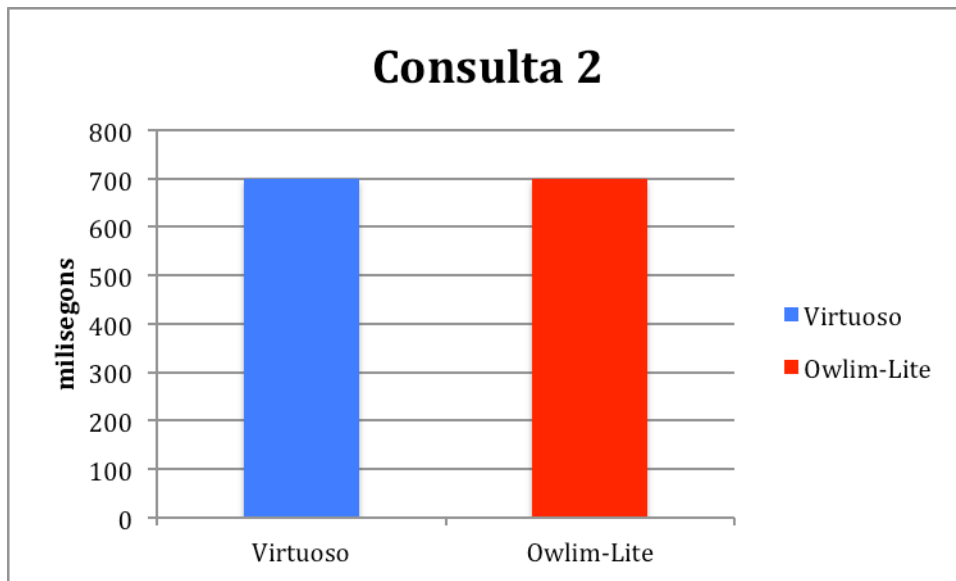
8.4.2 Consulta 2

La consulta 2 ens servirà per avaluar el rendiment d'una consulta del tipus DESCRIBE. Tractarem d'obtenir tots els resultats relacionats amb el recurs Tupolev.

```
DESCRIBE ?p ?o
where {
<http://www.owl-ontologies.com/avions.owl#Tupolev> ?p ?o.
}
```

The screenshot shows the OpenRDF Workbench web interface. The browser address bar indicates the URL: `localhost:8080/openrdf-workbench/repositories/D55/query?queryLn=SPARQL8`. The interface includes a sidebar with navigation options like 'Sesame server', 'Repositories', 'Explore', 'Modify', and 'System'. The main content area displays the 'Query Result (44)' for a SPARQL query. Below the title, there are controls for 'Download format' (set to RDF/XML) and 'Limit results' (set to None). A table of results is shown with three columns: Subject, Predicate, and Object.

Subject	Predicate	Object
http://www.owl-ontologies.com/avions.owl#Seu	rdf:type	owl:DatatypeProperty
http://www.owl-ontologies.com/avions.owl#Seu	rdfs:domain	_node35
http://www.owl-ontologies.com/avions.owl#Seu	rdfs:range	xsd:string
http://www.owl-ontologies.com/avions.owl#Nom	rdf:type	owl:DatatypeProperty
http://www.owl-ontologies.com/avions.owl#Nom	rdfs:domain	_node36
http://www.owl-ontologies.com/avions.owl#Nom	rdfs:range	xsd:string
http://www.owl-ontologies.com/avions.owl#Construeix	rdf:type	owl:ObjectProperty
http://www.owl-ontologies.com/avions.owl#Construeix	rdfs:domain	http://www.owl-ontologies.com/avions.owl#Constructor
http://www.owl-ontologies.com/avions.owl#Construeix	rdfs:range	http://www.owl-ontologies.com/avions.owl#Avió
http://www.owl-ontologies.com/avions.owl#Construeix	owl:inverseOf	http://www.owl-ontologies.com/avions.owl#esConstruitPer
http://www.owl-ontologies.com/avions.owl#esConstruitPer	owl:inverseOf	http://www.owl-ontologies.com/avions.owl#Construeix
http://www.owl-ontologies.com/avions.owl#Constructor	rdf:type	owl:Class

