



**ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ
ΙΔΡΥΜΑ ΘΕΣΣΑΛΟΝΙΚΗΣ**

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB INTELLIGENCE**

**Δημιουργία SPARQL Endpoint για το Τμήμα
Μηχανικών Πληροφορικής**

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΩΝΣΤΑΝΤΙΝΟΥ Μ. ΜΕΛΙΣΙΔΗ

Επιβλέπων : Δρ. Ευκλείδης Κεραμόπουλος
Επίκουρος Καθηγητής Α.Τ.Ε.Ι. Θεσσαλονίκης

Θεσσαλονίκη, Απρίλιος 2015



ΑΛΕΞΑΝΔΡΕΙΟ ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ
ΘΕΣΣΑΛΟΝΙΚΗΣ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΕΥΦΥΕΙΣ ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ – WEB INTELLIGENCE

Δημιουργία SPARQL Endpoint για το Τμήμα Μηχανικών Πληροφορικής

ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΚΩΝΣΤΑΝΤΙΝΟΥ Μ. ΜΕΛΙΣΙΔΗ

Επιβλέπων : Δρ. Ευκλείδης Κεραμόπουλος
Επίκουρος Καθηγητής Α.Τ.Ε.Ι. Θεσσαλονίκης

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή στις 22 Απριλίου 2015.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

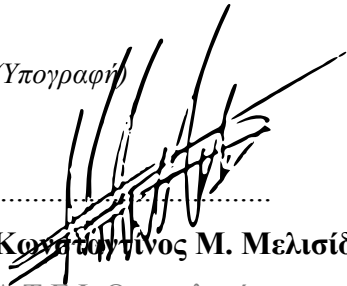
.....
Δρ. Ευκλείδης Κεραμόπουλος
Επίκουρος Καθηγητής
Α.Τ.Ε.Ι.Θ.

.....
Δρ. Παναγιώτης Αδαμίδης
Καθηγητής
Α.Τ.Ε.Ι.Θ.

.....
Δρ. Δημήτριος Δέρβος
Καθηγητής
Α.Τ.Ε.Ι.Θ.

Θεσσαλονίκη, Απρίλιος 2015

(Υπογραφή)



Κωνσταντίνος Μ. Μελισίδης

Α.Τ.Ε.Ι. Θεσσαλονίκης

© Απρίλιος 2015 – All rights reserved

Περίληψη

Ο στόχος της διπλωματικής εργασίας είναι η δημιουργία και παραμετροποίηση μιας πύλης διενέργειας ερωτημάτων SPARQL που να προσφέρει πληροφορίες για το πρόγραμμα σπουδών του τμήματος στο σημασιολογικό ιστό. Η πληροφορία που θα προβληθεί πρέπει να ακολουθεί συγκεκριμένα μορφότυπα, τις οντολογίες. Η συμμετοχή στον ιστό των δεδομένων καθιστά εφικτή τη προβολή και κατανόηση των δεδομένων από τις υπολογιστικές μηχανές, τα ευφυή προγράμματα και τους ανθρώπους. Απώτερος σκοπός είναι η ενεργή συμμετοχή του τμήματος στον ιστό των δεδομένων.

Λέξεις Κλειδιά: SPARQL Endpoint, Semantic Web, ontology, OWL, Virtuoso Universal Server, SPARQL, Virtuoso Conductor, Protégé.

Abstract

In the recent years have seen increasing amount of web information. The semantic web, can organize that kind of information, via ontologies. Information is represented directly to human and also can be understandable via machines. The gates for semantic information retrieval, are sophisticated software server named as SPARQL Endpoint. Via SPARQL Endpoint, we can request any kind of information provided by giving a SPARQL query. This structural information, can recombined with other ontologies and can provide new data. This can be done apparently by software agents. So Endpoints have a serious role in the new web of data. In this project thesis, we are configuring Openlink Virtuoso SPARQL Endpoint, for passing our ontology of MSc Web Intelligence to Semantic Web. Combined with open data, we can publish our ontology via ontology mapping to the LOD cloud. Our sector, education has a lot of benefits from open linked data, such as having universal ranking etc. Our ontology has been published in a variety of formats and has been reused with varying degrees of success. However, the methods and techniques of the semantic web could significantly enhance the value and utility of open data.

Keywords: SPARQL Endpoint, Semantic Web, ontology, OWL, Virtuoso Universal Server, SPARQL, Virtuoso Conductor, Protégé.

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω τον επιβλέποντα καθηγητή μου κ. Ευκλείδη Κεραμόπουλο για τη βοήθεια, τη συμπαράσταση και την εμπιστοσύνη που έδειξε στο πρόσωπό μου, καθώς επίσης και τη γυναίκα μου Γιώτα για την υπομονή και τη συμπαράστασή της.

Πίνακας περιεχομένων

Περίληψη	i
Abstract	iii
1 Εισαγωγή.....	1
1.1 Ιστός Δεδομένων - Web of Data	1
1.2 Πύλη διενέργειας ερωτημάτων – SPARQL Endpoint	2
1.2.1 Συνεισφορά	2
1.3 Οργάνωση κειμένου	3
2 Μεθοδολογία Σχεδίασης	4
2.1 Πύλες διενέργειας ερωτημάτων	4
2.2 Μεθοδολογία Σχεδίασης Οντολογιών.....	6
3 Σημαιολογικός Ιστός.....	8
3.1 Σημαιολογικός Ιστός - Semantic Web	9
3.2 Διασυνδεδεμένα Δεδομένα – Linked Data.....	11
3.2.1 Ανοικτά διασυνδεδεμένα Δεδομένα – Linked Open Data.....	11
4 Οντολογία.....	14
4.1 Οντολογία σημαιολογικού ιστού – Web ontology.....	14
4.1.1 Γλώσσες οντολογιών.....	15
4.1.2 Δομικά στοιχεία οντολογίας OWL.....	18
4.1.3 Ορθότητα οντολογίας.....	22
4.2 Οντολογία μεταπτυχιακού προγράμματος σπουδών.....	23
5 SPARQL.....	31
5.1 Δομή SPARQL.....	32
5.1.1 Ερωτήματα SPARQL.....	35
5.2 SPARQL Endpoint.....	38
6 Τεχνικές λεπτομέρειες.....	46
6.1 Virtuoso Universal Server.....	46
6.1.1 Εγκατάσταση.....	50

6.1.2 Διαμόρφωση.....	53
6.2 Protégé.....	72
6.3 Web Εφαρμογή	77
7 Επίλογος.....	79
7.1 Σύνοψη και συμπεράσματα.....	79
7.2 Μελλοντικές επεκτάσεις	80
Βιβλιογραφία.....	81
Παράρτημα Α	88
<i>Πηγές αναφοράς εικόνων - URIs.....</i>	<i>88</i>

1

Εισαγωγή

1.1 Ιστός Δεδομένων - Web of Data

Η ραγδαία εξέλιξη του παγκόσμιου ιστού εγγράφων, οδήγησε τον ιστό στο επόμενο στάδιο, τον ιστό των δεδομένων με σκοπό την μηχανογνώσιμη μορφή του [1] [4]. Η κατανάλωση της πληροφορίας πλέον μπορεί να ερμηνευτεί και από ευφυή λογισμικά, πέραν των ανθρώπων. Η δυνατότητα αυτή, προσδίδει νέες τεχνολογικές προκλήσεις στον ιστό δεδομένων. Η διασύνδεση των δεδομένων μεταξύ τους γίνεται βάσει κανόνων που επιφέρει ο σημασιολογικός ιστός.

Τα δεδομένα, πλέον, υφίστανται με μια διαφορετική μορφή από τα παραδοσιακά έγγραφα, τις οντολογίες. Η αναζήτηση μπορεί να γίνει στα δεδομένα ή και να συνδυαστεί με συναφή δεδομένα [2] [6]. Μια νέα κατηγορία λογισμικού, οι πύλες διενέργειας ερωτημάτων είναι υπεύθυνες για την ανάλυση και εκτέλεση των αναζητήσεων, μέσω κατάλληλα προτυποποιημένων ερωτημάτων SPARQL [2].

Σκοπός της διπλωματικής είναι η παρουσίαση των δεδομένων, για την περιγραφή του προγράμματος σπουδών, σε μηχαναγνώσιμη μορφή στον ιστό δεδομένων.

1.2 Πύλη διενέργειας ερωτημάτων – SPARQL Endpoint

Πρόκειται για μια υπηρεσία που έχει σκοπό την παροχή πληροφοριών σε κάθε επερώτηση που δέχεται απο τον χρήστη ή λογισμικό στη γλώσσα ερωτημάτων SPARQL [3] [4]. Οι πύλες είναι άρρηκτα συνδεδεμένες με την βασική ιδέα του σημασιολογικού ιστού.

Υπάρχουν και αντικρουόμενες απόψεις σχετικά με το λόγο ύπαρξης ενός SPARQL Endpoint, όπως φαίνεται στο παρακάτω απόσπασμα:

“Ποιος ο λόγος ύπαρξης ενός Endpoint, όταν τα ερωτήματα που δέχεται ένα Endpoint είναι περίπλοκα και μπορούν να διαρκέσουν αρκετά δευτερόλεπτα, και πολλές φορές, μπορούν να αποτύχουν, δίνοντας την εντύπωση ενός πρώιμου λογισμικού σε παραγωγή; Επίσης λόγω του φόρτου σε τέτοιες περιπτώσεις πιθανόν να υπάρχει και μη διαθεσιμότητα της υπηρεσίας” [7].

Η απάντηση δεν μπορεί να είναι άλλη παρά ο ορισμός των διασυνδεδεμένων δεδομένων και ανοικτών διασυνδεδεμένων δεδομένων. Η ουσία βρίσκεται στη δημιουργία του ιστού δεδομένων [5][6].

Άλλη εκδοχή προσπαθεί να υποβαθμίσει την πύλη διενέργειας ερωτημάτων με την αντικατάσταση απο RESTful API για ανοικτά δεδομένα [8].

Η πύλη διενέργειας ερωτημάτων, αποτελεί τον ακρογωνιαίο λίθο της σχέσης σημασιολογικού ιστού και SPARQL. Κύριος εκπρόσωπος στο τομέα είναι το λογισμικό Virtuoso Universal Server (middleware), που αποτελεί μια υπερ-υπηρεσία σε συνδυασμό πολλών διαφορετικών υπηρεσιών [9] [10] [11].

1.2.1 Συνεισφορά

Η μεταπτυχιακή διπλωματική εργασία έχει ως αντικείμενο τη δημιουργία κατάλληλης οντολογίας για την περιγραφή του μεταπτυχιακού προγράμματος σπουδών, τη δημιουργία πύλης διενέργειας ερωτημάτων και προβολή της οντολογίας διαμέσου αυτής.

Η συνεισφορά της διπλωματικής συνοψίζεται ως εξής:

1. Μελετήσαμε κατάλληλες οντολογίες
2. Αναπτύξαμε την κατάλληλη οντολογία για το Π.Μ.Σ
3. Υλοποιήσαμε τη δημιουργία και παραμετροποίηση του SPARQL Endpoint
4. Αναπτύξαμε μια web εφαρμογή για τον έλεγχο

1.3 Οργάνωση κειμένου

Η δομή της εργασίας αποτελείται από οκτώ (8) κεφάλαια. Στα πρώτα κεφάλαια παρέχονται βασικές προαπαιτούμενες γνώσεις και εν συνεχεία δίνεται η ολοκληρωμένη εικόνα του έργου. Πέρα από την εισαγωγή στο κεφάλαιο 1, γίνεται αναφορά στη μεθοδολογία σχεδίασης της οντολογίας και της πύλης διενέργειας ερωτημάτων στο κεφάλαιο 2.

Στο κεφάλαιο 3, γίνεται μια εισαγωγή στην έννοια του σημασιολογικού ιστού (Semantic Web) και στις δυνατότητες των νέων χαρακτηριστικών του. Η βασική δομική πληροφορία του σημασιολογικού ιστού είναι η οντολογία [1], που περιγράφεται λεπτομερώς στο κεφάλαιο 4. Δίνεται η ολοκληρωμένη οντολογία του τμήματος μηχανικών πληροφορικής του μεταπτυχιακού. Στο επόμενο κεφάλαιο γίνεται αναφορά στη SPARQL, τη γλώσσα ερωτημάτων του σημασιολογικού ιστού[3]. Η υπηρεσία παροχής τέτοιου είδους πληροφορίας υλοποιείται από το λογισμικό διενέργειας ερωτημάτων SPARQL Endpoint. Ακολουθεί το κεφάλαιο 6 των τεχνικών λεπτομερειών, στο οποίο περιγράφονται αναλυτικά τα λογισμικά που υιοθετήθηκαν, παραμετροποιήθηκαν και συνεργάστηκαν ώστε να περατωθεί ο σκοπός της εργασίας. Στο τελευταίο κεφάλαιο 7, δίνεται η σύνοψη και οι μελλοντικές επεκτάσεις που θα μπορούσαν να ενσωματωθούν στο πεδίο αυτό.

2

Μεθοδολογία Σχεδίασης

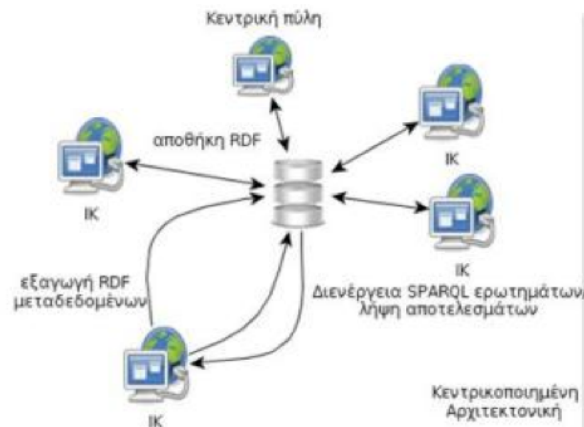
2.1 Πύλες διενέργειας ερωτημάτων

Το λογισμικό διενέργειας ερωτημάτων, υλοποιεί μια υπηρεσία παροχής δεδομένων. Η υπηρεσία είναι υπεύθυνη για την επεξεργασία ερωτημάτων SPARQL [3].

Η αρχιτεκτονική που έχουμε επιλέξει για την διεκπεραίωση της διπλωματικής είναι η κεντροποιημένη αρχιτεκτονική της υπηρεσίας, παρόμοια με μια κεντρική πύλη.

Η παροχή των RDF μεταδεδομένων [1] και η αποθήκευσή τους γίνεται κεντρικά σε ένα μοναδικό RDF αποθετήριο (RDF repository).

Η αρχιτεκτονική αυτή θα απλουστεύσει τη διαδικασία αναζήτησης καθώς θα εξασφαλίζει μοναδικό σημείο διενέργειας ερωτημάτων. Το κεντρικό αυτό RDF αποθετήριο θα είναι προσβάσιμο για την παροχή προηγμένων υπηρεσιών τόσο από το εσωτερικό υποδίκτυο της σχολής όσο και από τα επιμέρους εξωτερικά Endpoints μέσω της κεντρικής πύλης (εικόνα 1).



Εικόνα 1: Κεντροποιημένη Αρχιτεκτονική

Τα λογισμικά που διερευνήθηκαν για τη δημιουργία της πύλης είναι τα εξής:

- Virtuoso Universal Server

Είναι ένα εμπορικό προϊόν που περιλαμβάνει όλες τις λειτουργίες που θα χρησιμοποιήσουμε στην εργασία μας [9].

- Virtuoso open-source edition

Είναι η έκδοση Ελεύθερου Λογισμικού / Λογισμικού Ανοικτού Κώδικα (ΕΛ/ΛΑΚ) του Virtuoso universal server, στερείται ορισμένων προηγμένων λειτουργιών όπως διασύνδεση εικονικών ΒΔ [12] [13].

- WSO2 Data Services Server

Αποτελεί έργο ΕΛ/ΛΑΚ, και μπορεί να λειτουργήσει ως υπηρεσία παροχής πληροφοριών. Μπορεί να συνδυάσει αρκετές πηγές για την εξαγωγή πληροφορίας [14].

- TopBraid Live

Πρόκειται για εμπορικό προϊόν SPARQL Endpoint. Είναι προσανατολισμένο στο σημασιολογικό οικοσύστημα [15].

- Logging/Authenticating SPARQL client and server

Πρόκειται για ΕΛ/ΛΑΚ, με τη πρόσθετη δυνατότητα αυθεντικοποίησης των χρηστών [16].

Υπάρχει πληθώρα SPARQL Endpoints παγκοσμίως και η διαθεσιμότητα αυτών είναι ζωτικής σημασίας [17].

Επιλέξαμε το Virtuoso open-source edition, γιατί είναι ένα λογισμικό που υλοποιεί πολλές λειτουργίες πέραν του Endpoint όπως περιγράφεται αναλυτικά στο κεφάλαιο 6 [12] [13].

2.2 Μεθοδολογία Σχεδίασης Οντολογιών

Μια οντολογία είναι ένα λεξικό όρων που διατυπώνονται με μια τυπική σύνταξη και με τους γενικά αποδεκτούς ορισμούς τους [18]. Στόχος είναι η παραγωγή ενός λεξικολογικού ή ταξινομικού πλαισίου αναπαράστασης της γνώσης ώστε να μπορεί να χρησιμοποιείται από κοινού από τις διαφορετικές κοινότητες πληροφορικών συστημάτων [19] [20].

Η δημοφιλέστερη μεθοδολογία ανάπτυξης οντολογιών, θα πρέπει να ακολουθεί τα βασικά βήματα [61]:

- προσδιορισμός του σκοπού της οντολογίας
- κατασκευή της οντολογίας
- αξιολόγηση της οντολογίας
- τεκμηρίωση της οντολογίας

Τα βήματα που ακολουθήθηκαν για τη δημιουργία της οντολογίας είναι [2] [21] [22] [23] [24] [25]:

- προσδιορισμός του σκοπού της οντολογίας
- ανίχνευση και δημιουργία κλάσεων
- ιεραρχία των κλάσεων
- συσχετίσεις σχέσεων κλάσεων της οντολογίας
- ιδιότητες κλάσεων
- αξιώματα οντολογίας
- στιγμιότυπα οντολογίας
- έλεγχος ορθότητας οντολογίας
- αξιολόγηση της οντολογίας

Οι πιο γνωστές μεθοδολογίες για τη δημιουργία οντολογιών είναι:

- Methontology [26]
- On-To-Knowledge [27]
- KBSI IDEF5 [28]

Υπάρχουν και άλλες μεθοδολογίες που είναι άρρηκτα συνδεδεμένες με τα εργαλεία συγγραφής όπως WebODE [29], Chimera [30], protégé [21] και Cognitum Fluent Editor.

Για τη συγγραφή χρησιμοποιήθηκε η άνωθεν μεθοδολογία protégé, με το εργαλείο protégé [21].

3

Σημασιολογικός Ιστός

Πατέρας του σημασιολογικού ιστού φέρεται να είναι ο Σερ Τίμοθι Τζον Μπέρνερς Λι (Sir Timothy John Berners-Lee), που είναι ταυτόχρονα και πατέρας του παγκόσμιου ιστού [1] [31]. Εντούτοις, αναφορές σχετικές με τη σημασιολογική γνώση έγιναν δεκαετίες πριν, στη δεκαετία του '60, όπου τέθηκαν οι βασικές αρχές για τη δόμηση της πληροφορίας σε σημασιολογική γνώση [32] [33].

Ο δημιουργός οραματίστηκε το σημασιολογικό ιστό ως [31]:

“Έχω ένα όνειρο. Αυτό είναι να φτάσουμε στο επίπεδο όπου οι υπολογιστές να είναι ικανοί να αναλύουν όλες τις πληροφορίες που υπάρχουν στο Διαδίκτυο... με απώτερο σκοπό: Οι καθημερινές μας ασχολίες που έχουν να κάνουν π.χ. με την επικοινωνία, το εμπόριο, την γραφειοκρατία, να τις διαχειρίζονται οι μηχανές μεταξύ τους δίνοντας στον άνθρωπο περισσότερο ελεύθερο χρόνο”

3.1 Σημασιολογικός Ιστός - *Semantic Web*

Ο σημασιολογικός ιστός αποτελεί την εξέλιξη του παγκόσμιου ιστού που γνωρίζαμε μέχρι σήμερα (Web 2.0). Η εξέλιξη υιοθετεί τη νέα οπτική που μπορεί να μας δώσει ο παγκόσμιος ιστός, η πληροφορία να είναι σε μηχαναγνώσιμο μορφότυπο, δηλαδή κατανοητή και για τις μηχανές [34].

Πρόκειται να αποτελέσει μια οικουμενική βάση δεδομένων, όπου όλα τα στοιχεία θα συσχετίζονται μεταξύ τους. Στα επόμενα χρόνια, αναμένεται να παίξει σημαντικό ρόλο στη σύγχρονη κοινωνία και να αποτελέσει κομμάτι αυτής.

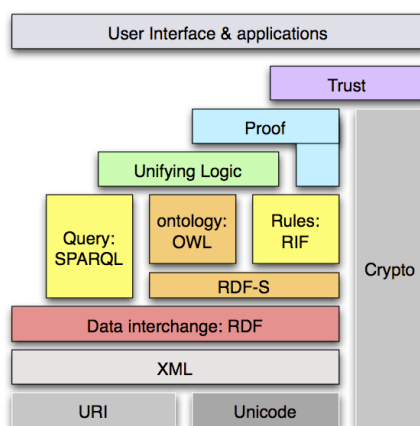
Η χρησιμοποίηση της υπάρχουσας γνώσης μαζί με νέα χαρακτηριστικά έχει χαρακτηρίσει τα δομικά στοιχεία του σημασιολογικού ιστού, τις οντολογίες. Η οντολογία ενθουλακώνει την απαραίτητη γνώση με μεταδεδομένα (metadata), για τη παροχή γνωσιακής μηχαναγνώσιμης πληροφορίας [1] [2]. Αναλυτικότερα για τις οντολογίες στο επόμενο κεφάλαιο 4.

Η οντολογία συνδέεται με άλλες οντολογίες και δημιουργεί τα διασυνδεδεμένα δεδομένα, περιγράφοντας τη γνώση που υπάρχει. Παραλληλίζοντας θα λέγαμε ότι δημιουργείται μια οικουμενική βάση δεδομένων της υπάρχουσας γνώσης. Η δομή των ανοικτών διασυνδεδεμένων δεδομένων (Linked Open Data) στον ωκεανό της σύγχρονης γνώσης, φαντάζει περίπλοκη [2] [7].

Δίνεται μια άλλη εκδοχή της επίνοιας του σημασιολογικού ιστού:

“Σημασιολογικός Ιστός είναι ένα σύνολο προτύπων που μετατρέπει το web σε μια μεγάλη βάση δεδομένων” [35].

Η τεχνολογία του σημασιολογικού ιστού βασίζεται σε συνδυασμό των ήδη υπάρχουσών τεχνολογιών (URI, XML) με νέες έννοιες όπως RDF, RDFS, OWL, OWL 2 [1] [2] [5].



Εικόνα 2. Η διαστρωμάτωση του Σημασιολογικού Ιστού Tim Berners-Lee

Για το επίπεδο των οντολογιών, η διαστρωμάτωση του Σημασιολογικού Ιστού προβλέπει την ύπαρξη γλωσσών για οντολογίες (ontologies), για κανόνες (rules) και για ερωτήματα SPARQL [1], όπως απεικονίζεται στην εικόνα 2.