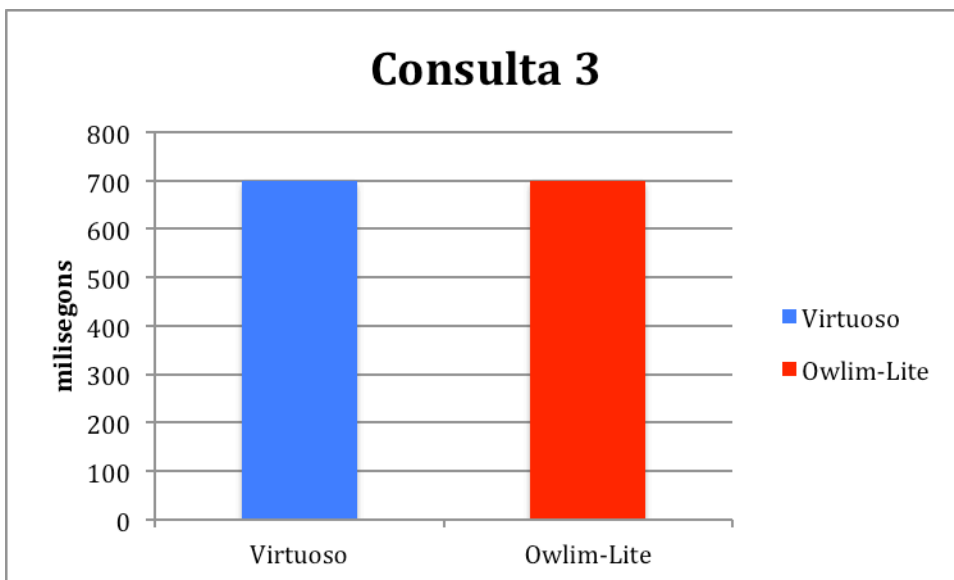
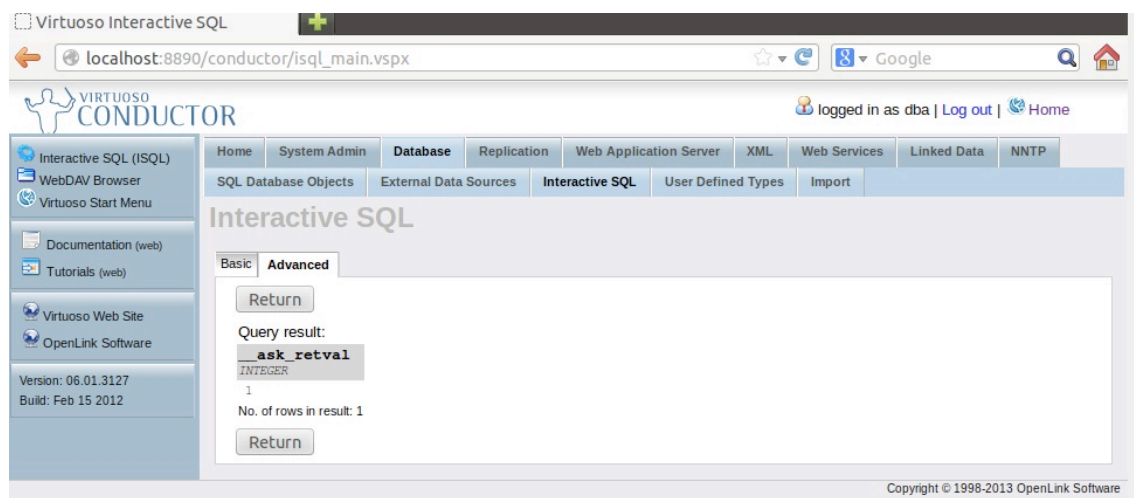


RDF Store	Temps d'execució
Virtuoso	< 1segon
Owlim-Lite	< 1 segon

8.4.3 Consulta 3

La consulta 3 ens servirà per avaluar el rendiment d'una consulta del tipus ASK. La consulta tractarà d'esbrinar si el constructor d'avions Tupolev té la seu a Rússia.

```
ASK
where {
<http://www.owl-ontologies.com/avions.owl#Tupolev> a ?type .
<http://www.owl-ontologies.com/avions.owl#Tupolev>      <http://www.owl-
ontologies.com/avions.owl#Seu> "Rusia"@es .
}
```