Οι τομείς που εμφανίζονται συχνότερα στην τωρινή εκδοχή του σημασιολογικού ιστού είναι αυτοί της παιδείας, της υγείας και της οικονομίας. Στη παιδεία, προστιθέμενη αξία είναι η διαδικασία της διευρευνητικής μάθησης και η ανάκτηση έγκυρης πληροφορίας.

Στο τομέα της υγείας, που ίσως είναι ο συχνότερα εμφανιζόμενος τομέας του σημασιολογικού ιστού – γίνεται μεγάλη προσπάθεια διασύνδεσης ιατρικών εννοιών και συσχέτιση τους, σε μια τεράστια ιατρική βάση δεδομένων με αλληλένδετες ιατρικές πληροφορίες, τις οντολογίες.

Τα μεταδεδομένα, είναι τα δεδομένα για τα δεδομένα. Αποτελούν μια δομημένη μέθοδο περιγραφής πόρων (resources) και συνδυάζεται με τα δεδομένα. Γενικολογώντας θα λέγαμε ότι πρόκειται για δομημένα δεδομένα που περιγράφουν απτές έννοιες, όπως χαρακτηριστικά αντικειμένων και σχέσεις μεταξύ αντικειμένων. Αλλάζοντας την οπτική θα λέγαμε ότι αποτελούν τον συσχετιστή μεταξύ πόρων περιεχομένων και της σημασίας των περιεχομένων [1] [2]. Ο όρος μεταδεδομένα αποδίδεται στον Philip R. Bagley [36].

Μπορούμε να κατηγοριοποιήσουμε τα μεταδεδομένα σε 2 μεγάλες κατηγορίες στα περιγραφικά (descriptive metadata) και τα δομικά (structural metadata). Τα δομικά μεταδεδομένα περιγράφουν το τρόπο της οργάνωσης των συστατικών μερών της πληροφορίας. Τα περιγραφικά από την άλλη, μας τροφοδοτούν με πληροφορίες σχετικές με την αναζήτηση, σε κατανοητό επίπεδο πιο κοντά στον άνθρωπο [37].

Στον παγκόσμιο ιστό τα μεταδεδομένα αποτελούν σημαντική πηγή πληροφορίας, και συναντώνται ως μετα-ετικέτες (meta tags) στις ιστοσελίδες. Στις αρχές των μηχανών αναζήτησης, αποτελούσε κύρια πηγή πληροφορίας και κατηγοριοποίησης [38]. Τα τελευταία δε χρόνια, δεν αποτελεί την κύρια πηγή, αλλά συνεισφέρει με κάποιο τρόπο [39]. Ακόμη πιο πρόσφατα με την εξέλιξη του σημασιολογικού ιστού τα meta tags, μας δίνουν τις πληροφορίες πόρων για τα μεταδεδομένα [1] [90].

Η υιοθέτηση των μεταδεδομένων, έχει δημιουργήσει διάφορα πρότυπα προς υιοθέτηση με σημαντικότερα τα Dublin core [40] [41] και ACR2 (Anglo-American Cataloging Rules) [42], GILS (Government Information Locator Service) [43], EAD, IMS, AGLS.

Επικρατέστερο πρότυπο είναι το Dublin core, όπου περιγράφεται το σύνολο στοιχείων περιγραφής και ορισμού (elements) για τα μεταδεδομένα, αποτελείται από δεκαπέντε (15) περιγραφές (έκδοση dc1.1) ενδεικτικά αναφέρουμε τα Τίτλος, Συγγραφέας, Θέμα, Κωδικός Πόρου, Πηγή, Δικαιώματα Χρήσης [41].

Η πρόταση της W3C για τα μεταδεδομένα στην HTML, RDFa αναγνωρίζει το πρότυπο Dublin core [90].

3.2 Διασυνδεδεμένα Δεδομένα – *Linked Data*

Ο όρος διασυνδεδεμένα δεδομένα, έχει ορισθεί από το δημιουργό του σημασιολογικού ιστού. Αναφέρεται ως ένα σύνολο πρακτικών για τη δημοσίευση και σύνδεση δομημένων δεδομένων στον σημασιολογικό ιστό. Απώτερος σκοπός των πρακτικών αυτών, είναι η μετεξέλιξη του παραδοσιακού ιστού εγγράφων σε ένα ιστό δεδομένων, όπου κάθε δεδομένο θα έχει κάποια σχέση σύνδεσης με κάποιο άλλο δεδομένο, έγγραφο ή οντότητα. Απαραίτητη προϋπόθεση και μόνο, η ήδη υπάρχουσα τεχνολογία του παραδοσιακού ιστού (XML, URI) με την ήδη χρησιμοποιούμενη αρχιτεκτονική [1] [6].

Ο ιστός δεδομένων υποδηλώνει την ύπαρξη συνδέσμων για κάθε διακριτή οντότητα δεδομένων, σχηματίζοντας ένα τεράστιο γράφο. Αυτό επιτρέπει τη πλοήγηση του χρήστη σε σύνολα δεδομένων με συνάφεια, επίσης καθιστά δυνατή την ανακάλυψη και αναζήτηση πληροφοριών και δεδομένων από ειδικό λογισμικό, όπως ευφυείς πράκτορες [44] [45].

Οι βασικοί κανόνες για τα διασυνδεδεμένα δεδομένα τέθηκαν από τον Tim Berners-Lee και είναι οι ακόλουθοι [6] [46]:

1. Χρησιμοποίηση URIs (ή IRIs) για τον προσδιορισμό οντοτήτων
2. Να είναι προσβάσιμα τα URIs μέσω HTTP, ώστε να είναι προσπελάσιμα από ανθρώπους ή λογισμικό πρακτόρων
3. Χρησιμοποίηση προτύπων (RDF, RDF/XML, SPARQL) για κάθε οντότητα που συμμετέχει μέσω URIs
4. Ορισμός συνδέσμων σε άλλα συναφή URIs, ώστε να γίνει ανακάλυψη νέων σχετικών οντοτήτων

3.2.1 Ανοικτά διασυνδεδεμένα Δεδομένα – *Linked Open Data*

Η επιτυχία των διασυνδεδεμένων δεδομένων επέφερε μια νέα εποχή, αυτή των ανοικτών διασυνδεδεμένων δεδομένων – Linked Open Data (LOD) [6] [46]. Ο δημιουργός και πάλι έθεσε τους κανόνες για τον προσδιορισμό των LOD το 2010 [6]. Συμβολικά υπάρχουν 5 διαφορετικά επίπεδα (αστέρια) που μπορεί να διαβαθμιστεί η πληροφορία ως LOD.

Ακολουθούν τα 5 επίπεδα για να χαρακτηριστεί το LOD:

1. Διαθέστε τα δεδομένα σας στο Web, σε οποιαδήποτε μορφή, με ανοικτή άδεια (open licence)
2. Διαθέστε τα δεδομένα σας σε δομημένη μορφή/μηχανογνώσιμη, έστω και με κλειστό πρότυπο
3. Χρησιμοποιήστε ανοικτά μορφότυπα (π.χ. cvs)
4. Χρησιμοποιήστε HTTP URIs, για να μπορούν οι άλλοι να αναφερθούν στα δεδομένα σας
5. Διασυνδέστε τα δεδομένα σας με άλλα, τρίτων, για την ανακάλυψη νέων δεδομένων

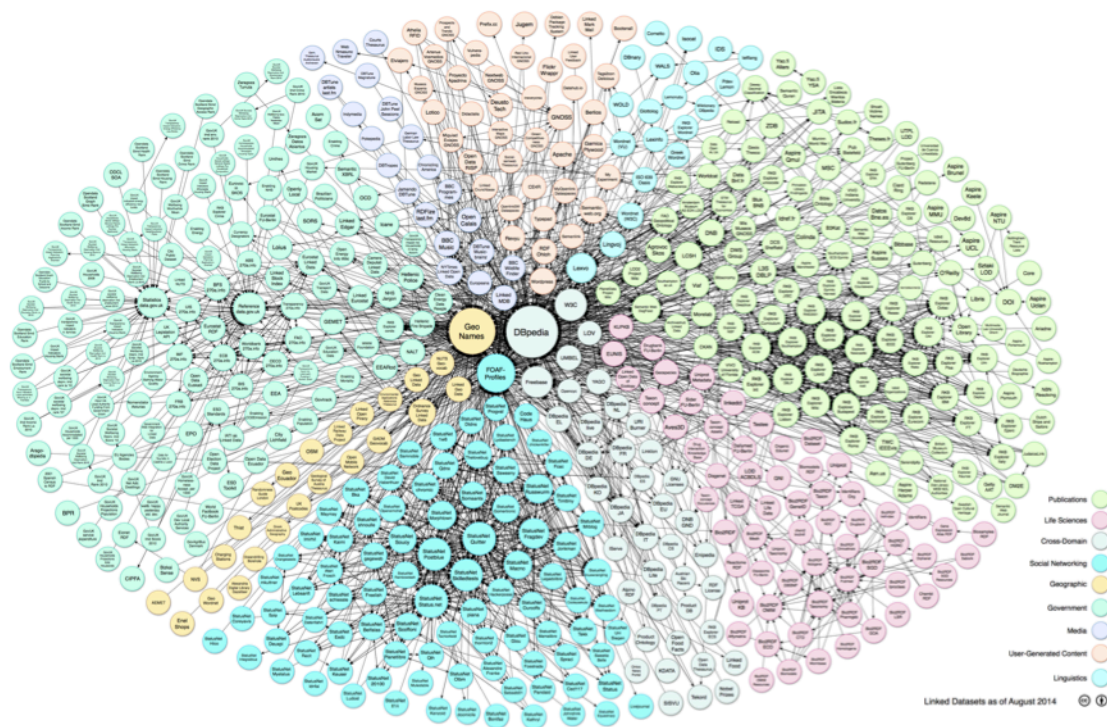
Μια πληροφορία γίνεται ισχυρότερη, όσο περισσότερα επίπεδα ακολουθεί [6].

Η εξάπλωση του LOD, υποστηρίχθηκε αρχικά απο το δημόσιο τομέα, τις κυβερνητικές οργανώσεις και μετέπειτα απο τις επιχειρήσεις. Για το δημόσιο τομέα είναι ένα άλμα και προωθεί τη διαφάνεια, την ηλεκτρονική διακυβέρνηση, την παιδεία, την οικονομία και τον πολιτισμό. Υπάρχουν έρευνες που αποδεικνύουν την οικονομική βελτίωση απο την χρήση των LOD [47].

Στον Ελλαδικό χώρο, το δημόσιο συμμετέχει ενεργά στο LOD και παλαιότερα υπήρξε και μια σημαντική συνιστώσα αυτού [48] [49] [50]. Η καίρια σημασία των LOD, έχει αναγνωριστεί και απο το κράτος και έχει θεσμοθετηθεί [50] [51] [52]. Χαρακτηριστικά παραδείγματα αποτελούν η Δι@ύγεια και η ΥπερΔι@ύγεια [51].

Οι επιχειρήσεις, υποστηρίζουν τα ανοικτά δεδομένα για ποικίλους λόγους, όπως για καλύτερα αποτελέσματα search engine optimization (SEO) [94], για εταιρική προβολή του brandname κ.α.

Το μέγεθος του LOD, φαίνεται στην παρακάτω εικόνα και δημιουργήθηκε με τη βοήθεια του LDSpider Crawling [53] [54] που είναι σε θέση να ανακαλύπτει οντολογίες στο παγκόσμιο ιστό.



Εικόνα 3: Linked Open Data (LOD) Community, Αύγουστος 2014

Η ραγδαία εξέλιξη του LOD, οδήγησε στη δημιουργία ενός νέου ιδρύματος του Open Data Institute (ODI) [55].

Το ODI ιδρύθηκε από τον εφευρέτη του Web και τον Sir Nigel Shadbolt με σκοπό να προάγει την κουλτούρα του LOD στην εκπαίδευση, στην υγεία, στην οικονομία και στην κοινωνία. Απαρτίζεται από δεκάδες συνεργάτες/χρήστες ανά τον κόσμο. Η εκπαίδευση, η κοινωνία ακόμη και η οικονομία μιας χώρας μπορεί να ωφεληθεί σημαντικά από τη χρήση των ανοικτών δεδομένων [47]. Μπορεί να μειώσει σημαντικά τη γραφειοκρατία μιας χώρας, δίνοντας όλες τις απαραίτητες πληροφορίες μέσω της συσχέτισης των ανοικτών δεδομένων. Ο συνδυασμός ανοικτών δεδομένων και ευφώνων πρακτόρων στο σημασιολογικό ιστό μπορεί να συμμετέχει στην έγκυρη πληροφόρηση και διαμόρφωση της κουλτούρας μιας χώρας, καθώς και στην ενίσχυση της δημοκρατίας εντός της χώρας.

Η Ελλάδα συμμετέχει ενεργά στο χώρο για τα ανοικτά διασυνδεδεμένα δεδομένα [56] [57]. Στις 21 Φεβρουαρίου υπάρχει ο εορτασμός της Παγκόσμιας Ημέρας Ανοικτών Δεδομένων, όπου διεξάγονται ημερίδες σχετικά με τις πληροφορίες του LOD ανά τον κόσμο [58].

4

Οντολογία

Η ανάγκη για αναπαράσταση της γνώσης, οδήγησε στη μοντελοποίηση της πληροφορίας σε κάθε γνωστικό τομέα με διαφορετική μέθοδο κάθε φορά παρόλο που επρόκειτο για την ίδια πληροφορία. Λύση στην ελαχιστοποίηση των μοντέλων της πληροφορίας έδωσαν οι οντολογίες.

4.1 Οντολογία σημασιολογικού ιστού – Web ontology

Η οντολογία βοηθά στη μοντελοποίηση της γνώσης και στη διαλειτουργικότητα μεταξύ των συστημάτων που αναπαριστούν γνώση. Οι οντολογίες ως δομικό υλικό, επιτρέπουν το διαμοιρασμό της γνώσης και την επαναχρησιμοποίηση της μεταξύ ομογενών και ετερογενών πληροφορικών συστημάτων [1] [2].

Για να αποδώσουμε τον ορισμό της έννοιας της οντολογίας, θα πρέπει πρώτα να επιλέξουμε την οπτική γωνία που μας ενδιαφέρει, διότι πρόκειται περί αμφισημίας.

Από το πρίσμα της φιλοσοφικής πλευράς, η Οντολογία (με κεφαλαίο Ο) ορίζεται ως η επιστήμη που μελετά τη φύση, την ύπαρξη, τα όντα και τις σχέσεις αυτών. Ειδικότερα ασχολείται με τον κλάδο της μεταφυσικής που έχει ως στόχο την περιγραφή των αληθινών πραγμάτων (όντων, φύσης) με γενικούς όρους και τον προσδιορισμό αυτών [2].

Στην αρχαιότητα ακόμη οι αρχαίοι Έλληνες είχαν ασχοληθεί με το ερώτημα: “Ποια είναι η ουσία των πραγμάτων μέσω των διαφόρων αλλαγών;”. Απαντήσεις δόθηκαν από διάφορους φιλόσοφους της αρχαιότητας.

Γλωσσολογικά η οντολογία είναι συνώνυμο της μεταφυσικής και αποδίδεται στον Αριστοτέλη, αρχικά αναφέρθηκε στο έργο «Λογικά ή Όργανον». Ο Αριστοτέλης στο έργο του, περιέγραψε τις διαφορετικές μορφές ύπαρξης ενός διακριτού συστήματος κατηγοριών (σχέση, ύλη, ποιότητα, συναίσθημα, ποσότητα, ενέργεια, χώρος, χρόνος) έτσι ώστε να κατηγοριοποιήσει οτιδήποτε υπάρχει στον κόσμο.

Υπήρξαν και διαφορετικές εκδοχές της οντολογίας στον σύγχρονο κόσμο, με κύριο εκπρόσωπο το Emmanuel Kant (1724-1804), ο οποίος πρόσθεσε πως η αίσθηση των πραγμάτων δεν καθορίζεται απόλυτα μόνο από τα πράγματα αλλά και από τη συμβολή όποιου τα αντιλαμβάνεται και τα κατανοεί.

Για την επιστήμη των υπολογιστών και δη για το σημασιολογικό ιστό υπάρχει πλήθος ορισμών, έχουν επικρατήσει όμως δύο (2) ως οι πιο δημοφιλείς:

Ο πρώτος ορισμός αναφέρει την οντολογία ως ένα τυπικό, σαφή προσδιορισμό μιας διαμοιρασμένης εννοιολογικής αναπαράστασης (conceptualization) [59].

Ο δεύτερος και πιο βελτιωμένος ορισμός, δίνεται ως μία ρητή και τυπική προδιαγραφή μιας επίνοιας [2].

Οι οντολογίες μπορούν να συσχετιστούν μεταξύ τους (ontology mapping), δημιουργώντας ένα πλέγμα συνδεδεμένων πληροφοριών. Πρόκειται για τη διαδικασία αντιστοίχισης πεδίων μεταξύ οντολογιών [2]. Έχοντας μια δική μας οντολογία, μπορούμε να την συσχετίσουμε με μία άλλη οντολογία που εμπεριέχει την υπάρχουσα (ισχυρισμός import). Υπάρχουν αρκετοί τρόποι για την υλοποίηση αυτή της αντιστοίχισης, μια πρόταση είναι το πλαίσιο SPARQL–RW, αποτελεί μια προσπάθεια για την διαφανή προς τον τελικό χρήστη, αντιστοίχιση των ομότιμων οντολογιών [60].

4.1.1 Γλώσσες οντολογιών

Η ερευνητική προσπάθεια για την δημιουργία γλωσσών καταλλήλων για την μοντελοποίηση και περιγραφή οντολογιών άρχισε σχεδόν ταυτόχρονα (~1990) με την εισαγωγή του όρου

οντολογία στα πεδία των Knowledge Engineering, Natural-language Processing Knowledge Representation κ.α. Έκτοτε προτάθηκαν διάφορες γλώσσες οι οποίες είχαν σύντομη ιστορία ή που εξελίχθηκαν και υπάρχουν έως σήμερα με κάποια μορφή [62].

Ακολουθεί μία παράθεση των κυριοτέρων από αυτές:

1. 1992 KIF, ONTOLINGUA, LOOM
2. 1993 OCML
3. 1995 FLogic
4. 1996 SHOE
5. 1997 OKBC protocol (Open Knowledge Base Connectivity)
6. 1999 XOL
7. 1999 RDF
8. 1999 DAML
9. 2001 OIL
10. 2002 DAML+OIL
11. 2004 OWL
12. 2004 RDF Schema
13. 2007 OWL2

Από όλες αυτές τις γλώσσες, οι SHOE, XOL, RDF, RDFS, DAML, OIL, DAML+OIL, OWL έχουν τρόπο σύνταξης βασισμένο στο συντακτικό της XML και θα παρουσιαστούν εκτενέστερα γιατί η διαχρονική εξέλιξή τους είναι συνδεδεμένη με την εξέλιξη του σημασιολογικού ιστού.

Επιπλέον οι γλώσσες RDF, RDFS, και OWL έχουν ιδιαίτερη σημασία, δεδομένου ότι έχουν αποτελέσει και σχετικά "Recommendation" του W3C, σε ότι αφορά την πορεία προς την υλοποίηση του σημασιολογικού ιστού [65] [66].

SHOE: Simple HTML Ontology Extensions [63] η γλώσσα SHOE δημιουργήθηκε ως μία προέκταση της HTML. Χρησιμοποιεί tags διαφορετικά από αυτά της HTML και ο κώδικάς της συνυπάρχει μέσα στο ίδιο φυσικό αρχείο μαζί με τον κώδικα HTML. Με τον τρόπο αυτό επιτυγχάνεται η παρεμβολή οντολογιών μέσα στο HTML έγγραφο. Η SHOE συνδυάζει frames και κανόνες. Μπορεί να αναπαραστήσει έννοιες (concepts), taxonomies of concepts, δυαδικές σχέσεις, στιγμιότυπα και κανόνες για την εξαγωγή συμπερασμάτων.

XOL: Σχεδιάστηκε ως η XML εκδοχή του OKBC-Lite, μικρού υποσυνόλου "primitives" (στοιχειωδών οντοτήτων) του πρωτοκόλλου OKBC. Πολύ αυστηρή γλώσσα που μπορεί να αναπαραστήσει μόνο έννοιες (concepts), taxonomies of concepts και δυαδικές σχέσεις [62].

RDF/RDF Schema: Resource Description Framework Η RDF είναι ένα μοντέλο δεδομένων [64]. Είναι μία γλώσσα για την αναπαράσταση πληροφοριών που αφορούν πόρους του παγκόσμιου ιστού. Προορίζεται κυρίως για την αναπαράσταση μεταδεδομένων. Γενικεύοντας την έννοια του πόρου του παγκόσμιου ιστού, η RDF μπορεί να αναπαραστήσει πληροφορίες για αντικείμενα του παγκόσμιου ιστού ακόμα και αν αυτά δεν είναι άμεσα προσπελάσιμα. Η γλώσσα RDF είναι κατάλληλη για τις περιπτώσεις όπου η πληροφορία πρέπει να επεξεργαστεί ή ανταλλαχθεί από μηχανές, σε αντίθεση π.χ. με την HTML που είναι κατάλληλη για την παρουσίαση της πληροφορίας σε ανθρώπους [65]. Η RDF Schema παρέχει ένα λεξιλόγιο για τον τρόπο που θα συγκροτηθεί το μοντέλο με τα δεδομένα της RDF. Η RDF Schema είναι μία σημασιολογική προέκταση της RDF. Παρέχει το μηχανισμό για την περιγραφή ομάδων με σχετικούς μεταξύ τους πόρους καθώς και την περιγραφή αυτής της σχέσης [66].

DAML, OIL, DAML+OIL

DAML: DARPA Agent Markup Language [62]. Η γλώσσα DAML δημιουργήθηκε στο πλαίσιο ερευνητικού προγράμματος του US Defense Advanced Research Projects Agency (DARPA) για την δημιουργία της υποδομής που θα επέτρεπε την επεξεργασία της πληροφορίας του παγκόσμιου ιστού με τρόπο κατάλληλο για επεξεργασία από μηχανές, με την χρήση οντολογιών για την αναπαράσταση των αντικειμένων και των μεταξύ τους σχέσεων. Σχεδιάστηκε ως επέκταση των XML και RDF.

OIL: Ontology Inference Layer [67]

Η γλώσσα OIL ήταν η ευρωπαϊκή προσπάθεια αντίστοιχη της αμερικανικής DAML. και έγινε στο πλαίσιο του προγράμματος "On-To-Knowledge" (the European EU-IST project No. 10132) [68].

DAML+OIL: Η γλώσσα DAML+OIL βασίζεται σε προηγούμενα πρότυπα του W3C όπως αυτά των RDF και RDF Schema, σε μία προσπάθεια να συνδυάσει την DAML και την OIL και να προεκτείνει αυτές τις γλώσσες με πλουσιότερα "primitives" (στοιχειώδεις οντότητες).

OWL / OWL2: Ontology Web Language

Η γλώσσα OWL επεκτείνει την ικανότητα των XML, RDF και RDF Schema στην αναπαράσταση των οντολογιών του παγκόσμιου ιστού. Έρχεται σαν αναθεώρηση της DAML+OIL και έχει τρεις διαφορετικές εκδοχές - υπογλώσσες, τις OWL Lite, OWL DL, και OWL Full με αυξανόμενη ικανότητα έκφρασης οντολογιών, αντιστοίχως [1].

4.1.2 Δομικά στοιχεία οντολογίας OWL

Η OWL είναι μία γλώσσα οντολογιών που είναι βασισμένη στις γνωστές μας RDF και RDFS, τα αρχικά της σημαίνουν Web Ontology Language και δημιουργήθηκε, όπως και η RDF για να μπορεί να ερμηνεύεται από τους υπολογιστές [1]. Η διαφορά της με την RDF είναι ότι πρόκειται για μία πολύ πιο πλούσια γλώσσα με πολύ μεγαλύτερο λεξιλόγιο και με πολύ καλύτερη δυνατότητα ερμηνείας από τους υπολογιστές.

Υπάρχουν τρεις εκδόσεις της OWL η OWL Lite, η OWL DL (περιέχει την Lite) και η OWL Full (περιέχει την DL) [1].

Τα βασικά χαρακτηριστικά της σύνταξης μια OWL οντολογίας είναι:

- Η κεφαλίδα
- Στοιχεία κλάσεων
- Στοιχεία ιδιοτήτων
- Τα στιγμιότυπα των κλάσεων

Κεφαλίδα (header)

Η κεφαλίδα είναι ουσιαστικά η ρίζα της οντολογίας και ορίζεται ως ένα στοιχείο `rdf:RDF` το οποίο καθορίζει έναν αριθμό από χώρους ονομάτων. Οι χώροι ονομάτων που χρησιμοποιούνται στα `rdf:RDF` tags υπάρχουν για να προσδιορίσουν το λεξιλόγιο άλλων tags που θα χρησιμοποιηθούν αργότερα μέσα στην οντολογία. Για να γίνει κατανοητό θα χρησιμοποιήσουμε ένα παράδειγμα κεφαλίδας `rdf:RDF` όπως φαίνεται παρακάτω:

```
<rdf:RDF
```

```
  xmlns="http://www.ourserver.com/ontology/university.owl#"
```

```
  xml:base="http://www.ourserver.com /ontology/university.owl"
```

```
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
```

```
  xmlns:owl="http://www.w3.org/2002/07/owl#"
```

```

xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<owl:Ontology rdf:about="http://www.ourserver.com/ontology/university.owl">
  <rdfs:label rdf:datatype="&xsd:string">MSc Web Intelligence Ontology</rdfs:label>
  <owl:versionInfo rdf:datatype="&xsd;decimal">1.1</owl:versionInfo>
  <rdfs:comment rdf:datatype="&xsd:string">A.T.E.I. Thessalonikis - IT Department -
Master of Science (MSc) in Web Intelligence Ontology</rdfs:comment>
  <rdfs:isDefinedBy
rdf:datatype="&xsd:string">http://dbpedia.org/resource/University</rdfs:isDefinedBy>
  <rdfs:seeAlso
rdf:datatype="&xsd:anyURI">http://el.dbpedia.org/ontology/University</rdfs:seeAlso>
</owl:Ontology>

```

Στο παράδειγμά μας τα `xmlns="http://www.ourserver.com/ontology/university.owl#"` και `xmlns:base="http://www.ourserver.com/ontology/university.owl#"` αποτελούν τους προεπιλεγμένους χώρους ονομάτων μας.

Η οντολογία ξεκινάει με μια συλλογή ισχυρισμών (assertions) για λόγους ευελιξίας. Οι ισχυρισμοί ομαδοποιούνται σε ένα στοιχείο `owl:Ontology` το οποίο περιέχει σχόλια, έλεγχο εκδόσεων και προσθήκη άλλων οντολογιών με τον ισχυρισμό `import`. Στη τρέχουσα έκδοση δεν χρησιμοποιήσαμε άλλες οντολογίες.

Στοιχεία Κλάσεων (Classes)

Μια OWL κλάση εκφράζεται σε RDF/XML μέσω ενός στοιχείου `owl:Class`. Υπάρχουν δύο προκαθορισμένες κλάσεις, οι `owl:Thing` και `owl:Nothing`. Η πρώτη είναι η πιο γενική κλάση, περιέχει τα πάντα και ονομάζεται υπερκλάση. Αντιθέτως η τελευταία ονομάζεται κενή κλάση και είναι υποκλάση της πρώτης.

Η `owl:Class` ετικέτα περιέχει την δήλωση ενός `rdf:ID` ή `rdf:about` στοιχείου το οποίο προσδιορίζει τοπικά το όνομα της κλάσης σε αυτό το αρχείο – οντολογία.

```

<owl:Class rdf:about="&base;Courses">
</owl:Class>

```

```

<owl:Class rdf:ID="Modules">
  <rdfs:subClassOf rdf:resource="&university;Courses"/>
  <rdfs:comment rdf:datatype="&xsd:string">Master of Science (MSc) in Web
  Intelligence. There are 2 semesters with total 10 Courses and 1 semester for MSc
  Thesis</rdfs:comment>
</owl:Class>

```

Διάφορα πρόσθετα στοιχεία είναι ιδιαίτερα σημαντικά για τον προσδιορισμό της κλάσης, ένα `rdfs:subClassOf` στοιχείο επιτρέπει μια αφηρημένη ομαδοποίηση να υποδιαιρείται σε μικρότερες ομάδες. Αυτό δείχνει ότι όλα τα μέλη της κλάσης που δηλώνονται είναι και μέλη της υπερκλάσης που ορίζεται από το `rdf:resource`.

Όπως φαίνεται και στο παράδειγμα το *Modules* ορίζεται σαν υποκλάση του *Courses* που σημαίνει ότι όλα τα *Modules* είναι ταυτόχρονα και *Courses*.

Ένα `owl:oneOf` στοιχείο συνδυασμένο με το `rdfs:subClassOf` μπορεί να χρησιμοποιηθεί για να ορίσεις μια κλάση απορρυθμίζοντας τα στιγμιότυπα που ανήκουν σε αυτήν την κλάση.

Επίσης το στοιχείο `owl:equivalentClass` δηλώνει ότι αυτή η κλάση είναι η ίδια με μία άλλη. Παράδειγμα: (Η *Lessons* είναι η ίδια με την *Modules*)

```

<owl:Class rdf:ID="Lessons">
  <owl:equivalentClass rdf:resource="#Modules"/>
</owl:Class>

```

Στοιχεία ιδιοτήτων (Properties)

Φυσικά δεν θα μπορούσαμε να δημιουργήσουμε μια ουσιαστική οντολογία απλά συσχετίζοντας κλάσεις. Έτσι χρησιμοποιώντας τις ιδιότητες των οντολογιών προβάλλουμε στοιχεία και συγκεκριμένα γεγονότα σχετικά με τα μέλη των κλάσεων. Πρόκειται για μία δυαδική σχέση και εκφράζεται με δύο τύπους ιδιοτήτων:

- Ιδιότητες τύπου δεδομένων (*datatype properties*)

Ένα `owl:DatatypeProperty` στοιχείο εκφράζει την σχέση μεταξύ ενός στιγμιότυπου και μιας τιμής δεδομένου. Όπως φαίνεται και στο παράδειγμα η ιδιότητα `courseECTS` δηλώνεται με μια τιμή για το στιγμιότυπο *Modules*

```

<owl:DatatypeProperty rdf:about="&university;courseECTS">
  <rdfs:comment rdf:datatype="&xsd:string">Course ECTS Units</rdfs:comment>
  <rdfs:domain rdf:resource="&university;Modules"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

```

- Ιδιότητες αντικειμένου (*object properties*)

Ένα στοιχείο owl:ObjectProperty εκφράζει την σχέση μεταξύ δυο στιγμιοτύπων. Όπως φαίνεται στο παράδειγμα ο ορισμός του ObjectProperty σαν hasPublication χρησιμοποιείται για να δηλωθεί μια συσχέτιση ενός Lecturer με ένα Publications.

```

<owl:ObjectProperty rdf:about="&university;hasPublication">
  <rdfs:comment rdf:datatype="&xsd:string">Publications for lecturer</rdfs:comment>
  <rdfs:domain rdf:resource="&university;Lecturer"/>
  <rdfs:range rdf:resource="&university;Publications"/>
</owl:ObjectProperty>

```

Οι σχέσεις μεταξύ των ιδιοτήτων εμπεριέχουν και πολλούς άλλους ισχυρισμούς, μεταξύ των οποίων είναι περιορισμοί ιδιοτήτων, ειδικές ιδιότητες, λογικοί συνδυασμοί και απαριθμήσεις. Οι ιδιότητες μπορούν να συνδέονται και με τύπους δεδομένων της XML Schema στους οποίους περιλαμβάνονται και οι γνωστοί integer, string, boolean, time και date.

Στιγμιότυπα (Instances - Individuals)

Τα στιγμιότυπα των κλάσεων καθορίζονται από τον προσδιορισμό της κλάσης στην οποία ανήκουν. Πρόκειται για ενθυλάκωση συγκεκριμένων οντοτήτων στην οντολογία. Για παράδειγμα η παρακάτω δήλωση ορίζει ένα στιγμιότυπο με ID M101 της κλάσης *Modules*.

```

<Modules rdf:ID="M101" />

```

Η επόμενη δήλωση ορίζει ένα στιγμιότυπο με ID *Stamatis I*:

```

<owl:NamedIndividual rdf:about="&university;Stamatis I">
  <rdf:type rdf:resource="&university;Publications"/>
  <publicationTitle rdf:datatype="&xsd:string">...</publicationTitle>

```

</owl:NamedIndividual>

Σημαντικό χαρακτηριστικό της OWL είναι ότι δεν υιοθετεί την υπόθεση μοναδικών ονομάτων (unique-names assumption), επειδή δύο στιγμιότυπα έχουν διαφορετικό όνομα ή διαφορετικό ID δεν συνεπάγεται ότι είναι δυο διαφορετικά μεμονωμένα στοιχεία [1].

Σύμφωνα με τους Noy και McGuinness [18], οι έννοιες (concepts) μιας οντολογίας πρέπει να σχετίζονται με τα αντικείμενα και τις σχέσεις του τομέα που μας ενδιαφέρει. Αυτά είναι πιθανότερο να είναι ουσιαστικά (αντικείμενα) ή ρήματα (σχέσεις), μέσα σε προτάσεις που περιγράφουν τον τομέα ενδιαφέροντός μας [6]. Γι' αυτό το λόγο πρέπει να αποφασίσουμε πως θα χρησιμοποιήσουμε την οντολογία και πόσο λεπτομερής ή γενική θα είναι. Θα πρέπει να αποφασίσουμε ποια βιώσιμη εναλλακτική λύση θα δουλέψει καλύτερα.

4.1.3 Ορθότητα οντολογίας

Η οντολογία που αναπτύχθηκε, ελέγχθηκε με τον αυτοματοποιημένο συλλογιστή (reasoner) Pellet [2] [69]. Ο αυτοματοποιημένος συλλογιστής είναι ένα εργαλείο που χρησιμοποιείται για τον έλεγχο ορθότητας και την εξαγωγή επιπρόσθετης πληροφορίας από την οντολογία. Ουσιαστικά χρησιμοποιεί τη γνώση που έχει ορισθεί από το χρήστη και μέσω αυτής σε συνδυασμό με τα αξιώματα, εξάγει νέα γνώση που βασίζεται στα αξιώματα που έχουν δοθεί.

Ένα απλό παράδειγμα συμπερασμού νέας γνώσης από τα παρακάτω αξιώματα:

<i>Essay</i>	<i>subClassOf</i>	<i>Courses</i>
<i>SemanticWeb_2</i>	<i>type</i>	<i>Essay</i>

Με τη βοήθεια του reasoner, μπορεί να συμπεραθεί πως το *SemanticWeb_2* είναι τύπου *Courses*.

Ο αυτοματοποιημένος συλλογιστής Pellet χρησιμοποιείται για οντολογίες OWL-DL και θεωρείται ένας από τους πιο αξιόπιστους. Τα κύρια χαρακτηριστικά του είναι:

1. Είναι συμβατός με όλα τα δομικά συστατικά της OWL-DL
2. Είναι ΕΛ/ΛΑΚ (open source) λογισμικό και είναι γραμμένος σε java. Χάρη αυτού του χαρακτηριστικού μπορεί να ενσωματωθεί ως πρόσθετο εργαλείο σε συντάκτες οντολογιών όπως Protégé και Jena.

4.2 Οντολογία μεταπτυχιακού προγράμματος σπουδών

Η οντολογία του μεταπτυχιακού προγράμματος σπουδών έχει σχεδιαστεί στο περιβάλλον protégé, μια εφαρμογή συγγραφής και ανάπτυξης OWL οντολογιών. Έχει ελεγχθεί από τον ενσωματωμένο reasoner pellet [69].

Δίνεται το αρχείο της οντολογίας university.owl σε μορφή RAW δίχως τα στιγμιότυπα:

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
  <!ENTITY university "http://urlOfOntology/ontology/university.owl#" >
] >

<rdf:RDF xmlns="http://urlOfOntology/ontology/university.owl#"
  xml:base="http://urlOfOntology/ontology/university.owl"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:university="http://urlOfOntology/ontology/university.owl#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <owl:Ontology rdf:about="http://urlOfOntology/ontology/university.owl">
    <rdfs:label rdf:datatype="&xsd:string">MSc Web Intelligence Ontology</rdfs:label>
    <owl:versionInfo rdf:datatype="&xsd:decimal">1.1</owl:versionInfo>
    <rdfs:comment rdf:datatype="&xsd:string">A.T.E.I. Thessalonikis - IT Department -
Master of Science (MSc) in Web Intelligence Ontology</rdfs:comment>
    <rdfs:isDefinedBy
rdf:datatype="&xsd:string">http://dbpedia.org/resource/University</rdfs:isDefinedBy>
    <rdfs:seeAlso
rdf:datatype="&xsd:anyURI">http://el.dbpedia.org/ontology/University</rdfs:seeAlso>
  </owl:Ontology>
```

```

<owl:ObjectProperty rdf:about="&university;hasPublication">
  <rdfs:comment rdf:datatype="&xsd:string">Publications for lecturer</rdfs:comment>
  <rdfs:domain rdf:resource="&university;Lecturer"/>
  <rdfs:range rdf:resource="&university;Publications"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&university;hasSucceed">
  <rdfs:comment rdf:datatype="&xsd:string">Student has succeed in MSc Thesis</rdfs:comment>
  <rdfs:domain rdf:resource="&university;Student"/>
  <rdfs:range rdf:resource="&university;Thesis"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&university;isVisitorLecturer">
  <rdfs:comment rdf:datatype="&xsd:string">Lecturer is visitor lecturer</rdfs:comment>
  <rdfs:range rdf:resource="&university;Courses"/>
  <rdfs:domain rdf:resource="&university;Lecturer"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&university;offeringThesis">
  <rdfs:domain rdf:resource="&university;Lecturer"/>
  <rdfs:range rdf:resource="&university;Thesis"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&university;studies">
  <rdfs:comment rdf:datatype="&xsd:string">Student studies a course</rdfs:comment>
  <rdfs:range rdf:resource="&university;Courses"/>
  <rdfs:domain rdf:resource="&university;Student"/>
  <rdfs:subPropertyOf rdf:resource="&owl;topObjectProperty"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&university;teaches">
  <rdfs:comment rdf:datatype="&xsd:string">Lecturer teaches courses</rdfs:comment>
  <rdfs:range rdf:resource="&university;Courses"/>
  <rdfs:domain rdf:resource="&university;Lecturer"/>

```

</owl:ObjectProperty>

<owl:ObjectProperty rdf:about="&university;workingThesis">

<rdfs:domain rdf:resource="&university;Student"/>

<rdfs:range rdf:resource="&university;Thesis"/>

</owl:ObjectProperty>

<owl:DatatypeProperty rdf:about="&university;courseContent">

<rdfs:comment rdf:datatype="&xsd:string">Course Description</rdfs:comment>

<rdfs:domain rdf:resource="&university;Modules"/>

<rdfs:range rdf:resource="&xsd:string"/>

</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;courseECTS">

<rdfs:comment rdf:datatype="&xsd:string">Course ECTS Units</rdfs:comment>

<rdfs:domain rdf:resource="&university;Modules"/>

<rdfs:range rdf:resource="&xsd;integer"/>

</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;courseGoal">

<rdfs:comment rdf:datatype="&xsd:string">Academics Learning goals</rdfs:comment>

<rdfs:domain rdf:resource="&university;Modules"/>

<rdfs:range rdf:resource="&xsd:string"/>

</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;courseID">

<rdfs:comment rdf:datatype="&xsd:string">Course unique ID</rdfs:comment>

<rdfs:domain rdf:resource="&university;Modules"/>

<rdfs:range rdf:resource="&xsd:string"/>

</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;courseName">

<rdfs:comment rdf:datatype="&xsd:string">MSc Course Name</rdfs:comment>

<rdfs:domain rdf:resource="&university;Modules"/>


```

    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;courseReferencesBibliography">
    <rdfs:comment rdf:datatype="&xsd:string">Course Bibliography.</rdfs:comment>
    <rdfs:domain rdf:resource="&university;Modules"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;courseSemester">
    <rdfs:comment rdf:datatype="&xsd:string">course semester</rdfs:comment>
    <rdfs:domain rdf:resource="&university;Modules"/>
    <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;courseStudentEvaluation">
    <rdfs:domain rdf:resource="&university;Modules"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;courseTeaching">
    <rdfs:domain rdf:resource="&university;Modules"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;firstName">
    <rdfs:comment rdf:datatype="&xsd:string">First name of people</rdfs:comment>
    <rdfs:domain rdf:resource="&university;Person"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;foaf">
    <rdfs:comment rdf:datatype="&xsd:string">Friend of a Friend URI</rdfs:comment>
    <rdfs:domain rdf:resource="&university;Person"/>

```

```

    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;infoURI">
    <rdfs:domain rdf:resource="&university;Person"/>
    <rdfs:range rdf:resource="&xsd:anyURI"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;lastName">
    <rdfs:comment rdf:datatype="&xsd:string">Last or family name for people</rdfs:comment>
    <rdfs:domain rdf:resource="&university;Person"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;publicationDate">
    <rdfs:domain rdf:resource="&university;Publications"/>
    <rdfs:range rdf:resource="&xsd:dateTime"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;publicationScientificMagazine">
    <rdfs:domain rdf:resource="&university;Publications"/>
    <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;publicationConference">
    <rdfs:domain rdf:resource="&university;Publications"/>
    <rdfs:range rdf:resource="&xsd:boolean"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:about="&university;publicationTitle">
    <rdfs:comment rdf:datatype="&xsd:string">Lecturer&apos;s publication title.</rdfs:comment>
    <rdfs:domain rdf:resource="&university;Publications"/>
    <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

```

```
<owl:DatatypeProperty rdf:about="&university;publicationUrl">
  <rdfs:domain rdf:resource="&university;Publications"/>
  <rdfs:range rdf:resource="&xsd:anyURI"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="&university;staffID">
  <rdfs:comment rdf:datatype="&xsd:string">Unique staff ID, existed on moodle</rdfs:comment>
  <rdfs:domain rdf:resource="&university;Lecturer"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="&university;studentID">
  <rdfs:comment rdf:datatype="&xsd:string">Student's ID via registration
order</rdfs:comment>
  <rdfs:domain rdf:resource="&university;Student"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="&university;thesisCompleted">
  <rdfs:domain rdf:resource="&university;Thesis"/>
  <rdfs:range rdf:resource="&xsd;boolean"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="&university;thesisDate">
  <rdfs:domain rdf:resource="&university;Thesis"/>
  <rdfs:range rdf:resource="&xsd;dateTime"/>
</owl:DatatypeProperty>
```

```
<owl:DatatypeProperty rdf:about="&university;thesisECTS">
  <rdfs:domain rdf:resource="&university;Thesis"/>
  <rdfs:range rdf:resource="&xsd;integer"/>
</owl:DatatypeProperty>
```

```

<owl:DatatypeProperty rdf:about="&university;thesisName">
  <rdfs:comment rdf:datatype="&xsd:string">MSc thesis</rdfs:comment>
  <rdfs:domain rdf:resource="&university;Thesis"/>
  <rdfs:range rdf:resource="&xsd:string"/>
</owl:DatatypeProperty>

<owl:Class rdf:about="&university;Courses">
  <rdfs:subClassOf rdf:resource="&university;EducationInstitute"/>
  <rdfs:comment rdf:datatype="&xsd:string">The postgraduate programme “Master of Science (MSc) in Web Intelligence” provides advanced knowledge in sophisticated computer technologies related to Intelligent Web Applications. The MSc is designed to provide graduate-level education in Computational Intelligence and Intelligent Technologies for internet applications. Courses for Master of Science (MSc) in Web Intelligence</rdfs:comment>
</owl:Class>

<owl:Class rdf:about="&university;EducationInstitute">
  <owl:versionInfo rdf:datatype="&xsd:integer">1</owl:versionInfo>
  <rdfs:comment rdf:datatype="&xsd:string">A.T.E.I. of Thessaloniki, Department of Information Technology, Master of Science (MSc) in Web Intelligence</rdfs:comment>
  <rdfs:seeAlso
rdf:datatype="&xsd:anyURI">http://el.dbpedia.org/ontology/University</rdfs:seeAlso>
</owl:Class>

<owl:Class rdf:about="&university;Lecturer">
  <rdfs:subClassOf rdf:resource="&university;Person"/>
  <rdfs:comment rdf:datatype="&xsd:string">Lecturers on MSc Web Intelligence</rdfs:comment>
</owl:Class>

<owl:Class rdf:about="&university;Modules">
  <rdfs:subClassOf rdf:resource="&university;Courses"/>
  <rdfs:comment rdf:datatype="&xsd:string">Master of Science (MSc) in Web Intelligence. There are 2 semesters with total 10 Courses and 1 semester for MSc Thesis</rdfs:comment>
</owl:Class>

```

```

<owl:Class rdf:about="&university;Person">
  <rdfs:subClassOf rdf:resource="&university;EducationInstitute"/>
  <rdfs:comment rdf:datatype="&xsd:string">Lecturers And Students</rdfs:comment>
</owl:Class>

<owl:Class rdf:about="&university;Publications">
  <rdfs:subClassOf rdf:resource="&university;EducationInstitute"/>
  <rdfs:comment rdf:datatype="&xsd:string">Publications from lecturers</rdfs:comment>
</owl:Class>

<owl:Class rdf:about="&university;Student">
  <rdfs:subClassOf rdf:resource="&university;Person"/>
</owl:Class>

<owl:Class rdf:about="&university;Thesis">
  <rdfs:subClassOf rdf:resource="&university;Courses"/>
  <rdfs:comment rdf:datatype="&xsd:string">MSc Thesis Provided</rdfs:comment>
</owl:Class>