RDF Store	Temps d'execució
Virtuoso	< 1segon
Owlrim-Lite	< 1 segon

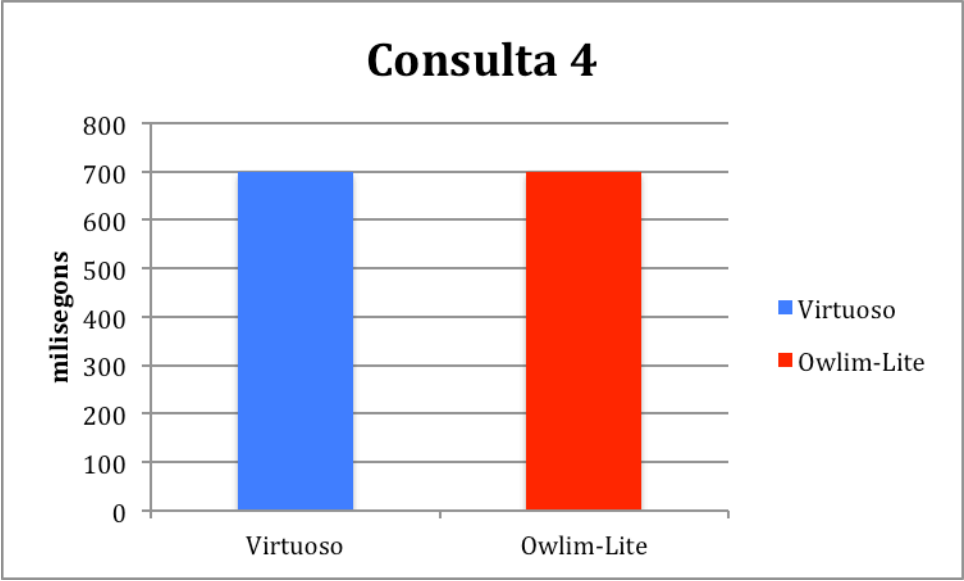
8.4.4 Consulta 4

La consulta 4 ens servirà per avaluar el rendiment d'una consulta del tipus CONSTRUCT. La consulta retornarà tots els grafs disponibles de la nostra ontologia.

```
CONSTRUCT { ?s ?p ?o }  
where { ?s ?p ?o }
```

The screenshot shows the OpenRDF Workbench web application. The browser address bar indicates the URL: `localhost:8080/openrdf-workbench/repositories/DS5/query?queryLn=SPARQL8`. The page title is "OpenRDF Workbench - Query ...". The interface includes a sidebar with navigation options: Sesame server, Repositories, Explore, Modify, and System. The main content area displays the query result for the query `CONSTRUCT { ?s ?p ?o } where { ?s ?p ?o }`. The result is titled "Query Result (139)". Below the title, there are options for "Download format" (RDF/XML) and "Download". The "Limit results" dropdown is set to "None". The results are presented in a table with three columns: Subject, Predicate, and Object. The table contains 19 rows of data, each representing a triple from the ontology.

Subject	Predicate	Object
rdf:type	rdf:type	rdfs:Resource
rdfs:Resource	rdf:type	rdfs:Resource
rdf:Property	rdf:type	rdfs:Resource
rdfs:subPropertyOf	rdf:type	rdfs:Resource
owl:sameAs	rdf:type	rdfs:Resource
rdfs:subClassOf	rdf:type	rdfs:Resource
owl:SymmetricProperty	rdf:type	rdfs:Resource
owl:TransitiveProperty	rdf:type	rdfs:Resource
owl:equivalentProperty	rdf:type	rdfs:Resource
owl:equivalentClass	rdf:type	rdfs:Resource
rdf:type	rdf:type	rdf:Property
rdfs:subPropertyOf	rdf:type	rdf:Property
rdf:subject	rdf:type	rdf:Property
rdf:predicate	rdf:type	rdf:Property
rdf:object	rdf:type	rdf:Property
rdf:first	rdf:type	rdf:Property
rdf:rest	rdf:type	rdf:Property



RDF Store	Temps d'execució
Virtuoso	< 1segon
Owlim-Lite	< 1 segon

8.4.5 Consulta 5

Per tal d'avaluar quin és el rendiment en l'execució de consultes d'ambdues aplicacions amb un volum de dades i complexitat més gran, he decidit utilitzar els fitxers de la dbpedia mencionats a l' inici de la secció.