<rdf:Description>
  <rdf:type rdf:resource="&owl;AllDisjointClasses"/>
  <owl:members rdf:parseType="Collection">
    <rdf:Description rdf:about="&university;Lecturer"/>
    <rdf:Description rdf:about="&university;Modules"/>
    <rdf:Description rdf:about="&university;Student"/>
    <rdf:Description rdf:about="&university;Thesis"/>
  </owl:members>
</rdf:Description>
</rdf:RDF>

```

5

SPARQL

Η SPARQL (Simple Protocol and RDF Query Language) [70] [74] είναι η επίσημη γλώσσα ερωτημάτων που προσφέρει ανάκτηση πληροφοριών από RDF graphs [1]. Διαθέτει μεθόδους για την πρόσβαση στην πρωτογενή πληροφορία, έχει τη δυνατότητα εξαγωγής υπογραφημάτων καθώς και της δημιουργίας νέων γραφημάτων προερχόμενων από τις απαντήσεις συγκεκριμένων ερωτημάτων. Η γλώσσα SPARQL είναι προσανατολισμένη στα δεδομένα και δεν αποσκοπεί να εξάγει λογικά συμπεράσματα, ως είθισται, σε μια γενική γλώσσα προγραμματισμού [71]. Είναι άρρηκτα συνδεδεμένη με το σημασιολογικό ιστό. Τυποποιήθηκε από την Ομάδα Εργασίας για την Πρόσβαση RDF Δεδομένων (RDF Data Access Working Group) της W3C και το 2008 έγινε επίσημη σύσταση της W3C σχετικά με τη SPARQL 1.0. Αποτελεί εξέλιξη των rdfDB, RDQL, SeRQL και αποτελεί πρόταση του W3C [70].

Η SPARQL είναι για την RDF ότι ακριβώς είναι για τις σχεσιακές βάσεις δεδομένων η SQL ή για την XML η XQuery. Και επειδή ο Σημασιολογικός Ιστός βασίζεται στην αναπαράσταση γνώσης με RDF, η SPARQL και τα σχετιζόμενα με αυτήν πρωτόκολλα έχουν κυρίαρχη σημασία μέσα σ' αυτόν [70] [71].

Τα SPARQL ερωτήματα βασίζονται σε μοτίβα τριάδων (triple patterns). Ένα μοτίβο τριάδας είναι το ίδιο με μία RDF τριάδα, με τη μόνη διαφορά ότι ένας ή περισσότεροι από τους πόρους-συνιστώσες του είναι μεταβλητές. Μία μεταβλητή συμβολίζεται ?name ή \$name, όπου name είναι το όνομά της. Για τις μεταβλητές το πρόθεμα ? ή \$ είναι το ίδιο.

Το SPARQL Endpoint που εκτελεί ένα ερώτημα αναζητά από όλους τους πόρους, εκείνους που επαληθεύουν τα μοτίβα τριάδων του ερωτήματος, σύμφωνα με τις RDF προτάσεις που υπάρχουν στη βάση γνώσης με την οποία η SPARQL μηχανή είναι συνδεδεμένη [72].

Η SPARQL έχει πολλές ομοιότητες με την SQL (ANSI SQL '92). Ένα ερώτημα σε SPARQL μπορεί να περιέχει συζεύξεις, διαζεύξεις, προαιρετικούς περιορισμούς, περιορισμούς ως προς το πλήθος των αποτελεσμάτων και γενικά σχεδόν όλα τα βασικά δομικά στοιχεία που έχει και η SQL [73].

Υπάρχουν υλοποιήσεις της SPARQL για πολλές γλώσσες προγραμματισμού και εργαλεία όπως για:

- τη σύνδεση με ένα άκρο/πύλη (διεπαφή) SPARQL και την ημιαυτόματη κατασκευή ενός ερωτήματος
- τη μετάφραση των ερωτημάτων SPARQL σε άλλες γλώσσες ερωτημάτων (π.χ. SQL, XQuery, κλπ.).

Η SPARQL 1.1 ορίστηκε ως W3C Recommendation στις 21 Μαρτίου 2013 [74].

Επιστέγασμα η περιγραφή του δημιουργού της:

“Το να προσπαθήσουμε να χρησιμοποιήσουμε το Σημασιολογικό Ιστό χωρίς τη SPARQL είναι σαν να προσπαθούμε να χρησιμοποιήσουμε μια σχεσιακή βάση δεδομένων χωρίς SQL” [75].

5.1 Δομή SPARQL

Η δομή της SPARQL 1.1 μπορεί να αποτυπωθεί σε 4 διαφορετικές κατηγορίες [74]:

- I. SELECT, είναι πανομοιότυπη με το ερώτημα SELECT των γλωσσών SQL 4^{ης} γενιάς με βασική διαφορά ότι αντί για εγγραφές και πίνακες κρύβονται από πίσω τριπλέτες και γράφοι. Η σύνταξη της SELECT έχει ως εξής [3] [4]:

```
SELECT <?subject ?predicate ?object ?variable1 ... $variableN >
      WHERE { ?subject ?predicate ?object.
              ...
            }
```

Η τριπλέτα στο WHERE, ?subject ?predicate ?object, αποτελεί τη συνθήκη που θα πρέπει να είναι αληθής για την επιτυχή εκτέλεση του ερωτήματος.

Η γενική της μορφή είναι:

```
                <URI>        <URI>        <URI>.
    ή
    ?subject        ?predicate        ?object.
```

Η τριπλέτα τερματίζεται με τελεία (.). Μπορεί να συνδυαστεί με άλλες τριπλέτες για να δημιουργηθούν πιο περίπλοκα ερωτήματα, προϋπόθεση να είναι αληθής κάθε μία τριπλέτα, εκτός και εάν είναι προαιρετική (optional).

```
SELECT *
WHERE {
    ?s msc:firstName ?firstname.
    ?s msc:lastName ?lastname.
    ?s msc:studies ?id.
OPTIONAL { ?s msc:workingThesis ?thesis }
}
```

Όταν περισσότερες από 2 τριπλέτες έχουν κοινό ?subject μπορούν να το παραλείψουμε κάνοντας χρήση του ερωτηματικού (;).

```
SELECT *
WHERE {
    ?s msc:firstName ?firstname;
    msc:lastName ?lastname;
    msc:studies ?id.
OPTIONAL { ?s msc:workingThesis ?thesis }
}
```

Σε ερωτήματα με μεγάλα URIs μπορούμε για λόγους ευκολίας να χρησιμοποιήσουμε τα λεγόμενα PREFIX,

```
PREFIX msc: <http://urlOfOntology/ontology.owl#>
SELECT str(?e) as ?course ?ECST ?semester
```



```

WHERE {
    ?s msc:courseECTS ?ECST.
    ?s msc:courseName ?e.
    ?s msc:courseSemester ?semester .
}

```

αντί των κανονικών URIs:

```

SELECT str(?e) as ?course ?ECST ?semester
WHERE {
    ?s <http://urlOfOntology/ontology.owl#courseECTS> ?ECST.
    ?s <http://urlOfOntology/ontology.owl#courseName> ?e.
    ?s <http://urlOfOntology/ontology.owl#courseSemester> ?semester.
}

```

Υπάρχουν πολλά ακόμη χαρακτηριστικά για τη δομή SELECT WHERE όπως unions, regular expressions, filters, bind, from named, order κ.α. Πολύ καλό βοήθημα για το τρόπο σύνταξης της είναι το iSPARQL QBE, που μας εμφανίζει τη δομή κάθε εμπλεκόμενου χαρακτηριστικού, όπως περιγράφεται παρακάτω στο κεφάλαιο 5 (§ 5.2).

- II. CONSTRUCT, έχει ακριβώς το ίδιο συντακτικό με τη SELECT, με τη διαφορά ότι ως αποτέλεσμα δημιουργείται ένας νέος γράφος. Παράδειγμα κατασκευής γράφου:

```

CONSTRUCT { <http://dbpedia.org/resource/University> ?p ?o . }
WHERE {
    SERVICE <http://DBpedia.org/sparql> {
        <http://dbpedia.org/resource/University> ?p ?o . } }

```

- III. DESCRIBE: ανακτά τις περιγραφές των πόρων. Παράδειγμα περιέγραψε το ID M202:

```
DESCRIBE <http://ip/ontology/university.owl#M202>
```

- IV. ASK: χρησιμοποιείται για δυϊκά (λογικά) ερωτήματα στους γράφους με αποτέλεσμα Ναι (True) ή ΟΧΙ (False).

Παράδειγμα υπάρχει το όνομα Euclid στο namespace foaf:

```

PREFIX foaf:<http://xmlns.com/foaf/0.1/>
ASK { ?x foaf:name "Euclid" }

```

Στο τελευταίο SPARQL / Update 1.1 υπάρχει και η προσθήκη νέων εντολών όπως:

V. INSERT DATA

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

```
INSERT DATA { <http://example/book3> dc:title "A new book" ;  
dc:creator "A.N.Other" . }
```

VI. DELETE DATA

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

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

Η τρέχουσα έκδοση της SPARQL 1.1 μας επιστρέφει στα αποτελέσματα όλα τα literals για κάθε triplet, όπως πχ `^^string`. Τα αποτελέσματα της SPARQL αποτελούν μια πλήρη περιγραφή του ερωτήματος καθώς και τη ιδιότητα/τύπο για κάθε επιστρεφόμενη τιμή απο το χώρο XMLSchema (επίθεμα `:xsd`). Με τη τεχνική SPARQL Query Rewriting μπορούμε να ανακατασκευάσουμε το SPARQL ερώτημα έχοντας πιο συγκεκριμένο αποτέλεσμα, είτε βελτιστοποιώντας το ερώτημα είτε εφαρμόζοντας τεχνικές για πιο περίπλοκες συσχετίσεις [76] [77].

5.1.1 Ερωτήματα SPARQL

Δίνονται μερικά απο τα ερωτήματα που χρησιμοποιήθηκαν στη web εφαρμογή, καθώς και παραδείγματα χρήσης για LOD:

- Ποιές οντολογίες υπάρχουν ως προκαθορισμένες στο endpoint (imported schemas), δηλωμένες owl:Class;

```
SELECT distinct ?graph_uri  
WHERE { GRAPH ?graph_uri { ?s rdf:type owl:Class } }
```

- Βρες ποιος έχει/είχε δηλώσει το μάθημα με κωδικό M205;

```
SELECT *  
WHERE { ?subject ?t msc:M205. }
```

- Περιγράψε στο Namespace msc το ID Lecturer1.

DESCRIBE msc:Lecturer1

- Δώσε μου τα ονόματα των students με τα μαθήματα.

PREFIX msc:<http://urlOfOntology/ontology.owl#>

SELECT ?lastnameIS ?firstnameIS ?thesisIS (GROUP_CONCAT(?id; separator = "|") as ?cs)

WHERE { ?s msc:firstName ?firstname.

?s msc:lastName ?lastname. ?s msc:studies ?id.

OPTIONAL { ?s msc:workingThesis ?thesis }

bind(xsd:string(?lastname) as ?lastnameIS)

bind(xsd:string(?firstname) as ?firstnameIS)

bind(xsd:string(?thesis) as ?thesisIS)}

ORDER BY ?id

- Ποιος είναι ο lecturer1 (στιγμιότυπο) στην οντολογία μας;

PREFIX msc: <http://urlOfOntology/ontology.owl#>

SELECT ?type ?name

WHERE { msc:Lecturer1 ?type ?o.

bind(xsd:string(?o) as ?name)}

- Ποια είναι τα URIs των Πανεπιστημίων που βρίσκονται στις ΗΠΑ;

SELECT DISTINCT ?label ?homepage

WHERE {

?school a <http://schema.org/CollegeOrUniversity>

{ ?school dbpedia-owl:country dbpedia:United_States }

UNION

{ ?school dbprop:country dbpedia:United_States }

```

UNION
{ ?school dbpprop:country "U.S."@en }
UNION
{ ?school dbpprop:country "United States"@en }
OPTIONAL { ?school rdfs:label ?label .
FILTER (LANGMATCHES(LANG(?label), 'en')) }
OPTIONAL { ?school foaf:homepage ?homepage }
OPTIONAL { ?school geo:lat ?lat ; geo:long ?long }
OPTIONAL { ?school dbpedia-owl:endowment ?endowment }
}

```

- Εμφάνισε την αλφαβητική λίστα χωρών.

```

PREFIX ontology: <http://dbpedia.org/ontology/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?country ?capital
WHERE { ?c rdf:type ontology:Country;
rdfs:label ?country.
OPTIONAL { ?c ontology:capital ?cap_uri.
?cap_uri rdfs:label ?capital.
}}
ORDER BY ASC(?country )

```

- Εμφάνισε τους συγγραφείς που γεννήθηκαν στην Ελλάδα, από το 1920-1990.

```

SELECT distinct ?name ?birthDate ?occupation
WHERE {
?person dbpedia-owl:birthPlace <http://dbpedia.org/resource/Greece>;
dbpedia-owl:birthDate ?birthDate;
foaf:name ?name;
dbpedia-owl:occupation ?occupation.
FILTER (?birthDate > "1920-01-01"^^xsd:date &&
?birthDate < "1990-01-01"^^xsd:date ).

```

```

FILTER( lang(?name) = "en" ).