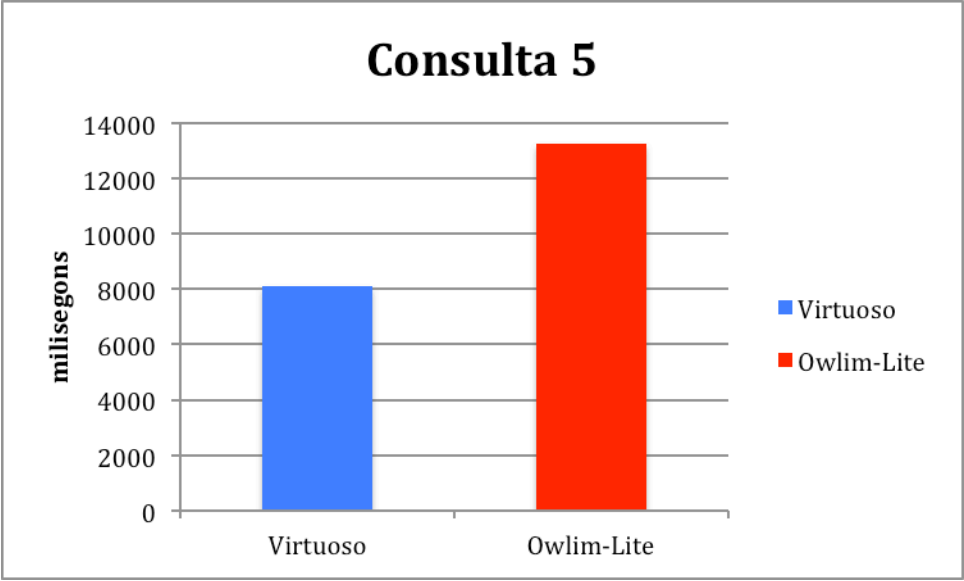
L'objectiu d'aquesta consulta és retornar totes les homepages d'arquitectes de recursos de l'àrea de Barcelona disponibles a la wikipedia.

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84_pos#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX p: <http://dbpedia.org/property/>

SELECT ?s ?a
where {
  <http://dbpedia.org/resource/Barcelona> geo:lat ?bcnLat .
  <http://dbpedia.org/resource/Barcelona> geo:long ?bcnLong .
  ?s geo:lat ?lat .
  ?s geo:long ?long .
  ?s p:architect ?a .
  ?s foaf:homepage ?homepage .
  FILTER (
    ?lat    <= ?bcnLat    + 0.03190235436 &&
    ?long   >= ?bcnLong   - 0.08679199218 &&
    ?lat    >= ?bcnLat    - 0.03190235436 &&
    ?long   <= ?bcnLong   + 0.08679199218
  )
}
```

The screenshot shows the Virtuoso Interactive SQL web interface. The browser address bar displays 'localhost:8890/conductor/isql_main.vsp'. The interface includes a top navigation bar with tabs like 'Home', 'System Admin', 'Database', 'Replication', 'Web Application Server', 'XML', 'Web Services', 'Linked Data', and 'NNTP'. Below this is a sub-navigation bar with 'SQL Database Objects', 'External Data Sources', 'Interactive SQL' (selected), 'User Defined Types', and 'Import'. The main content area is titled 'Interactive SQL' and has two tabs: 'Basic' and 'Advanced'. The 'Basic' tab is active, showing a 'Query result:' section. The result is a table with two columns: 's' (VARCHAR) and 'a' (ANY). The first row contains the URL 'http://dbpedia.org/resource/Camp_Nou' and the name 'Francesc Mitjans-Miró Lorenzo García Barbon Josep Soteras Mauri'. Below the table, it states 'No. of rows in result: 1'. There are 'Return' buttons at the top and bottom of the result area. The footer of the interface indicates 'Copyright © 1998-2013 OpenLink Software'.

s	a
http://dbpedia.org/resource/Camp_Nou	Francesc Mitjans-Miró Lorenzo García Barbon Josep Soteras Mauri



RDF Store	Temps d'execució
Virtuoso	8103 ms
Owlrim-Lite	13267 ms

9. Conclusions

Un cop completat l'anàlisi tècnic, el procés d'instal·lació i les proves de rendiment, descriuré les conclusions a las que he arribat tenint en compte la meva experiència amb les dues aplicacions i els resultats obtinguts. Les conclusions les he dividit en dues grans àrees:

- Instal·lació i usabilitat
- Rendiment

Instal·lació i usabilitat

La principal diferència entre Virtuoso i Owlím-Lite és que Virtuoso es presenta amb un paquet complet, és a dir, no necessita instal·lar software addicional de tercers per tal de poder extreure el màxim rendiment de l'aplicació, mentre que Owlím-Lite no inclou un entorn gràfic d'origen i, per tant, força a l'usuari a instal·lar-ne un (Sesame).

Aquest fet impacta de forma directe en el procés d'instal·lació del producte. Mentre que la instal·lació de Virtuoso és molt senzilla i totalment transparent per a l'usuari (click -> next -> next), Owlím-Lite necessita d'uns coneixements superiors per part de l'usuari, i fer un seguiment precís de les instruccions d'instal·lació proporcionades pel fabricant.

Les dues aplicacions proporcionen la documentació necessària tant per completar la instal·lació del producte, com per extreure'n el màxim partit. Aquesta documentació és fàcilment accessible a la web oficial dels dos productes. Fent una recerca a internet de documents relacionats amb les dues aplicacions (fora de la documentació oficial dels fabricants) he pogut constatar que el nombre de documents disponibles i la qualitat dels mateixos és superior en el cas de Virtuoso.

Totes dues solucions presenten un conjunt de funcionalitats molt completes que permeten a l'usuari treure'n el màxim rendiment i treballar de forma còmode gràcies als seus entorns gràfics des del primer dia. Personalment, he trobat l'entorn gràfic de Virtuoso més senzill, intuïtiu i treballat que el de Owlím-Lite (Sesame). Un exemple molt senzill i il·lustratiu és el fet que cada vegada que s'executi una consulta amb Sesame i s'obtinguin resultats es perdrà el contingut de la consulta, obligant a l'usuari a tornar a escriure la consulta en cas que la vulgui modificar o tornar a executar.