FILTER( regex(str(?occupation), 'Writer' )
} limit 100

```

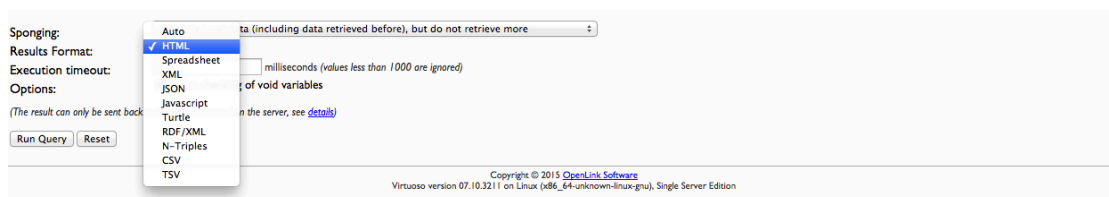
5.2 SPARQL Endpoint

Η πύλη διενέργειας ερωτημάτων του virtuoso, εμφανίζεται στο τελικό χρήστη ως μια ιστοσελίδα, όπου και μπορεί να υποβληθεί το ερώτημα SPARQL [78]. Η μορφή του συντάκτη των ερωτημάτων φαίνεται στην παρακάτω εικόνα:



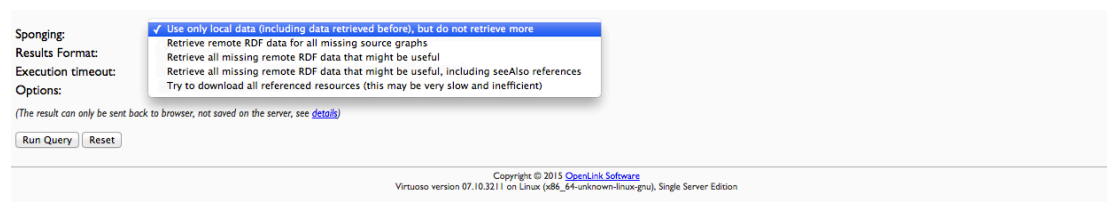
Εικόνα 4 : Virtuoso SPARQL Query Editor

Η μορφή της είναι σκοπίμως απλή, όπου μπορούμε να δώσουμε το προκαθορισμένο RDF γραφή που θα ασχοληθεί το ερώτημα μας (PREFIX). Στο query text, γράφουμε τη επιθυμητή ερώτηση σε SPARQL σύνταξη. Το αποτέλεσμα της ερώτησης μπορεί να εμφανιστεί με πολλούς τρόπους όπως HTML, XML, JSON, Turtle, N-Turtle κοκ. Δίνεται η λίστα με τις πιθανές μορφές εξαγωγής του αποτελέσματος.



Εικόνα 5 : Virtuoso SPARQL Query Editor- Results Format

Παρέχεται η δυνατότητα, το ερώτημα να επενεργεί μόνο σε τοπικά datasets, μόνο σε εξωτερικούς RDF graphs, σε συνδυασμό αυτών και άλλων όπως φαίνεται στην επόμενη εικόνα.



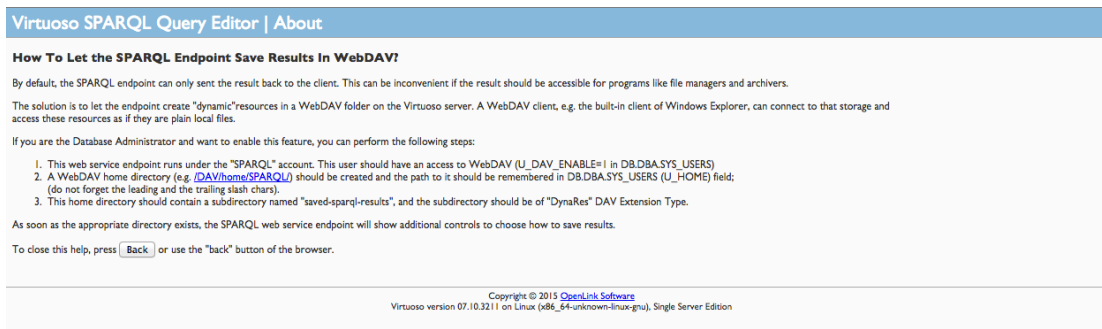
Εικόνα 6 : Virtuoso SPARQL Query Editor – Sponging [79]

Μπορούμε να δούμε όλα τα προεγκατεστημένα namespaces που υπάρχουν στο virtuoso και που μπορούν να χρησιμοποιηθούν στο SPARQL ερώτημα με το επίθημα : , όπως στην οντολογία μας msc: .

Prefix	URI
OVL	http://www.openlinksw.com/schemas/OVL#
a	http://www.w3.org/2005/Atom
address	http://schemas.talis.com/2005/address/schema#
admin	http://websns.net/mvcb/
atom	http://atomowl.org/ontologies/atomrdf#
aws	http://soap.amazon.com/
b3s	http://b3s.openlinksw.com/
batch	http://schemas.google.com/gdata/batch
bibo	http://purl.org/ontology/bibo/
bif	bif:
bugzilla	http://www.openlinksw.com/schemas/bugzilla#
c	http://www.w3.org/2002/12/cal/icaltzd#
category-en	http://dbpedia.org/resource/Category:
cb	http://www.crunchbase.com/
cc	http://web.resource.org/cc/
cnet	http://api.cnet.com/rest/v1.0/
content	http://purl.org/rss/1.0/modules/content/
cv	http://purl.org/captsolo/resume-rdf/0.2/cv#
cvbase	http://purl.org/captsolo/resume-rdf/0.2/base#
dawgt	http://www.w3.org/2001/sw/DataAccess/tests/test-dawg#
dbpedia	http://dbpedia.org/resource/
dbpedia-cs	http://cs.dbpedia.org/resource/
dbpedia-de	http://de.dbpedia.org/resource/
dbpedia-el	http://el.dbpedia.org/resource/
dbpedia-es	http://es.dbpedia.org/resource/
dbpedia-fr	http://fr.dbpedia.org/resource/
dbpedia-it	http://it.dbpedia.org/resource/

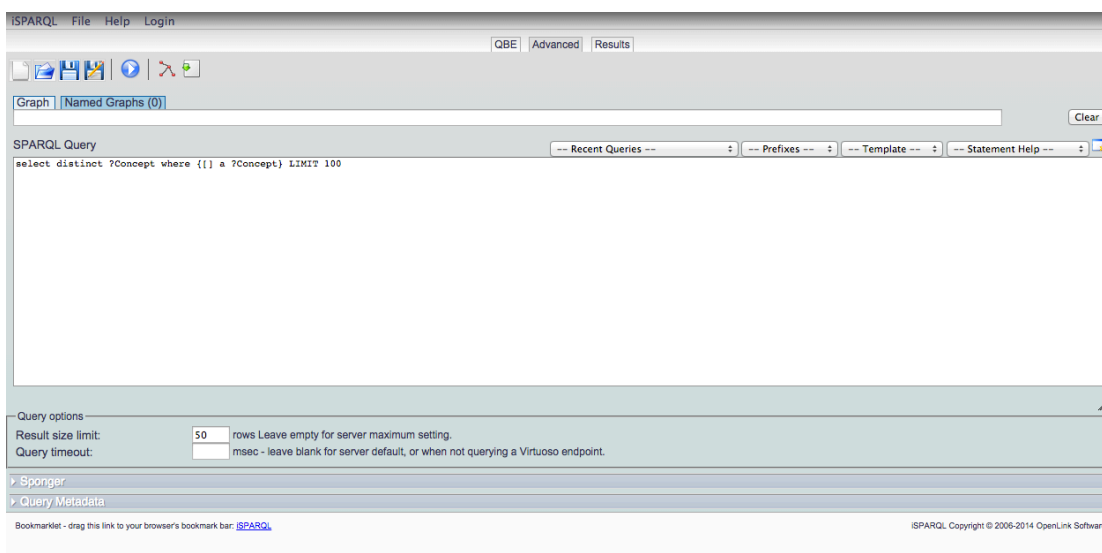
Εικόνα 7 : Virtuoso SPARQL Query Editor –Predefined Namespaces Prefix

Ένα σημαντικό χαρακτηριστικό που μπορεί να αποδώσει το virtuoso Endpoint είναι ότι μπορεί να αποδώσει το αποτέλεσμα όχι σε κάποια συγκεκριμένη μορφή αλλά εσωτερικά στο σύστημα WebDAV για περαιτέρω επεξεργασία.



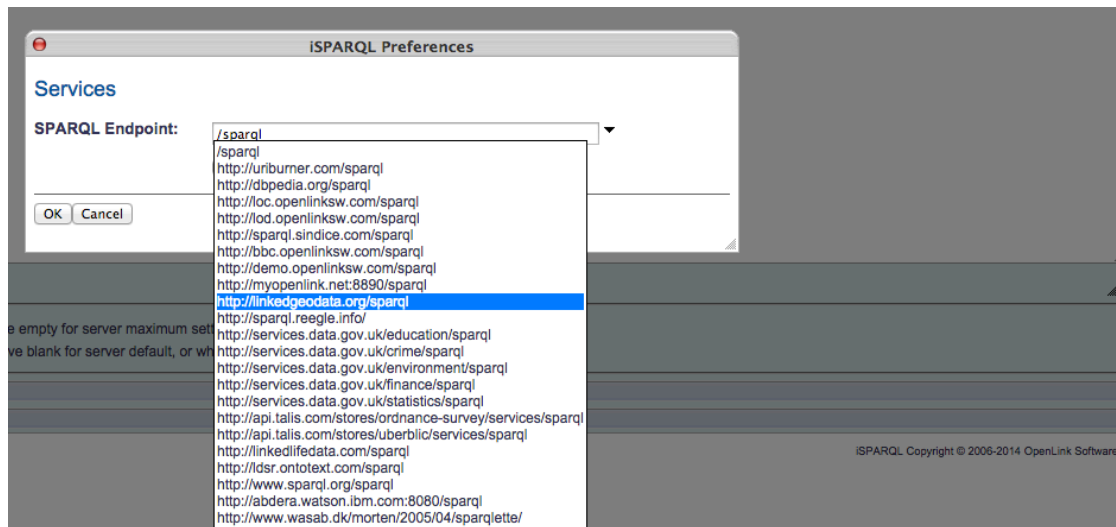
Εικόνα 8 : Virtuoso SPARQL Query Editor – results feedback

Το πρόσθετο πακέτο iSPARQL (VAD), αποτελεί ένα πιο ολοκληρωμένο web περιβάλλον για τη συγγραφή ερωτημάτων SPARQL. Βασικό χαρακτηριστικό είναι ότι μπορούμε να συνδεθούμε σε αυτό ως ταυτοποιημένοι χρήστες WebDAV και να αποθηκευτούν όλες μας οι ενέργειες. Το iSPARQL φαίνεται στην παρακάτω εικόνα 9,



Εικόνα 9 : Virtuoso iSPARQL Query Editor

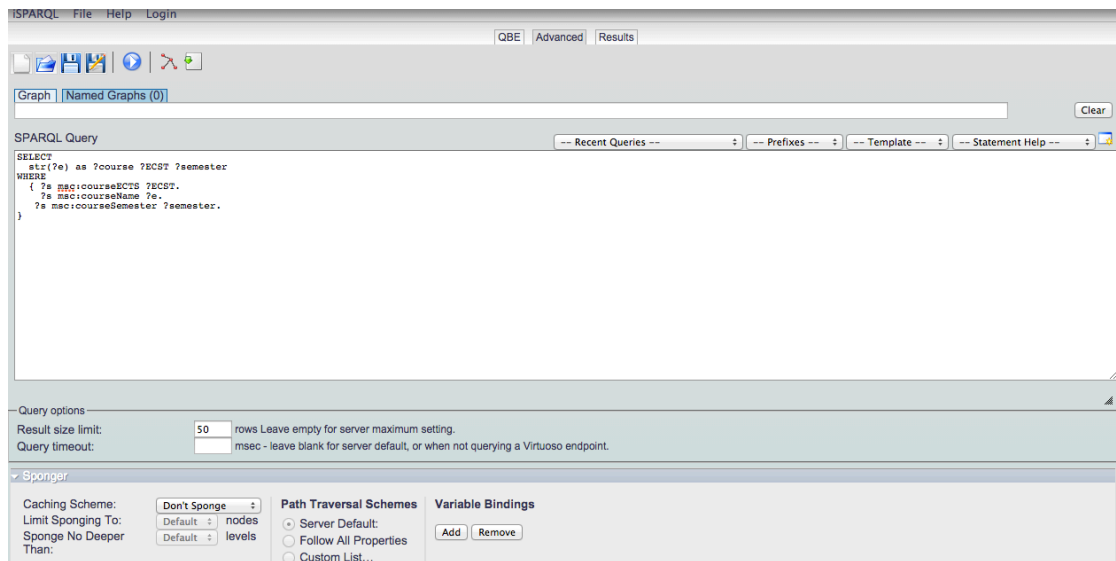
απο τις ρυθμίσεις μπορούμε να αλλάξουμε το SPARQL Endpoint σε ένα άλλο καθορισμένο πέραν του τοπικού, όπου εκτελείται το iSPARQL. Η λίστα έχει προκύψει απο τα πρόσθετα που έχουμε εγκαταστήσει μέσω του conductor.



Εικόνα 10 : Virtuoso iSPARQL Query Editor – Selecting Endpoint

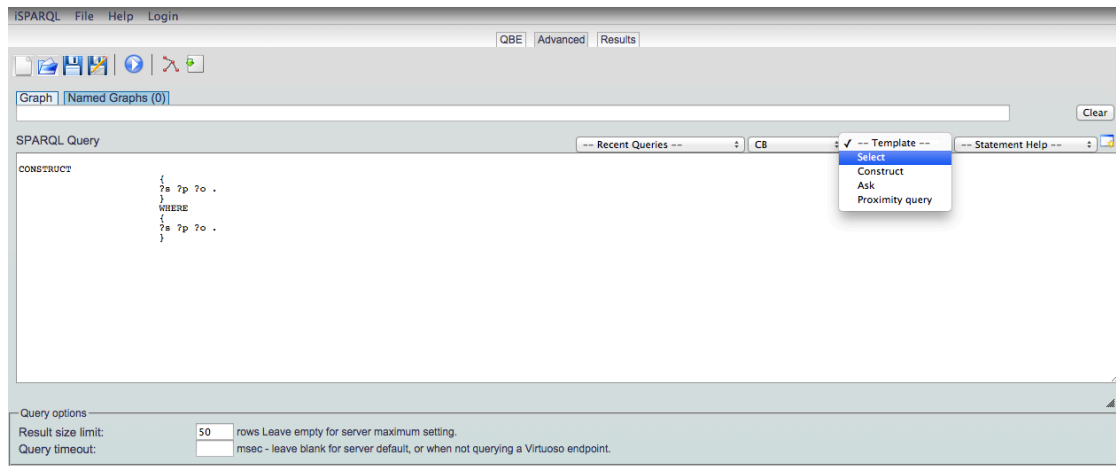
Στην κύρια οθόνη, διακρίνονται οι καρτέλες QBE, Advanced και Results:

Ο απλός συντάκτης ερωτημάτων βρίσκεται στην καρτέλα Advanced και έχει παραπλήσια πεδία όπως και το προηγούμενο απλό περιβάλλον,



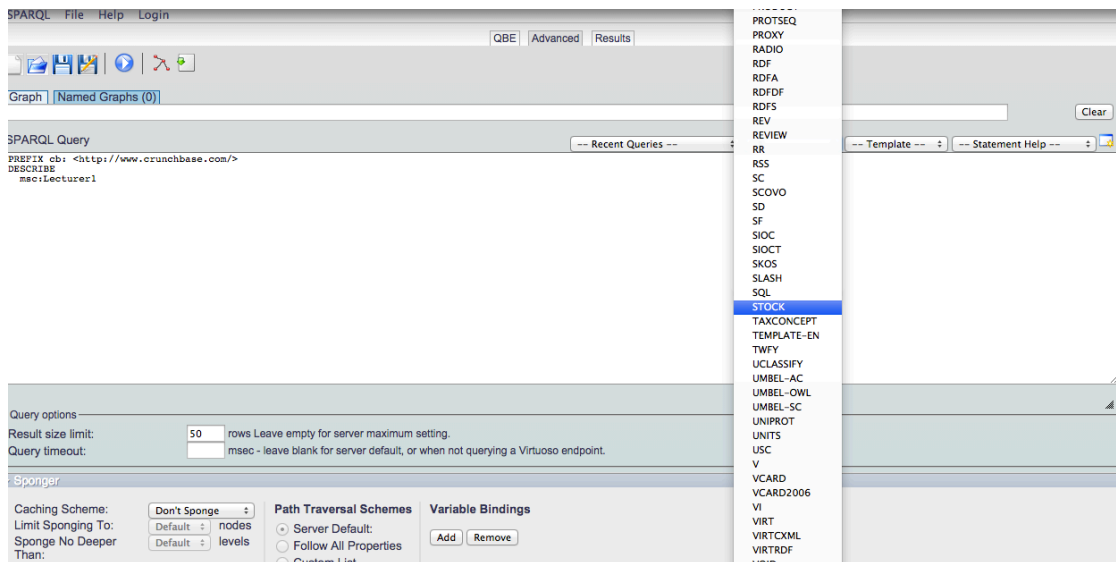
Εικόνα 11 : Virtuoso iSPARQL Query Editor – Advanced tab

αλλά με πολλές ακόμη δυνατότητες, όπως ένα βοηθητικό εργαλείο για τη σύνταξη των εντολών,



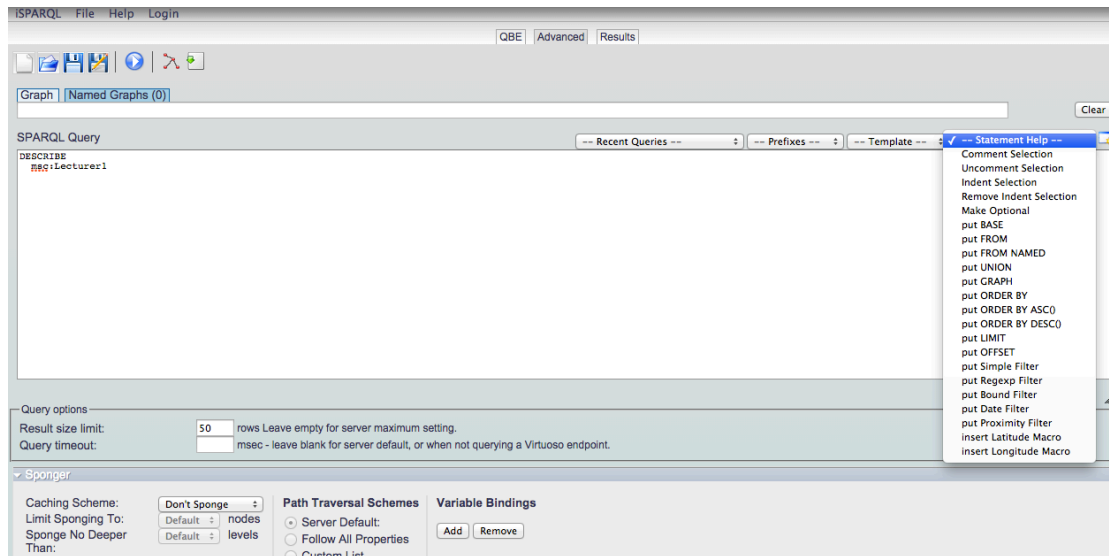
Εικόνα 12 : Virtuoso iSPARQL Query Editor – Template Syntax Helper

για την επιλογή μεταξύ των prefix



Εικόνα 13 : Virtuoso iSPARQL Query Editor – Prefix Selector

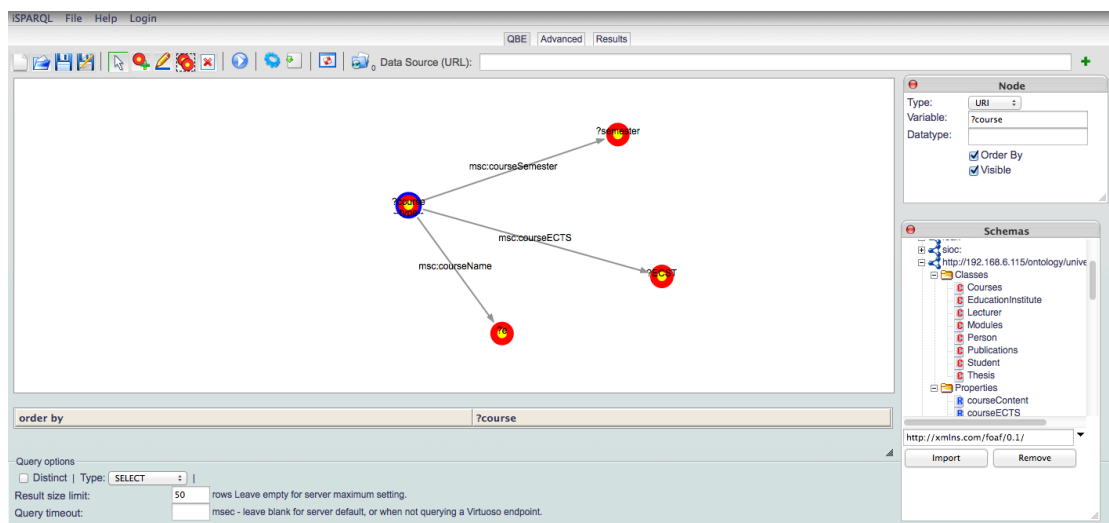
και οδηγό βοήθειας των δηλώσεων που απαρτίζουν το ερώτημα.



Εικόνα 14 : Virtuoso iSPARQL Query Editor – Commands Syntax Helper

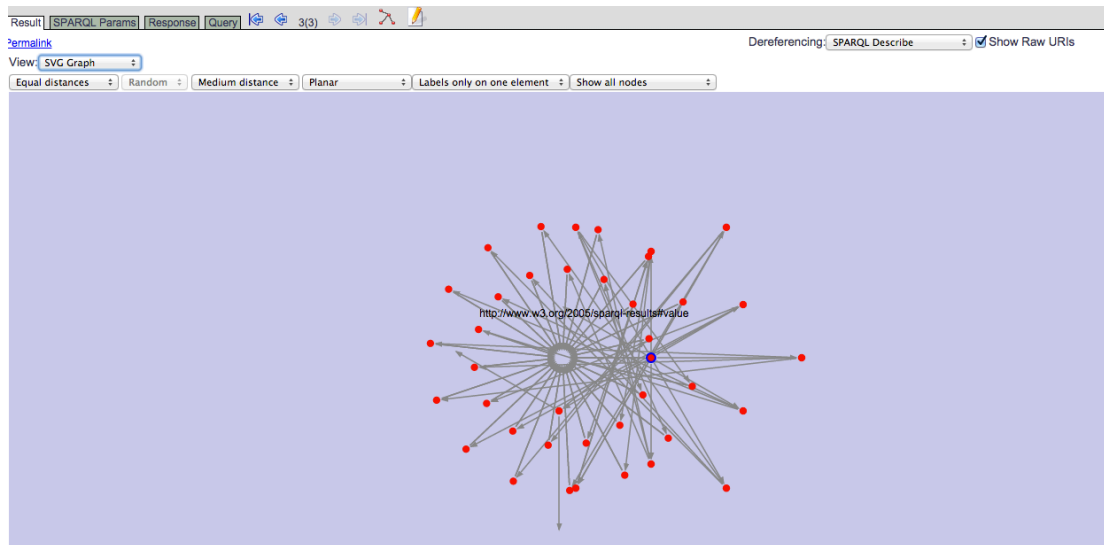
Στην καρτέλα QBE (Query By Example), μπορεί να αναπαρασταθεί οπτικά ο γράφος RDF που προέκυψε από το ερώτημα καθώς και βοηθητικές πληροφορίες περί των schemas RDF όπως φαίνεται στην εικόνα 15. Στο Schemas, μπορούμε να κάνουμε εισαγωγή ή αφαίρεση νέων γράφων και ταυτόχρονα να δούμε όλα τα classes και properties για κάθε διακριτό γράφο που υπάρχει στο schema μας.

Αποτελεί ένα ισχυρό εργαλείο, για τη συγγραφή των ερωτημάτων SPARQL και μπορεί να συνδυαστεί με το Sponger. Μπορεί να λειτουργήσει και ως εργαλείο εξερεύνησης για τους γράφους, μέσω της αντίστοιχης υποκαρτέλας Schemas. Προτείνεται ως το σημείο εκκίνησης για την δημιουργία γράφων, από νέους χρήστες.

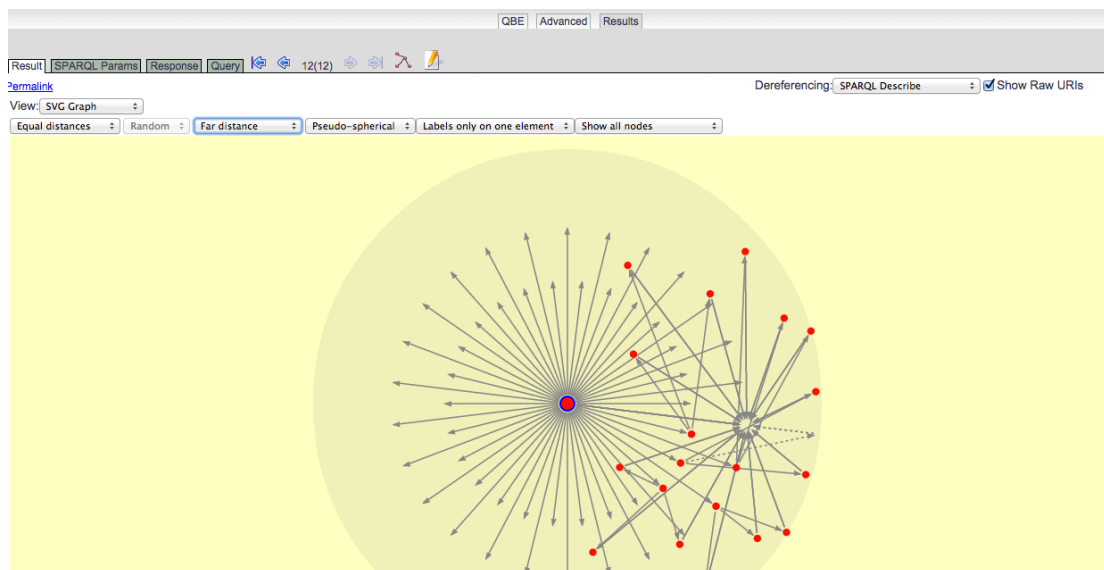


Εικόνα 15 : Virtuoso iSPARQL Query Editor – QBE display

Η οπτική αναπαράσταση της πληροφορίας, μπορεί γίνει με διάφορους τρόπους,



Εικόνα 16 : Virtuoso iSPARQL Query Editor – visual results graph SVG



Εικόνα 17 : Virtuoso iSPARQL Query Editor – visual results graph SVG (properties)

είτε με τον κλασσικό τρόπο με κείμενο, με N-turtles, JSON, HTML κ.α.

6

Τεχνικές λεπτομέρειες

Η συγκεκριμένη υλοποίηση χρειάστηκε εργαλεία αποτελούμενα από α) ένα λειτουργικό σύστημα Unix like και συγκεκριμένα το debian GNU/Linux ver. 7.6 β) το λογισμικό Virtuoso Open-source 7.1.1 [13] και γ) Protégé ver 5.0.0 [80].

6.1 Virtuoso Universal Server

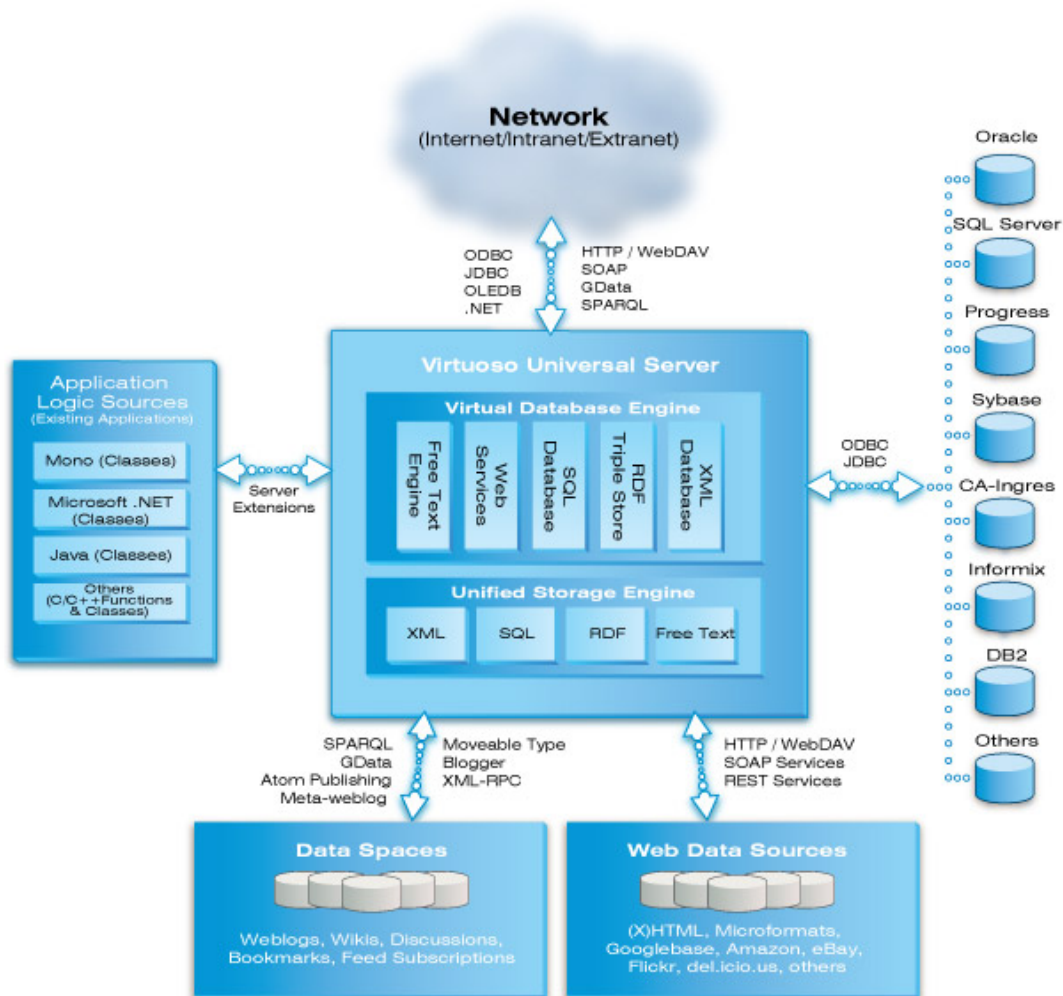
Η ολοκληρωμένη πλατφόρμα Virtuoso Open-Source είναι η επιλογή μας, για την υλοποίηση του SPARQL Endpoint. Υπάρχουν αρκετές εκδόσεις είτε εμπορικές είτε ανοικτού κώδικα. Οι διαφορές μεταξύ εμπορικής και μη έκδοσης, βρίσκονται σε συγκεκριμένες λειτουργίες, όπως τα ερωτήματα XML σε συνδυασμό με τη λειτουργία διασύνδεσης της πλατφόρμας με άλλα DBMS για database-federation (Virtual Database Functionality) [81]. Συνεπώς η έκδοση ανοικτού κώδικα που χρησιμοποιήσαμε δε περιλαμβάνει ορισμένες λειτουργίες που δίνονται με την εμπορική έκδοση του Virtuoso Universal Server [10].

Οι βασικές συνιστώσες της αρχιτεκτονικής περιγράφονται ως :

- I. Relational Data Management
- II. RDF Data Management
- III. XML Data Management

- IV. Free Text Content Management & Full Text Indexing
- V. Document Web Server
- VI. Linked Data Server
- VII. Web Application Server
- VIII. Web Services Deployment (SOAP or REST)

Η διαφοροποίηση στις λειτουργίες μεταξύ ανοικτού κώδικα και μη, δεν επηρεάζει την αρχιτεκτονική της συνολικής πλατφόρμας, όπως φαίνεται στην εικόνα παρακάτω.



Εικόνα 19 : Αρχιτεκτονική Virtuoso Universal Server / Open-Source Edition

Αναλύοντας το Virtuoso Universal Server, θα μπορούσαμε να το χαρακτηρίσουμε ως ένα ολοκληρωμένο πληροφοριακό σύστημα διαφόρων συνεργαζόμενων υπηρεσιών με απώτερο σκοπό την παροχή πληροφορίας στην ψηφιακή εποχή και δη στον σημασιολογικό ιστό. Αναλυτικότερα πρόκειται για μια πλατφόρμα (middleware) υπηρεσιών ανεξαρτήτου λειτουργικού συστήματος που ονομάζεται Universal Server.

Το OpenLink Universal Server (middleware) υλοποιεί σε ένα και μοναδικό εξυπηρετητή τους ακόλουθους διακριτούς servers [9] [82]:

- I. Web Server (HTTP 1.0/1.1) με δυνατότητα εκτέλεσης πληθώρας τεχνολογιών πέραν την δικής του τεχνολογίας VSP (virtuoso server pages) όπως PHP, Zend (server Extensions Interface VSEI)
- II. WebDAV/File Server [82]
- III. Database Server (ORDBMS) με δυνατότητες διασύνδεσης (ODBC, JDBC, OLE-DB) με όλες τις γνωστές βάσεις δεδομένων και πρότυπα (SQL-92) καθώς και όλους τους γνωστούς τύπους σύνδεσης σε αυτές, όπως XML, XPATH, XSLT
- IV. Web Services Generator με τεχνολογίες SOAP, UDDI. Βασικό χαρακτηριστικό του universal server είναι ότι κάθε είδους πληροφορία που διαθέτει είναι διαθέσιμη μέσω web services. Η δημιουργία μιας web service απαιτεί μόνο την ύπαρξη μιας SQL Stored procedure [82] [9]
- V. Mail Server SMTP, POP3, IMAP4
- VI. NNTP Server (Network News Transfer Protocols)

Καθένας από τους παραπάνω servers, διαθέτει χαρακτηριστικά εφάμιλλα των πλέον αναγνωρίσιμων server όπως Apache2, MySQL και όλα αυτά μαζί ενώνονται με ένα μαγικό τρόπο και μας δίνουν τον πολύ ξεχωριστό οικουμενικό server [82].

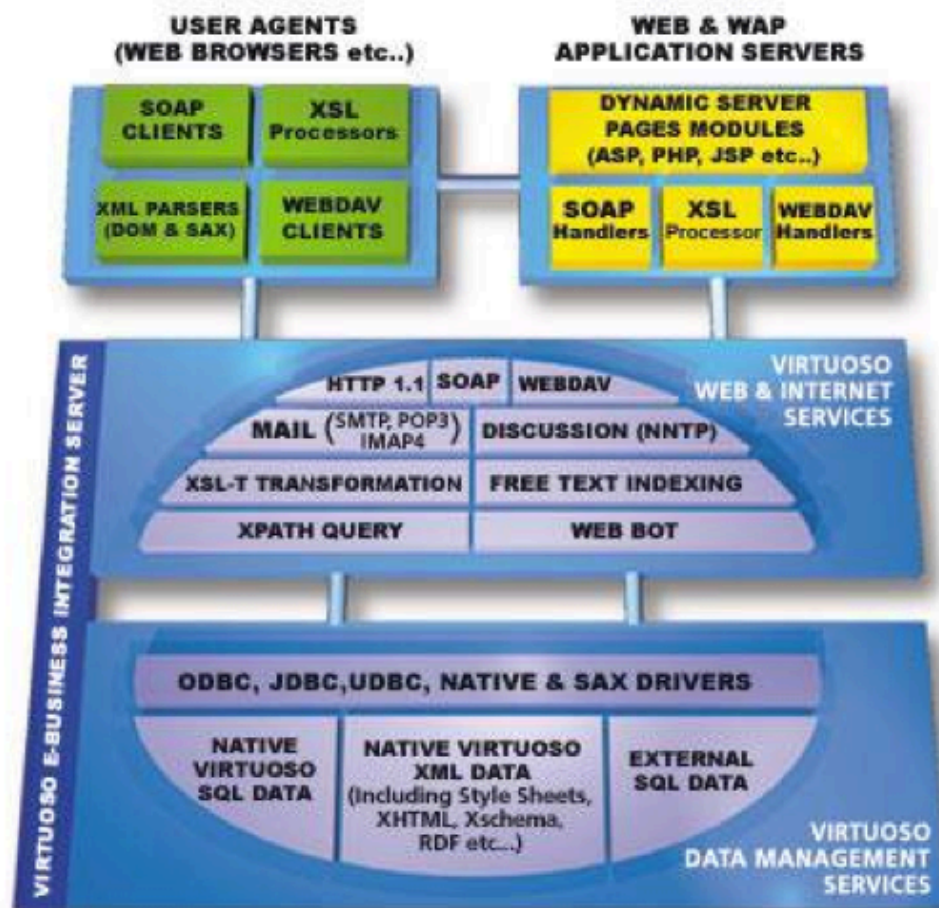
Ένα από κύρια πλεονεκτήματά του είναι η παραγωγή XML/DTD/RDF δυναμικά από τα υπάρχοντα ομογενή ή ετερογενή δεδομένα (δομημένα ή αδόμητα) όπως XML, HTML, SQL ερωτημάτων [79] [82].

Γιατί να υπάρχει το Virtuoso :

Γιατί η πληροφορία μας παρέχει δύναμη και πλεονέκτημα έναντι των άλλων που δεν την κατέχουν. Με την ικανότητα του virtuoso να συνδυάζει όλες τις παραδοσιακές πηγές data όπως XML, DB, text, html, αδόμητο κείμενο και άλλες σε μια κατανοητή μορφή RDF/ OWL για το σημασιολογικό ιστό, γίνεται ο πλέον αναγκαίος server για την παροχή της πληροφορίας στην νέα εποχή της κατανεμημένης πληροφορίας [82].

Το Virtuoso SPARQL Endpoint είναι ο κύριος εκφραστής αυτής της ικανότητας του universal server.

Η ενοποίηση όλων των παραπάνω τεχνολογιών και η αξιοποίηση των διαθέσιμων εργαλείων του virtuoso το καθιστά ισχυρό σύμμαχο στη εποχή της πληροφορίας.



OpenLink Virtuoso Product Architecture

Εικόνα 20 : Αρχιτεκτονική Openlink Virtuoso Product

Το virtuoso universal server στην ουσία είναι ένα middleware με ενσωματωμένο database engine. Συνδυάζει αρκετές τεχνολογίες από τις παραδοσιακές RDBMS, ORDBMS, virtual database, RDF, XML, free text - αδόμητη μορφή, web server και file server [9] [82].

Πρόκειται για μια πλατφόρμα λογισμικού που αποτελείται από τον openlink universal server, και μπορούμε να ενσωματώσουμε διάφορα εργαλεία (VAD) για να δημιουργήσουμε το Endpoint [82]. Το σημαντικότερο ίσως VAD είναι το εργαλείο conductor, που είναι η διεπαφή για την παραμετροποίηση του server.

Έχοντας εγκαταστήσει και το `http_server`, μπορούμε να έχουμε και ένα γενικής χρήσης `web server` στο γενικό πρωτόκολλο HTTP 1.0 και HTTP 1.1. Δυστυχώς κατά τη διάρκεια της συγγραφής έχει γίνει πλέον διαθέσιμο και το HTTP 2 και πρόκειται να υποστηριχθεί στις επόμενες εκδόσεις του `universal server`. Προκαθορισμένος ο `http sever` υποστηρίζει τεχνολογία `vsp` (Virtuoso Server Pages) [82], ενώ έχουμε και τη δυνατότητα να τον ρυθμίσουμε και σε διάφορες γνωστές πλέον τεχνολογίες όπως `php`, `asp` [82]. Η τεχνολογία `vsp`, μας επιτρέπει να γράψουμε SQL εντολές μέσα στο `html` έγγραφο, το οποίο θα μεταφραστεί από τον `universal server`.

6.1.1 Εγκατάσταση

Η εγκατάσταση της Open-Source έκδοσης, έγινε από το αποθετήριο της εταιρίας Openlink στο GITHUB [12] [13]. Πρόκειται για την έκδοση `7.1.1-dev`. Η διαδικασία εγκατάστασης που ακολουθήσαμε, προϋποθέτει τη πρόσβαση σε τερματικό (`console`) σε ένα `unix-like` λειτουργικό σύστημα όπως είναι το GNU/Linux. Τα ακόλουθα βήματα είναι σε γραμμή εντολών και έχουν ως εξής :

Μεταφόρτωση του πηγαίου κώδικα στο τοπικό μας σύστημα μέσω `git` [83],

1. `git clone https://github.com/openlink/virtuoso-opensource.git`

εγκατάσταση των προαπαιτούμενων βιβλιοθηκών για την σωστή λειτουργία,

2. `sudo apt-get install automake glibtoolize flex bison gperf gawk m4 make openssl libssl-dev libtool`

κατευθυντήριες οδηγίες προς τον μεταγλωττιστή (προϋπόθεση της open-source έκδοσης 64bit OS) `gcc [xx]`,

3. `CFLAGS="-O2 -m64"`

4. `./autogen.sh # should only be needed in git clone`

5. `./configure`

διαδικασία μεταγλώττισης και

6. `make`

εγκατάσταση στο τοπικό μηχάνημα. Είναι μια χρονοβόρα διαδικασία κατά την οποία γίνεται μεταγλώττιση του πηγαίου κώδικα για το συγκεκριμένο μηχάνημα.

7. `sudo make install`

8. `sudo make check`

Αποδιδόμενες πληροφορίες της εγκατάστασης:

output is Virtuoso Open Source Edition (Column Store) 7.1.1-dev configuration summary

Installation variables

<i>layout</i>	<i>default</i>
<i>prefix</i>	<i>/usr/local/virtuoso-opensource</i>
<i>exec_prefix</i>	<i>\${prefix}</i>

Installation paths

<i>programs</i>	<i>\${exec_prefix}/bin</i>
<i>include files</i>	<i>\${prefix}/include</i>
<i>libraries</i>	<i>\${exec_prefix}/lib</i>
<i>manual pages</i>	<i>\${datarootdir}/man</i>
<i>vad packages</i>	<i>\${datarootdir}/virtuoso/vad</i>
<i>database</i>	<i>\${prefix}/var/lib/virtuoso/db</i>
<i>hosting</i>	<i>\${exec_prefix}/lib/virtuoso/hosting</i>

Options

<i>BUILD_OPTS</i>	<i>xml ssl imsg pldebug pthreads</i>
-------------------	--------------------------------------

Το βασικό αρχείο ρυθμίσεων για τη πλατφόρμα βρίσκεται στη διαδρομή

/usr/local/virtuoso-opensource/var/lib/virtuoso/db/virtuoso.ini

μπορεί επίσης να διαμορφωθεί και από το web interface / conductor.

Για την διαμόρφωση διαμέσου του cli (command-line interface), θα χρειαστούμε και ένα πρόσθετο εργαλείο (*vload*) για τη διαχείριση RDF γράφων, από το αποθετήριο [84].

1. *git clone https://github.com/timrdf/csv2rdf4lod-automation*

Για την απευθείας εκτέλεση του virtuoso, από το ΛΣ, δημιουργήθηκε το script συστήματος / *VirtuosoServer.sh* που είναι υπεύθυνο για την αυτόματη εκτέλεση λειτουργιών όταν το σύστημα εισέρχεται σε runlevel 2 (κανονική εκκίνηση) ή τον τερματισμό του ΛΣ – runlevel 0 [85].

2. sudo update-rc.d VirtuosoServer.sh defaults

Δίνεται το shell script VirtuosoServer.sh σε RAW μορφή:

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:      VirtuosoServer initialize © dsphinx 2015
# Default-Start: 2
# Default-Stop: 0 6
### END INIT INFO

PATH=/sbin:/bin:/usr/bin:/home/dsphinx/bin/
VIRT=/usr/local/virtuoso-opensource/var/lib/virtuoso/db
ONTOLOGYOWL=/var/www/ontology/university.owl
ONTOLOGYURI=http://SERVERIP/ontology/university.owl

./lib/init/vars.sh
./lib/lsb/init-functions

do_stop () {
    [ "$VERBOSE" = no ] || log_action_begin_msg "VirtuosoServer stop "
    killall virtuoso-t
    log_action_begin_msg "VirtuosoServer shutdown script "
}

case "$1" in
    start)
        killall virtuoso-t
        ( /usr/local/virtuoso-opensource/bin/virtuoso-t -f -c "$VIRT/virtuoso.ini" ) &
        sleep 2
        # run once to load ontology
        vload rdf "$ONTOLOGYOWL" "$ONTOLOGYURI"
        echo "VirtuosoServer start "
```

```

;;
restart|reload|force-reload)
    echo "Error: argument '$1' not supported" >&2
    do_stop
    exit 3
;;
stop)
    do_stop
;;
*)
    echo "Usage: $0 start|stop" >&2
    exit 3
;;
esac