El fet que openLink dissenyi tant la BBDD com l'entorn gràfic, li proporciona un avantatge significatiu en quan a experiència d'usuari. Sesame és un producte molt complet, no obstant, no és propietari d' Ontotext (encara que aquesta companyia participa activament en el seu desenvolupament) i per tant, es molt més complexa, poder dotar d'una experiència d'ús completa, quan depens d'un producte de tercers.

Rendiment

En termes generals, els resultats de les proves realitzades mostren un millor rendiment de Virtuoso (tant en càrrega de dades com en consulta), especialment quan treballem amb grans volums de dades.

Amb volums de dades petits, els resultats s'igualen molt, tant en la càrrega de dades com en l'execució de consultes, on Owlím-Lite es mostra lleugerament més competitiu que Virtuoso.

Un altre punt a destacar és la sensibilitat del sistema a l'hora de carregar dades. Mentre que a Virtuoso he pogut carregar totes les dades sense haver-hi de realitzar cap modificació, Owlím-Lite s'ha mostrat més sensible, i he hagut de modificar algunes línies d'entrada per tal de poder continuar amb la càrrega.

Tenint en compte les limitades característiques de les proves realitzades (en quan a equip, consultes i jocs de dades), he volgut comparar els resultats obtinguts amb d'altres proves realitzades anteriorment disponibles a internet per veure si els resultats podien donar-se per bons.

Ranking of single machine data stores

Query 1	Query 2	Query 3	Query 4
1. Virtuoso (7)	1. Virtuoso (7)	1. 4store (7)	1. Virtuoso (7)
2. 4store (14)	2. 4store (17)	2. BigOWLIM(16)	2. Bigdata (19)
3. BigOWLIM (24)	3. BigOWLIM (18)	3. Bigdata (22)	3. BigOWLIM (20)
4. Bigdata (25)	4. Bigdata (28)	4. Virtuoso (25)	4. 4store (24)

La taula anterior mostra els resultats obtinguts en l'estudi realitzat per Venkata Patchigolla^[26] (Universitat de Purdue) al novembre del 2011. Tot i que, les aplicacions sota estudi són versions superiors de les analitzades en aquest treball, les conclusions són molt similars. Virtuoso mostra de forma consistent un millor rendiment que Owlím-Lite.

Els objectius plantejats han estat assolits plenament. Vaig començar el TFC sense conèixer gairebé res sobre la web semàntica, les ontologies, SPARQL i les bases de dades semàntiques i en aquests moments puc afirmar que tinc un coneixement acceptable (encara que m'agradaria poder haver aprofundit molt més en algun d'ells) sobre totes aquestes matèries.

En línies generals, considero que la planificació del projecte ha estat correcte. La definició de les tasques i el temps establert per a completar-les eren adequats. No obstant, no puc estar content amb el seguiment realitzat. Hauria d'haver previst aquesta possible contingència, augmentant la dedicació (en hores no en temps) de cadascuna de les tasques.

Un cop acabat el projecte, han quedat algunes línies de treball on m'hagués agradat poder aprofundir una mica més. La utilització d'índexs, els sistemes de cache, concurrència o l'escalabilitat dels sistemes han quedat fora de l'anàlisi, deixant-lo, finalment, incomplet. Incloure un anàlisi d'aplicacions reals de les BBDD analitzades, conèixer els clients que l'utilitzen o el tipus de mercat on treballen haurien ajudat a entendre millor els resultats obtinguts en les proves de rendiment.

10. Glossari

- **BD** : Base de dades
- **DTD** (Document Type Definition) : Definició de tipus de document
- **OWL** (Ontology Web Language): Llenguatge per a representar ontologies basades en la lògica descriptiva.
- **RDF** (Resource Description Framework): Llenguatge per a la representació de recursos a internet. SAIL :
- **SGDB** : Sistema gestor de base de dades.
- **SOAP** (Simple Object Access Protocol): Protocol per a l'enviament de missatges entre els serveis web.
- **SPARQL** : (Simple Protocol and RDF Query Language): Protocol simple i llenguatge de consulta RDF.
- **TRREE** : (Triple Reasoning and Rule Entailment Engine): Raonament de triplets i motor de vinculació de regles.
- **Unicode** : Assigna enters no negatius als caràcters escrits de cada idioma i estableix la codificació binària d'aquests nombres.
- **URI** (Universal Resource Identifier): Serveix per a identificar un recurs de forma única.