```

6.1.2 Διαμόρφωση

Η εξ ορισμού εγκατάσταση, μας δίνει τις βασικές λειτουργίες για την περαιτέρω παραμετροποίηση του virtuoso. Οι ρυθμίσεις μπορούν αλλάξουν είτε απο γραφικό περιβάλλον μέσω του Virtuoso Conductor στο τμήμα Interactive SQL, είτε από τη γραμμή εντολών μεσω isql-vt (SQL interface).

❖ Παραμετροποίηση από γραμμή εντολών

Ο χρήστης του συστήματος στο virtuoso DB, που είναι υπεύθυνος για τα ερωτήματα στο Endpoint, ονομάζεται SPARQL και έχει κάποιους συγκεκριμένους ρόλους (ROLES). Ως προεπιλογή έχουν δοθεί ελάχιστα προνόμια (ROLES).

Για τις ανάγκες του SPARQL Endpoint, θα χρειαστεί να ρυθμίσουμε την ικανότητα του virtuoso για την αποδοχή εκτέλεσης απομακρυσμένου ερωτήματος SPARQL, μέσω της δεσμευμένης λέξης SERVICE [4]. Τα βήματα έχουν ως εξής :

- `isql-vt localhost:1111 dba <dba_password>`

```
SQL> grant select on "DB.DBA.SPARQL_SINV_2" to "SPARQL";
SQL> grant execute on "DB.DBA.SPARQL_SINV_IMP" to "SPARQL";
```

Αυτό θα αποτρέψει το ενοχλητικό μήνυμα όταν δεν έχει δικαιώματα ο χρήστης SPARQL του συστήματος εμφανίζοντας το μήνυμα λάθους :

Virtuoso 42000 Error SQ200: Must have select privileges on view DB.DBA.SPARQL_SINV_2

Μια άλλη σημαντική λεπτομέρεια, είναι η απόδοση δικαιωμάτων στο χρήστη SPARQL ώστε να μπορεί να εκτελέσει τις λειτουργίες για τα διασυνδεδεμένα δεδομένα (linked data). Η προκαθορισμένη ρύθμιση έχει κλειδωμένα τα δικαιώματα. Δίνονται οι δηλώσεις isql-vt για την διαμόρφωση :

```
-- Interactive SQL area

GRANT execute ON SPARQL_INSERT_DICT_CONTENT TO "SPARQL"

GRANT execute ON SPARQL_INSERT_DICT_CONTENT TO SPARQL_UPDATE

-- DELETE and insert

GRANT execute ON DB.DBA.SPARQL_MODIFY_BY_DICT_CONTENTS TO "SPARQL"

GRANT execute ON DB.DBA.SPARQL_MODIFY_BY_DICT_CONTENTS TO
SPARQL_UPDATE

-- DELETE only

GRANT execute ON DB.DBA.SPARQL_DELETE_DICT_CONTENT TO "SPARQL"

GRANT execute ON DB.DBA.SPARQL_DELETE_DICT_CONTENT TO
SPARQL_UPDATE

-- SPARQL

GRANT execute ON DB.DBA.SPARUL_RUN TO "SPARQL"

GRANT execute ON DB.DBA.SPARUL_RUN TO SPARQL_UPDATE

-- DELETE graph

GRANT execute ON SPARUL_CLEAR TO "SPARQL"

GRANT execute ON SPARUL_CLEAR TO SPARQL_UPDATE

GRANT DELETE ON RDF_QUAD TO "SPARQL"
```

```
GRANT DELETE ON RDF_QUAD TO SPARQL_UPDATE
```

```
-- version 7
```

```
GRANT execute ON DB.DBA.RDF_OBJ_ADD_KEYWORD_FOR_GRAPH TO  
"SPARQL"
```

```
GRANT execute ON DB.DBA.RDF_OBJ_ADD_KEYWORD_FOR_GRAPH TO  
SPARQL_UPDATE
```

```
GRANT execute ON DB.DBA.L_O_LOOK TO "SPARQL"
```

```
GRANT execute ON DDB.DBA.L_O_LOOK TO SPARQL_UPDATE
```

Για την εγκατάσταση/εισαγωγή μιας νέας οντολογίας (αρχεία δεδομένων - dataset), χρησιμοποιούνται 2 κύριοι μέθοδοι είτε με προγράμματα μέσω γραμμής εντολών, είτε με εισαγωγή αυτών μέσω Conductor Linked Data > Quad Store Upload.

Ο πρώτος τρόπος γίνεται είτε με το εργαλείο γραμμής εντολών (cli) vload [84] [93].

- `vload owl/rdf/ttl <ontology.owl> URI`

είτε μέσω `isql` με τις ακόλουθες εντολές `ls_dir` και `ls_dir_all` (recursive dir) [91]

- `isql-vt localhost:1111 dba <dba_password>`
`$ ld_dir ('/path/to/files', '*.ttl.gz', 'http://dbpedia.org');`
`$ ld_dir_all ('/path/to/files', '*.ttl.gz', 'http://dbpedia.org');`
`$ rdf_loader_run(); # run the loading`

Τα αρχεία δεδομένων που θα πρέπει να φορτωθούν θα πρέπει να συμβαδίζουν με τα εξής μορφότυπα [91] [92]:

Επέκταση	Μορφότυπο
.grdf	Geospatial RDF
.nq	N-Quads
.nt	N-Triples
.owl	OWL
.rdf	RDF/XML
.trig	TriG
.ttl	Turtle
.xml	RDF/XML

Πίνακας 1: υποστηριζόμενα μορφότυπα Virtuoso Loader

Σύμφωνα πάντα με τον όγκο της πληροφορίας (dataset) που θα ανεβάσουμε στο Virtuoso, θα πρέπει να το υποστηρίξουμε με το ανάλογο υλικό σε δυνατότητες (RAM, CPUs) και να ρυθμίσουμε κατάλληλα το αρχείο virtuoso.ini [92].

Η σχέση μεταξύ dataset και μνήμης RAM δίνεται από τον παρακάτω τύπο :

$$NumberOfBuffers = (Free Memory * 0.66)/8000$$

RAM	NumberOfBuffers	MaxDirtyBuffers
2 GB	170000	130000
4 GB	340000	250000
8 GB	680000	500000
16 GB	1360000	1000000
32 GB	2720000	2000000
48 GB	4000000	3000000
64 GB	5450000	4000000

Πίνακας 2: Αναλογία RAM με dataset

Για την εισαγωγή εξωτερικών Graphs στο τοπικό Schema, υπάρχουν διάφοροι τρόποι [93] όπως:

- SPARQL Insert
- SPARQL Load Into Graph URI
- SPARQL Load

Για παράδειγμα η δήλωση :

```
Load <http://www.w3.org/People/Berners-Lee/card#i> into graph
<http://www.w3.org/People/Berners-Lee/card#i>
```

εισαγάγει στο τοπικό Schema το Graph <http://www.w3.org/People/Berners-Lee/card#i>.

Για τη διαγραφή Graphs μας θα χρειαστούμε να δώσουμε τα απαραίτητα δικαιώματα (ROLES) SPARQL_UPDATE στον χρήστη SPARQL που είναι υπεύθυνος για κάθε ερώτημα SPARQL. Συνεπώς για να μπορεί να διαγράψει το γράφο, θα πρέπει να του αποδοθεί το δικαίωμα ανανέωσης (SPARQL_UPDATE):

- `clear GRAPH <URI>;`

Για την ενεργοποίηση του επιπέδου καταγραφής στα αντίστοιχα αρχεία [86], δίνουμε:

- `trace_on()`;

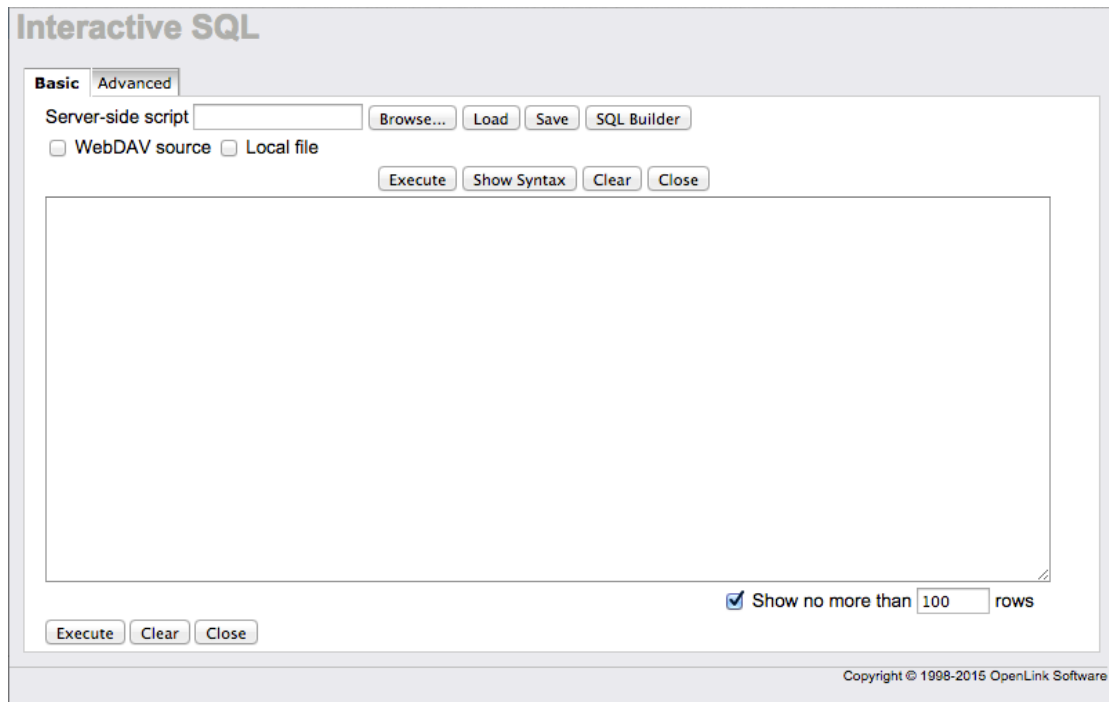
❖ Παραμετροποίηση από Web Interface / Conductor

Η παραμετροποίηση του virtuoso, γίνεται από το web interface στη διεύθυνση όπου βρίσκεται εγκατεστημένο το λογισμικό και στην δεσμευμένη TCP port 8890. URI: π.χ http://sparql_endpoint_ip:8890/. Ο Conductor, είναι το web interface που επιτρέπει να διαμορφώσουμε τη συμπεριφορά του SPARQL Endpoint [82] [9].

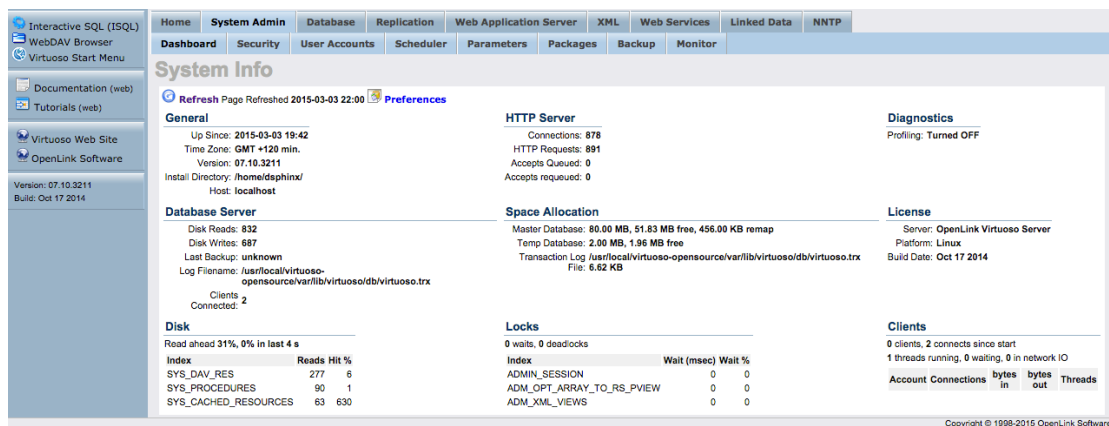


Εικόνα 21 : Virtuoso Conductor – main interface

Συνδεόμαστε ως dba (database administrator). Πέραν του γραφικού περιβάλλοντος, μπορούμε να χρησιμοποιήσουμε και το Interactive SQL για να δώσουμε τις εντολές interactive. Στην ουσία πρόκειται για ένα frontend του εργαλείου γραμμής εντολών isql.



Εικόνα 22 : Virtuoso Conductor – isql Web frontend



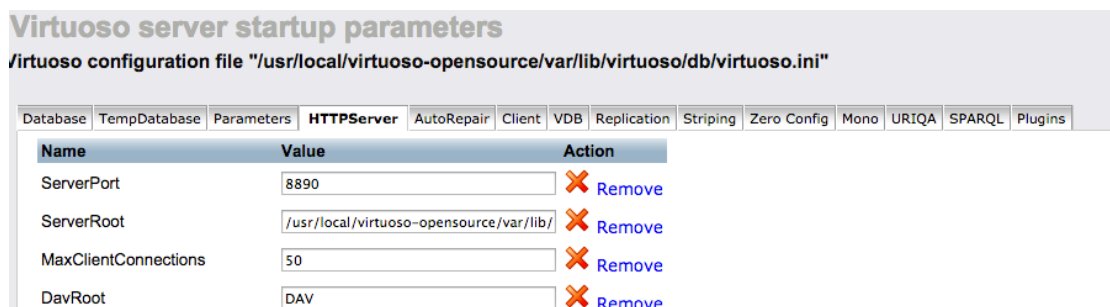
Εικόνα 23 : Virtuoso Conductor – dashboard

▪ Καρτέλα System Admin

Η καρτέλα System Admin, μας παρέχει τις δυνατότητες για :

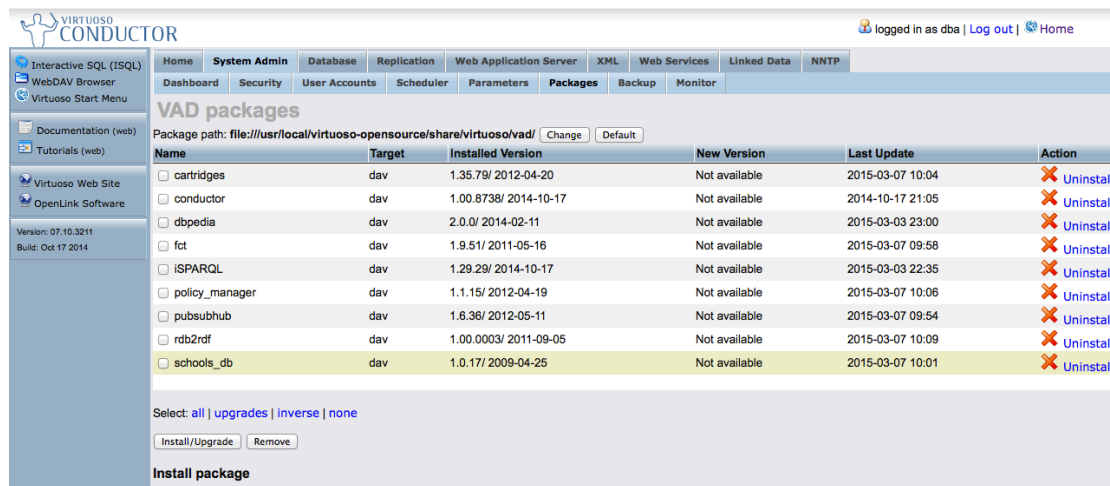
- Υποκατηγορία Security, να εκδώσουμε πιστοποιητικό για σύνδεση στην υπηρεσία με ασφαλή σύνδεση https, να δημιουργήσουμε access lists που θα επιτρέπουν ή όχι τη σύνδεση υπό συνθήκες.
- Υποκατηγορία User Accounts, να διαχειριστούμε τους χρήστες του συστήματος (dba, SPARQL, dav), να δημιουργήσουμε νέους με συγκεκριμένους ρόλους (roles), όπως SPARQL_SELECT για μόνο ερωτήματα SPARQL ή για ανανέωση των datasets μας με SPARQL_UPDATE. Παρέχεται επίσης η δυνατότητα για εισαγωγή λογαριασμών χρηστών από LDAP servers.

- Υποκατηγορία Scheduler, δρομολογητής συμβάντων, όπου μπορούμε να χρονοπρογραμματίσουμε συγκεκριμένες εργασίες μέσω εντολών.
- Υποκατηγορία Parameters, στη καρτέλα αυτή φαίνονται όλες οι πιθανές εκδοχές των διακοπών που ρυθμίζουν τον server κατά την εκκίνηση του. Στην ουσία πρόκειται και πάλι για ένα front end του καθολικού αρχείου παραμέτρων virtuoso.ini. Βασικές λειτουργίες όπως οι επιλεχθείσες TCP ServerPort 1111 για τον SPARQL server, η TCP 8890 port για το HTTPServer. Αλλαγή του MaxClientConnections σε 50 ανα χρονική μονάδα και στον SPARQL server και στο HTTP Server.



Εικόνα 24 : Virtuoso Conductor – HTTP Server settings

- Υποκατηγορία Packages, όπου μπορούμε να διαχειριστούμε πρόσθετα εργαλεία VAD (Virtuoso Application Distribution) για το virtuoso server.



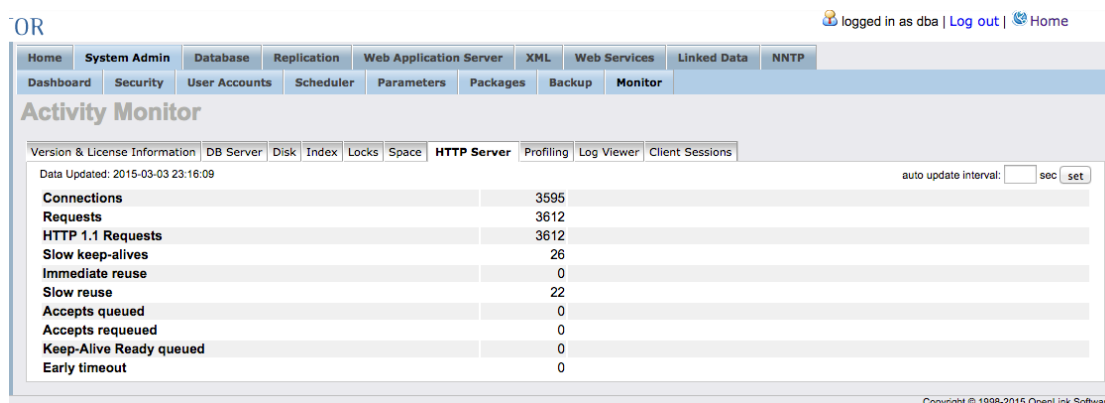
Εικόνα 25 : Virtuoso Conductor – VAD package manager

Προστέθηκαν αρκετά εργαλεία (VAD), μερικά απο τα οποία είναι : το εργαλείο iSPARQL - The OpenLink interactive SPARQL Query Builder με πρόσβαση σ' αυτό στο URI <http://server:8890/isparql/> (εικόνα 9).

Επίσης το dbpedia VAD για συγχρονισμό με τη dbpedia schemas. Στην ουσία τα VAD είναι SGML documents, που περιέχουν της νέες πρόσθετες λειτουργίες όπως δημιουργία πινάκων, triggers κοκ. Σε περίπτωση αποτυχίας κατά τη διάρκεια της

εγκατάστασης ενός VAD, ο universal server επανέρχεται στην προηγούμενη κατάσταση πριν την αποτυχία [87].

- Υποκατηγορία Backup, για διαδικασίες δημιουργίας αντιγράφων ασφαλείας χρονοπρογραμματισμένες ή μη.
- Υποκατηγορία Monitor, για την παρακολούθηση του virtuoso server σε πραγματικό χρόνο, καθώς και απόδοση στατιστικών.



Εικόνα 26 : Virtuoso Conductor – Activity monitor

▪ Καρτέλα Database

Η καρτέλα Database, μας παρέχει με τις δυνατότητες για :

- Υποκατηγορία SQL Database objects

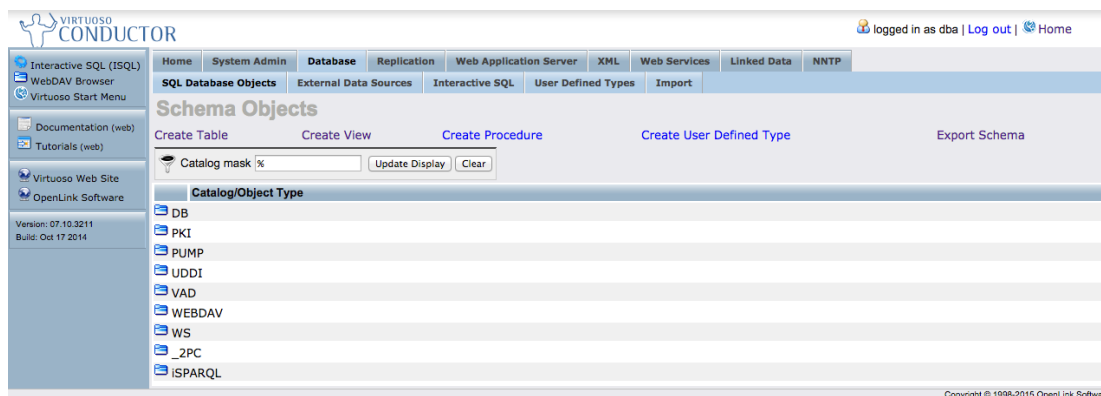
Το virtuoso project προέκυψε με τη συγχώνευση των διαφορετικών Openlink data access middleware και του Kubl RDMBS [9][11][82]. Περιέχει ένα ολοκληρωμένο ORDBMS σύστημα ΒΔ με την ονομασία Kubl που ξεκίνησε το 1994.

Ο πυρήνας για όλες τις υπηρεσίες που μπορεί να προσφέρει το openlink universal server, παραμένει το σύστημα βάσης δεδομένων, όπου και μπορούν να συγχρονιστούν όλες οι υπηρεσίες. Στη παρακάτω εικόνα (εικόνα 28), φαίνονται όλα τα schemas που διαχειρίζεται ο server. Το DB schema, όπως γίνεται κατανοητό αφορά τους εσωτερικούς πίνακες/σχηματισμούς για το κομμάτι της ORDBMS. Το schema VAD, αποτελεί το χώρο όπου μπορούν να χρησιμοποιήσουν τα νέα πρόσθετα ενθέματα που έχουμε εγκαταστήσει μέσω του package manager. Το webdav schema, αποτελεί το ακρογωνιαίο λίθο για το webdav file server που διαθέτει ως υπηρεσία. Ως WS schema, για όλες τις web services που χρησιμοποιούν τα δεδομένα του universal server. Το schema iSPARQL για τις ρυθμίσεις του ένθετου interactive SPARQL editor.

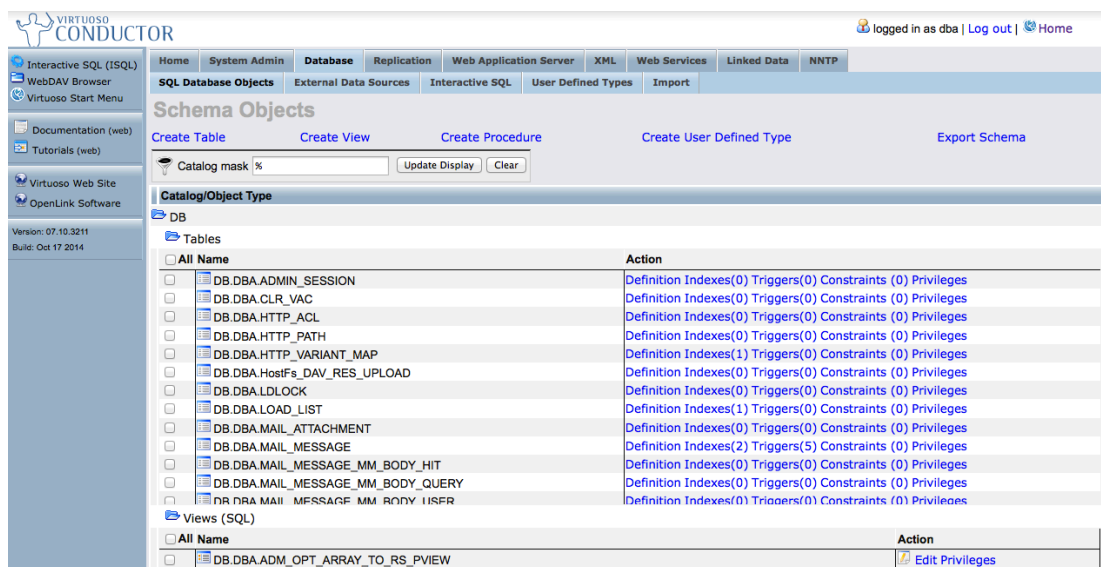
Ως σύγχρονο DBMS, μας δίνει τη δυνατότητα για δημιουργία όψεων (VIEWS) μεταξύ όλων των DB των ενσωματωμένων υπηρεσιών για περαιτέρω κατηγοριοποίηση. Μπορούμε επίσης

να δημιουργήσουμε και stored procedures για μεγαλύτερη ευκολία. Να σημειωθεί ότι κάθε λειτουργία/ κλήση σε VSP (server pages) λειτουργεί αυτόματα και ένα stored procedure.

Φυσικά μπορούμε να δημιουργήσουμε και δικούς μας τύπους δεδομένων (user defined type) και να κάνουμε εξαγωγή κάθε schema για λόγους ανάλυσης ή και για αντίγραφα ασφαλείας.



Εικόνα 27 : Virtuoso Conductor – SQL DB Objects

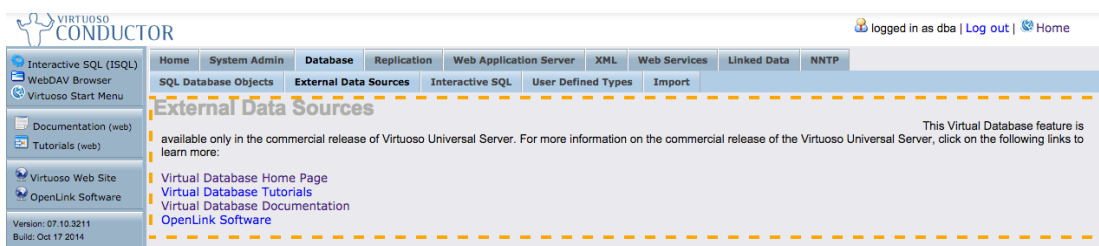


Εικόνα 28 : Virtuoso Conductor – Schema Objects

- Υποκατηγορία External Data Sources

Πρόκειται για ένα από τα πιο δυνατά σημεία του universal server. Είναι η δυνατότητα ενοποίησης της πληροφορίας σε ανώτερο λογικό επίπεδο μεταξύ του server με άλλες εξωτερικές πηγές όπως άλλες εξωτερικές DB δεδομένων. Η λειτουργία για federated search, δίνεται μέσω αυτής της δυνατότητας. Δυστυχώς όμως, η δυνατότητα παρέχεται μόνο στην εμπορική έκδοση του λογισμικού και όχι στην Openlink source Edition που έχουμε

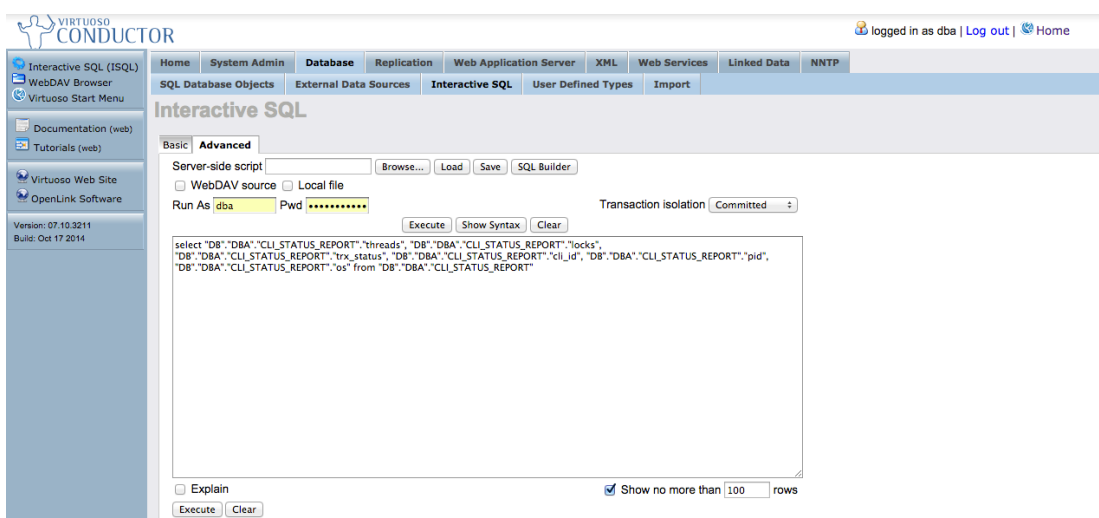
εγκαταστήσει. Μπορούμε να συνδέσουμε, πέραν των συστημάτων DB και απλό αδόμητο κείμενο (free text) και XML, RDF τύπους κοκ.



Εικόνα 29 : Virtuoso Conductor – External Data Sources

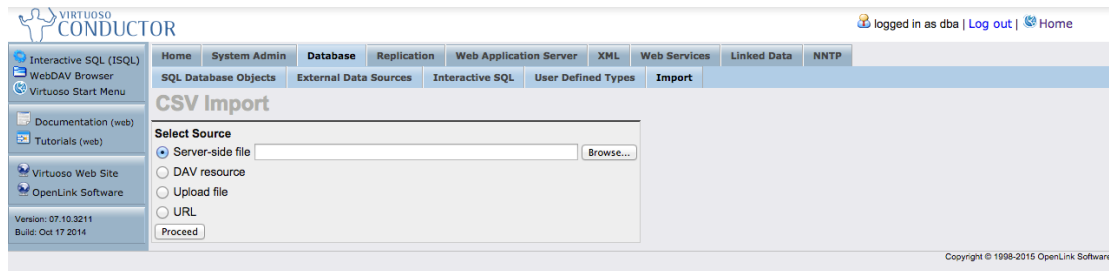
- Υποκατηγορία Interactive SQL

Πρόκειται για την δυνατότητα αμφίδρομης επικοινωνίας με το σύστημα βάσης δεδομένων μέσω SQL εντολών. Είθισται να είναι το πρωταρχικό μέρος, για την παραμετροποίηση του universal server.



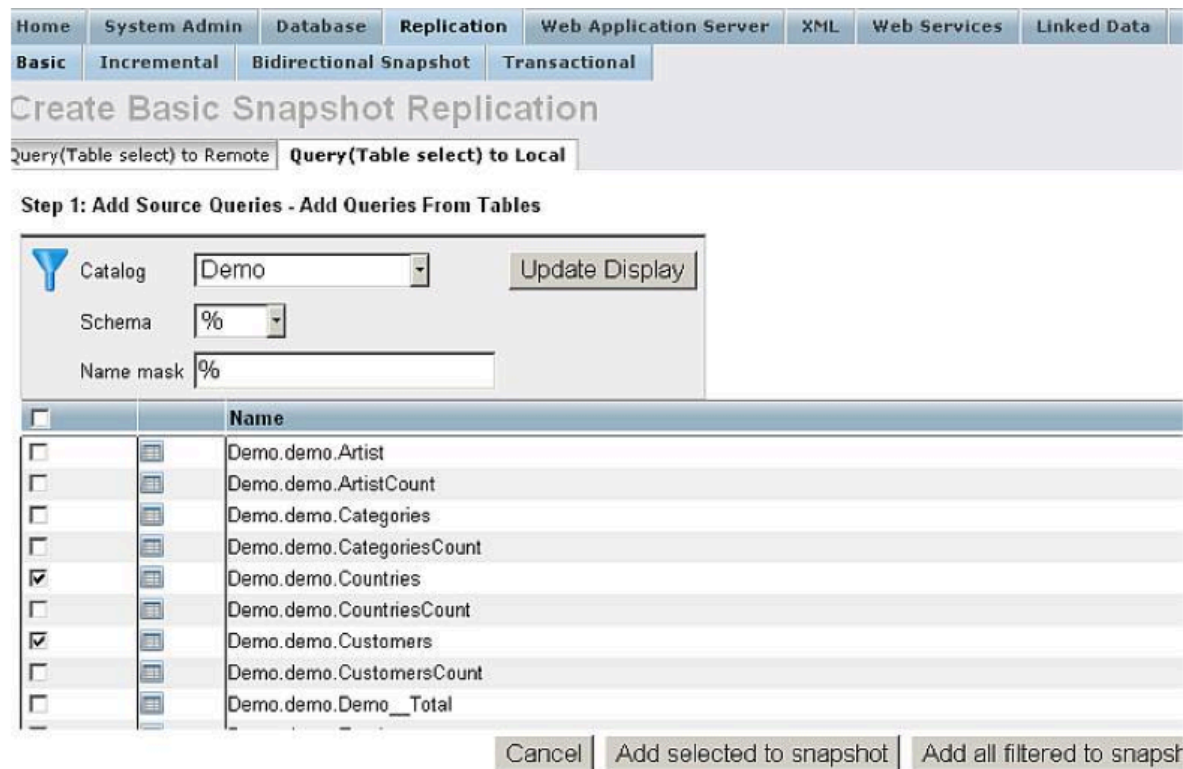
Εικόνα 30 : Virtuoso Conductor – Advanced Interactive SQL

Ακολουθούν οι επιλογές για δημιουργία τύπου δεδομένων ορισμένο από τον χρήστη και η εισαγωγή δεδομένων μέσω CSV (comma separated) files.



Εικόνα 31 : Virtuoso Conductor – Import raw data - cvs

Επόμενη κατηγορία είναι το replication. Αυτό το χαρακτηριστικό υπάρχει μόνο στις εμπορικές εκδόσεις του virtuoso universal server και αποτελεί βασικό προτέρημα στην επιλογή του universal server, όπου δίνεται η ικανότητα συγχρονισμού τοπικών schemas με απομακρυσμένα schemas. Μπορούμε δηλαδή να κάνουμε ακριβή αντίγραφα των τοπικών ΒΔ σε απομακρυσμένους server. Η διαδικασία μπορεί να γίνει χρονοπρογραμματισμένα είτε άμεσα. Υποστηρίζει τον ταυτόχρονο συγχρονισμό – bidirectional, αρχείων καταγραφών κινήσεων transactions, με δυνατότητες rollback σε χρονοθυρίδες.

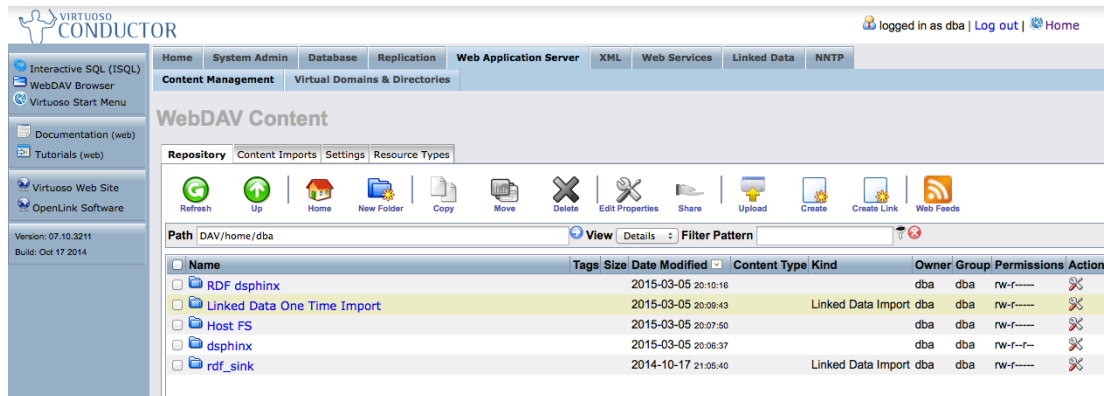


Snapshot Replication

Εικόνα 32 : Virtuoso Conductor – Replication (Universal Server)

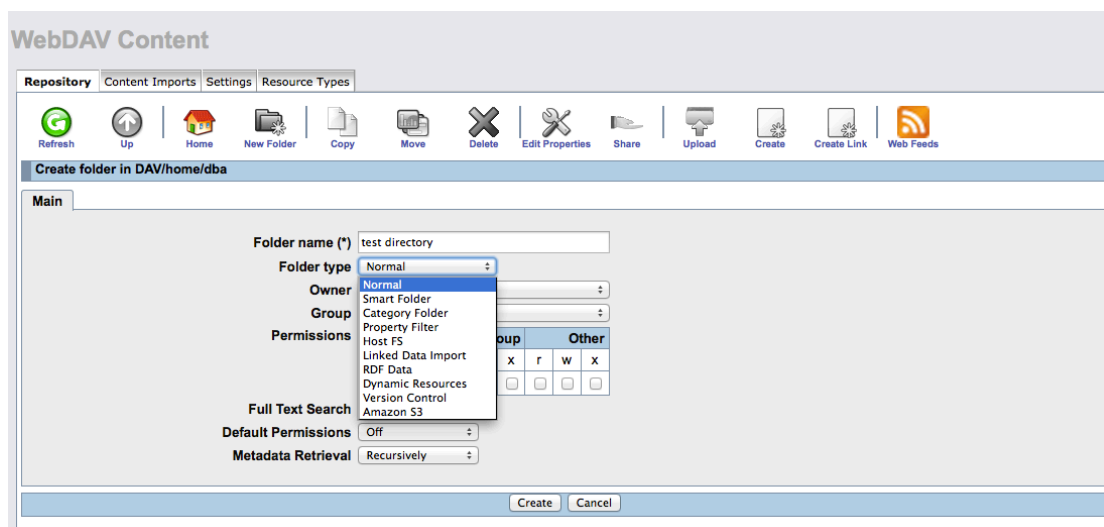
- Web Application Server

Στη παρούσα καρτέλα, γίνεται η διαχείριση αποθετηριών WebDAV για κάθε χρήστη webdav. Ο προκαθορισμένος φάκελος rdf_sink είναι το αποθέτηριο για τα RDF / OWL data που θα μπορεί ο εκάστοτε χρήστης να ανεβάσει και να τα χρησιμοποιήσει απο το SPARQL Endpoint.