- **WSDL** (Web Service Description Language): Llenguatge de definició d'interfaces.
- **XML** (eXtensible Mark-up Language): Llenguatge extensible de marques.
- **XML Schema** : Llenguatge per a especificar documents XML.
- **W3C** : World Wide Web Consortium.

11. Bibliografia i Referències

- [1] **Berners-Lee, T.** (1999). Weaving the Web. San Francisco: Harper.
- [2] **Wikipedia – “Semantic Web”** - http://en.wikipedia.org/wiki/Semantic_Web (a 4 de Març del 2013)
- Aula Virtual – “WSemantica”** - http://aulavirtual.catedra.com.co:8081/mnt/apache/editordata/editordatadian/0/Redes_semanticas1537/files/WSemantica.pdf (visitada 04/03/2013)
- [3] **Wikipedia – “Uniform Resource Identifier”** - http://en.wikipedia.org/wiki/Uniform_resource_identifier (visitada 08/03/2013)
- [4] **Wikipedia – “Unicode”** - <http://en.wikipedia.org/wiki/Unicode> (visitada 08/03/2013)
- [5] **Wikipedia – “XML”** - <http://en.wikipedia.org/wiki/XML> (visitada 10/03/2013)
- [6] **W3C – “XML Schema”** - <http://www.w3.org/XML/Schema> (visitada 10/03/2013)
- [7] **W3C – “RDF”** - <http://www.w3.org/2001/sw/wiki/RDF> (visitada 11/03/2013)
- [8] **W3C – “RDFS”** - <http://www.w3.org/2001/sw/wiki/RDFS> (visitada 11/03/2013)
- [9] **W3C – “OWL”** - <http://www.w3.org/2001/sw/wiki/OWL> (visitada 14/03/2013)
- [10] **W3C – “SPARQL”** - <http://www.w3.org/2001/sw/wiki/SPARQL> (visitada 14/03/2013)
- W3C – “RDF SPARQL Query”** - www.w3.org/TR/rdf-sparql-query/ (visitada 14/03/2013)
- [11] **Wikipedia – “Ontology”** - [http://en.wikipedia.org/wiki/Ontology_\(computer_science\)](http://en.wikipedia.org/wiki/Ontology_(computer_science)) (visitada 22/03/2013)
- Wikipedia – “Web Ontology Language”** - http://en.wikipedia.org/wiki/Web_Ontology_Language (visitada 22/03/2013)
- W3C – “OWL2”** - <http://www.w3.org/TR/owl2-overview/> (visitada 26/03/2013)
- W3C – “OWL Guide”** - <http://www.w3.org/TR/owl2-overview/> (visitada 26/03/2013)

[12] Stanford – “Desarrollo de Ontologías” -

http://protege.stanford.edu/publications/ontology_development/ontology101-es.pdf (visitada 26/03/2013)

[13] Protégé – “Main Page” - <http://protege.stanford.edu/> (visitada 18/03/2013)

[14] Wikipedia – “Sistema Gestores de Bases de Datos” -

http://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_bases_de_datos (visitada 18/03/2013)

[15] SECCPERU – “Base de datos semanticas” -

<http://seccperu.org/files/bdsemantica.pdf> (visitada 18/03/2013)

W3C – “Semantic Web Standards” -

http://www.w3.org/2001/sw/wiki/Main_Page (visitada 18/03/2013)

[16] UNALMED – “Acercamiento modelos bases de datos” -

http://www.unalmed.edu.co/~mstabare/acercamiento_modelos_bases.pdf (visitada 22/03/2013)

[17] Wikipedia – “Logical Schema” -

http://en.wikipedia.org/wiki/Logical_schema (visitada 24/03/2013)

[18] Wikipedia – “Modelo Entidad Relación” -

http://es.wikipedia.org/wiki/Modelo_entidad-relaci%C3%B3n (visitada 24/03/2013)

[19] Wikipedia – “Large TripleStores” -

<http://www.w3.org/wiki/LargeTripleStores> (visitada 24/03/2013)

[20] Wikipedia – “TripleStores” - <http://en.wikipedia.org/wiki/Triplestore>

(visitada 24/03/2013)

[21] Wikipedia – “XPath” - <http://es.wikipedia.org/wiki/XPath> (visitada

25/03/2013)

W3C – “XPath” - <http://www.w3.org/TR/xpath/> (visitada 25/03/2013)

W3CSchools – “XPath” - <http://www.w3schools.com/xquery/> (visitada

25/03/2013)

[22] Wikipedia – “XQuery” - <http://es.wikipedia.org/wiki/XQuery> (visitada

25/03/2013)

W3C – “XQuery” - <http://www.w3.org/TR/xquery/> (visitada 25/03/2013)

W3CSchools – “XQuery” - <http://www.w3schools.com/xpath/> (visitada

25/03/2013)

[23] Openlink – “Virtuoso Main page” - <http://virtuoso.openlinksw.com/> (visitada 21/03/2013)

[24] Ontotext – “Owlim Main page” - <http://www.ontotext.com/owlim> (visitada 21/03/2013)

[25] Dbpedia – “Downloads” - <http://dbpedia.org/Downloads38> (visitada 18/04/2013)

[26] Purdue University– “Comparison of Clustered RDF Data Stores” - <http://docs.lib.purdue.edu/techmasters/43/> (visitada 03/05/2013)