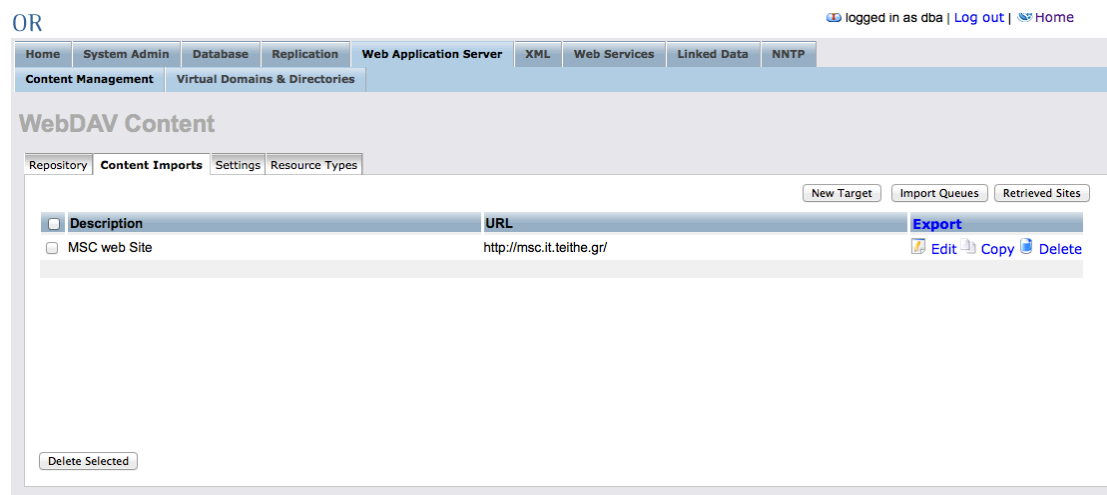
Εικόνα 33 : Virtuoso Conductor – WebDAV Contents

Εδώ μπορούμε να δημιουργήσουμε διαφορετικές κατηγορίες φακέλων που θα αλληλεπιδρούν με τις διάφορες υπηρεσίες διαμέσου αυτών. Οι κατηγορίες μπορούν να είναι απλοί φάκελοι, φάκελοι αποθέτηρια RDF, φάκελοι που έχουν δεδομένα απο συγχρονισμούς ανοικτών δεδομένων (open link data), φάκελοι με δυναμικά αποτελούμενο περιεχόμενο απο συγκεκριμένα URIs, amazon S3 φάκελοι κοκ.



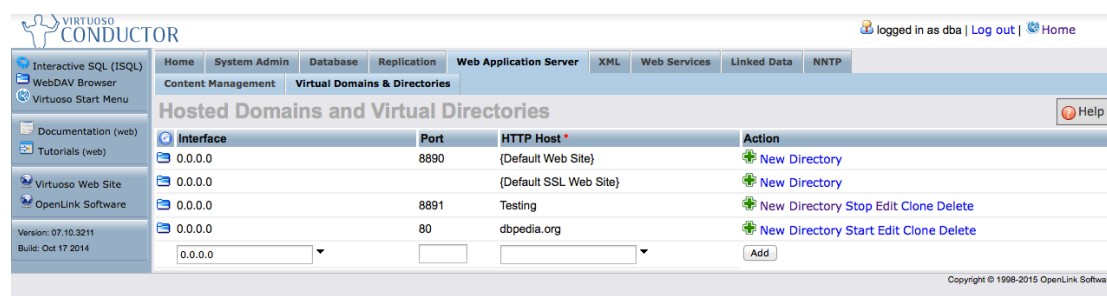
Εικόνα 34 : Virtuoso Conductor – WebDAV Settings

Δίνεται επίσης η δυνατότητα εισαγωγής data απο συγκεκριμένα URI, όπως φαίνεται παρακάτω. Στην υποκαρτέλα Resource Types, έχουμε τη δυνατότητα να δηλώσουμε τους τύπους αρχείων (resource type) που θα διαχειρίζεται ο server κατά την εκτέλεση στο συγκεκριμένο URI.



Εικόνα 35 : Virtuoso Conductor – WebDAV Settings

Στη υποκατηγορία Virtual Domains & Directories, δίνεται η παραμετροποίηση για τα virtual domain, services που θα υποστηρίξει ο HTTP Server. Ως επιλογή για κάθε δυνατή IP διεύθυνση που θα ασχοληθεί ο server λογίζεται η 0.0.0.0.

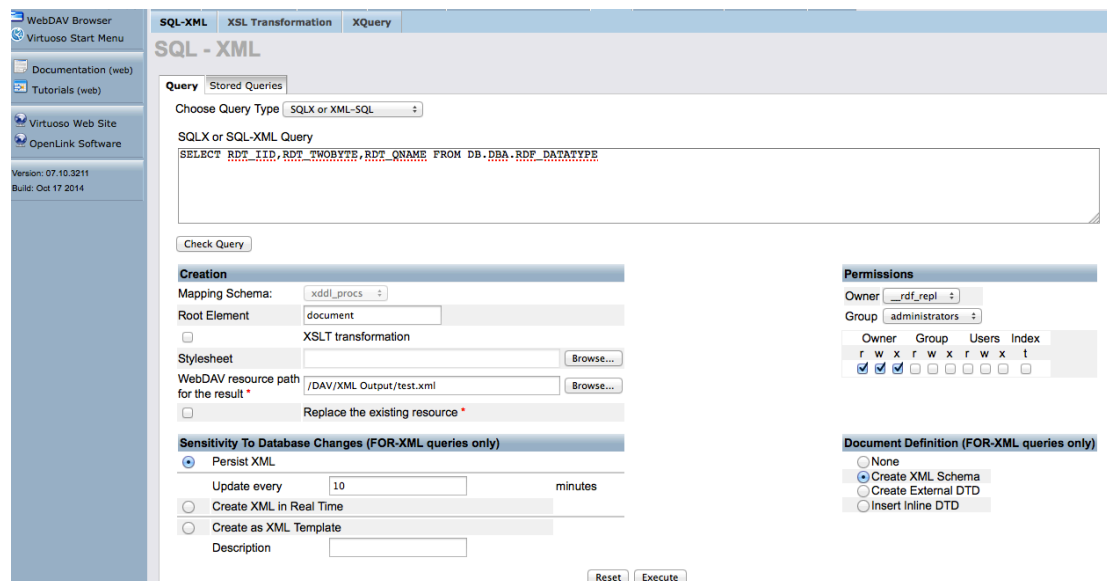


Εικόνα 36 : Virtuoso Conductor – Web Application Server

- Καρτέλα XML

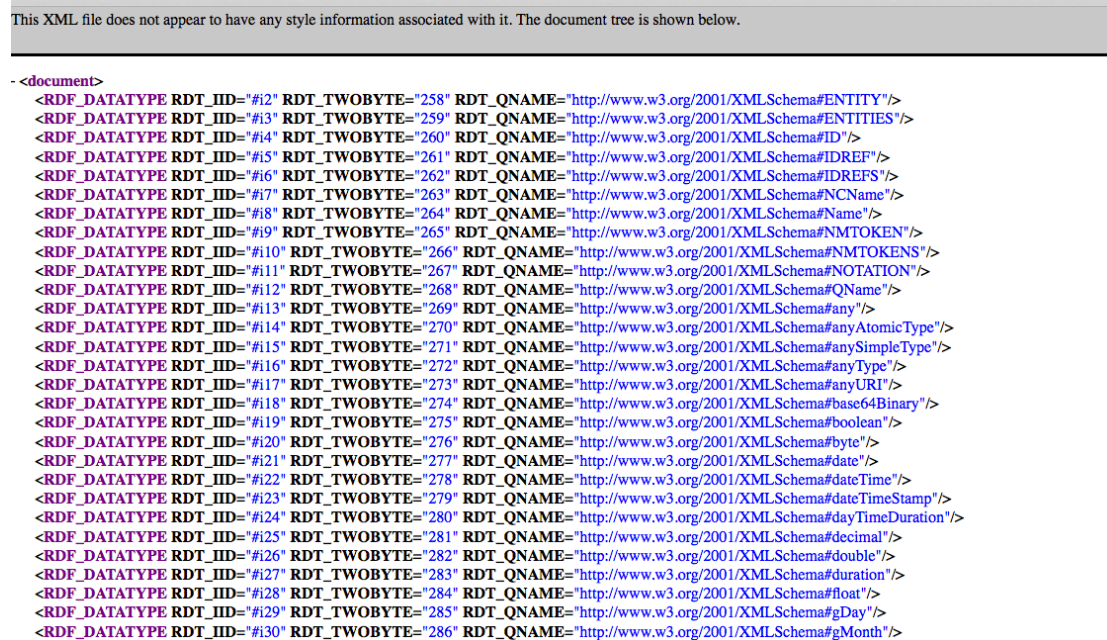
Διαμέσου της καρτέλας, μπορούμε να διαχειριστούμε όλη τη βάση δεδομένων εξάγοντας σε μορφή XML. Μπορούμε εκτελώντας SQLX or SQL-XML Query στη ΒΔ, να παράγουμε τα XML αρχεία ως το αποτέλεσμα αυτών. Αυτό μπορεί να γίνει είτε άμεσα είτε έμμεσα μέσω χρονοπρογραμματισμού ή στην on the fly λειτουργία. Δίνεται η δυνατότητα να

αποθηκεύσουμε και τα XML queries σε οποιαδήποτε WebDAV φάκελο για περαιτέρω χρήση όπως για XML Mapping Schema, XSLT Transformation – Style sheet XML.



Εικόνα 37 : Virtuoso Conductor – SQL XML settings

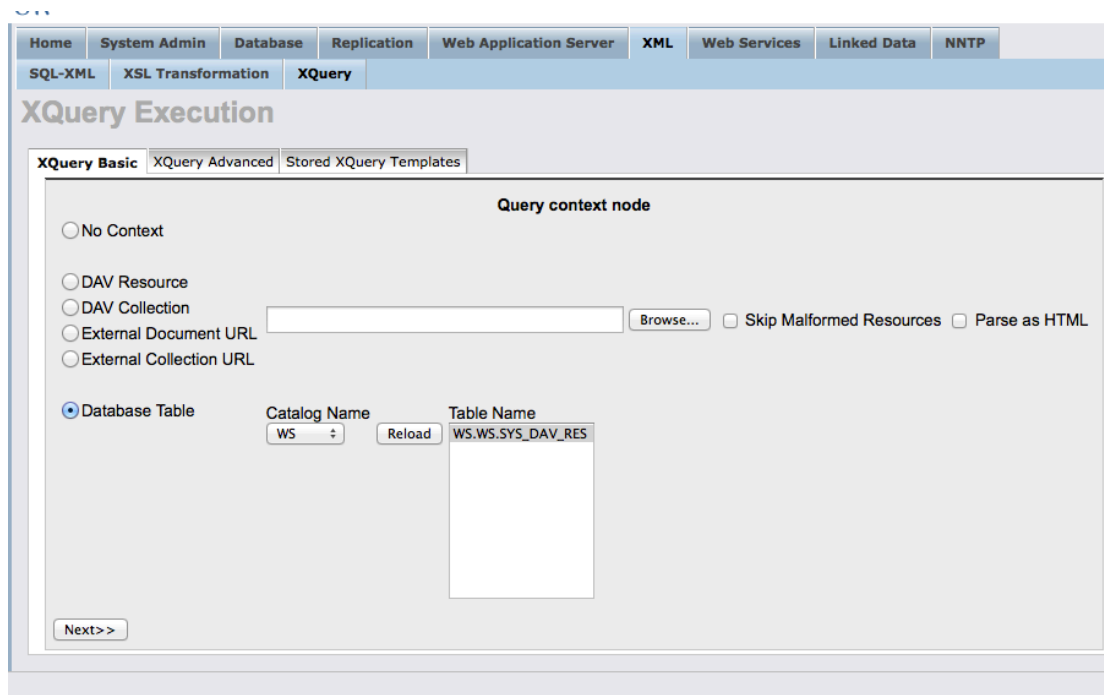
δίνοντας μας το αποτέλεσμα σε XML :



Εικόνα 38 : Virtuoso Conductor – SQL XML settings output

Υποκαρτέλα Xquery :

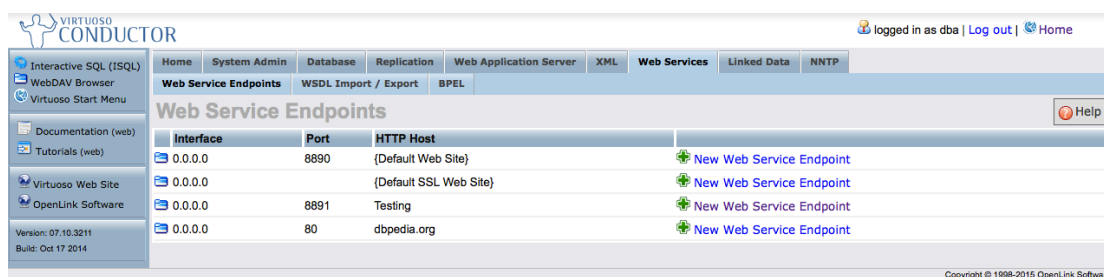
Μπορούμε να διατυπώσουμε ερωτήματα σε προϋπάρχοντα XML αρχεία, αποθηκευμένα XML queries, HTML documents, ακόμη και βάσεις δεδομένων. Μέσω του Xquery και των αποθηκευμένων XML data από κάθε διακριτή υπηρεσία στο WebDAV μπορούμε να έχουμε μια εικόνα για το κάθε αποτέλεσμα και για πάσης φύσεως ερωτήσεις.



Εικόνα 39 : Virtuoso Conductor –XML - Xquery

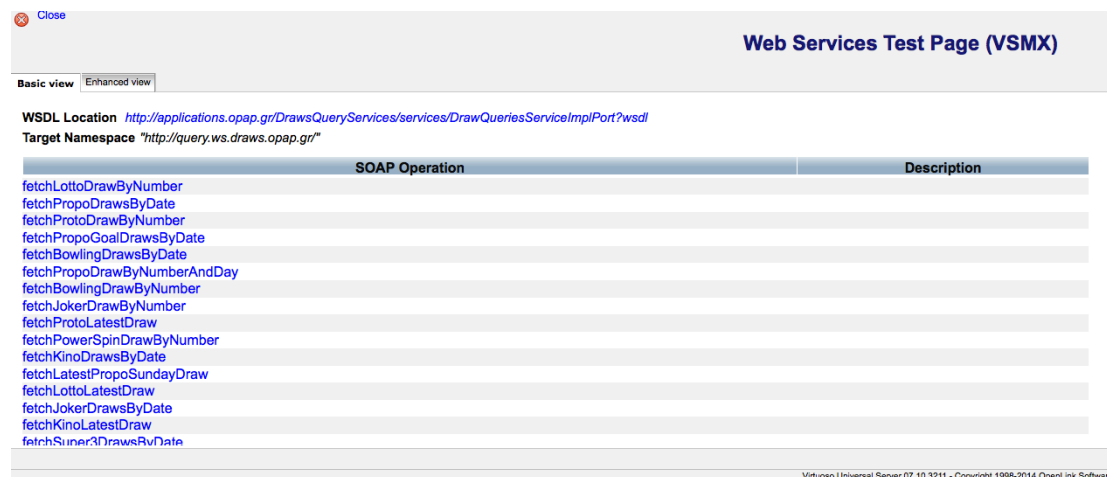
▪ Web Services

Αποτελούν τον ακρογωνιαίο λίθο του universal server. Μπορούμε να δημιουργήσουμε είτε δικές μας υπηρεσίες μέσω απλών SQL Stored procedures είτε διαμορφώσουμε κατάλληλα τις ήδη υπάρχουσες. Στη δεσμευμένη TCP port 8890, εκτελείται το SPARQL Endpoint Web Service έχοντας κύριο σκοπό τη εξυπηρέτηση SPARQL ερωτημάτων.



Εικόνα 40 : Virtuoso Conductor –Web Services

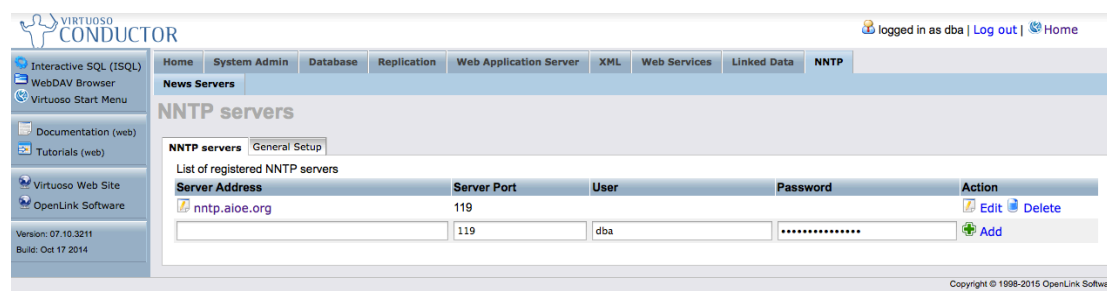
Μέσω WSDL Import, μπορούμε να εισάγουμε ως data στο σύστημα μας σε συγκεκριμένο schema πληροφορίες από άλλα εξωτερικά Web Services.



Εικόνα 41 : Virtuoso Conductor –Web Services settings

▪ NNTP

Το virtuoso universal server, μπορεί να λειτουργήσει ως συλλέκτης (aggregator) NNTP news από διαφορετικές πηγές, να ενώσει σε λογικό επίπεδο τις πληροφορίες και να τις διαθέσει αυτός ως NNTP server. Η πληροφορία συγκέντρωσης, είναι αρκετά μεγάλη σε όγκο και απολύτως χρήσιμη για τομείς όπως eBusiness Intelligence.

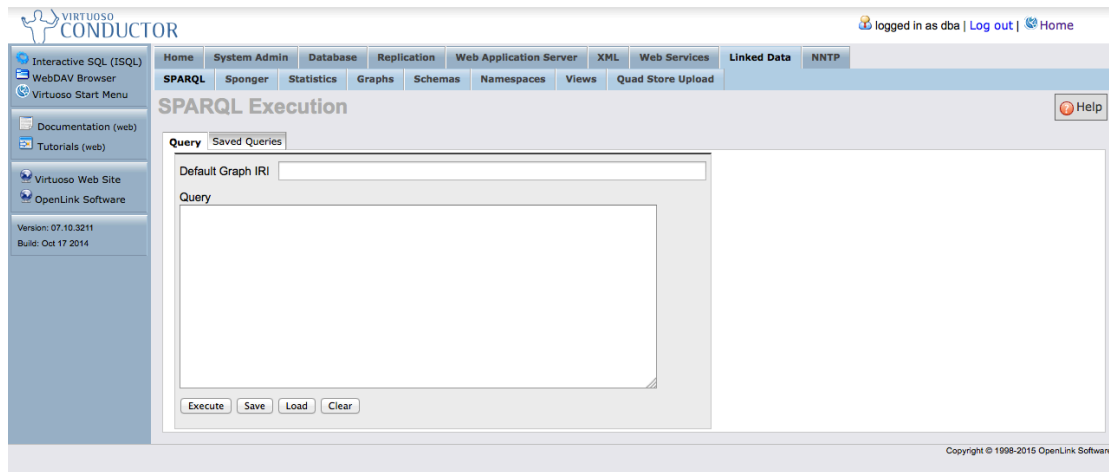


Εικόνα 42 : Virtuoso Conductor –NNTP Servers Settings

▪ Καρτέλα Linked Data

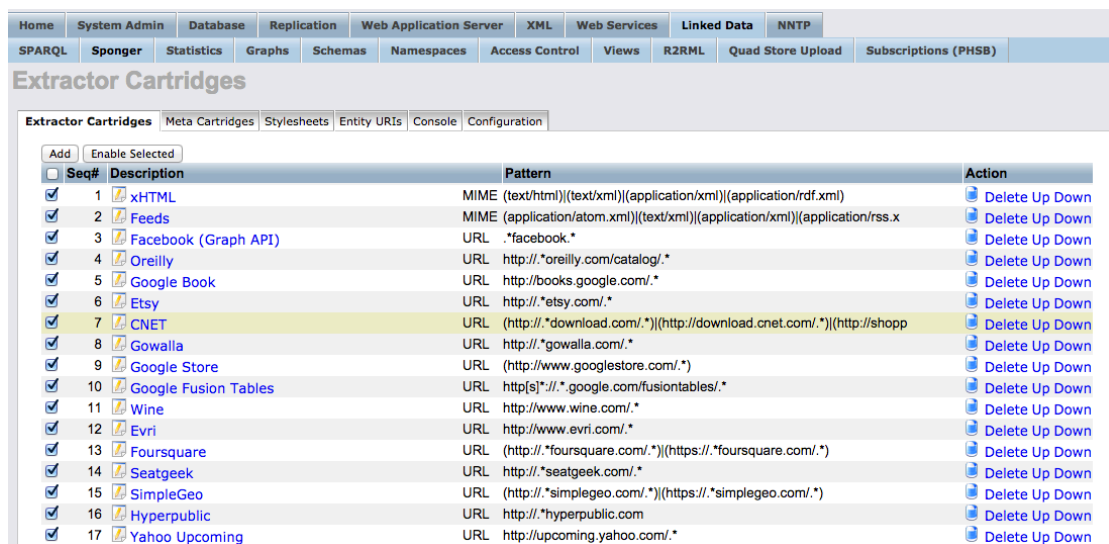
Το επιθυμητό χαρακτηριστικό του σημασιολογικού ιστού, υλοποιείται με τη διασύνδεση των δεδομένων. Μέσω αυτής της καρτέλας, ρυθμίζεται ο server για τη συμμετοχή του με άλλες πηγές ανοικτών ή μη δεδομένων.

Στην υποκατηγορία SPARQL, μπορούμε απευθείας να εκτελέσουμε ερωτήματα SPARQL σε συνδεδεμένα δεδομένα.



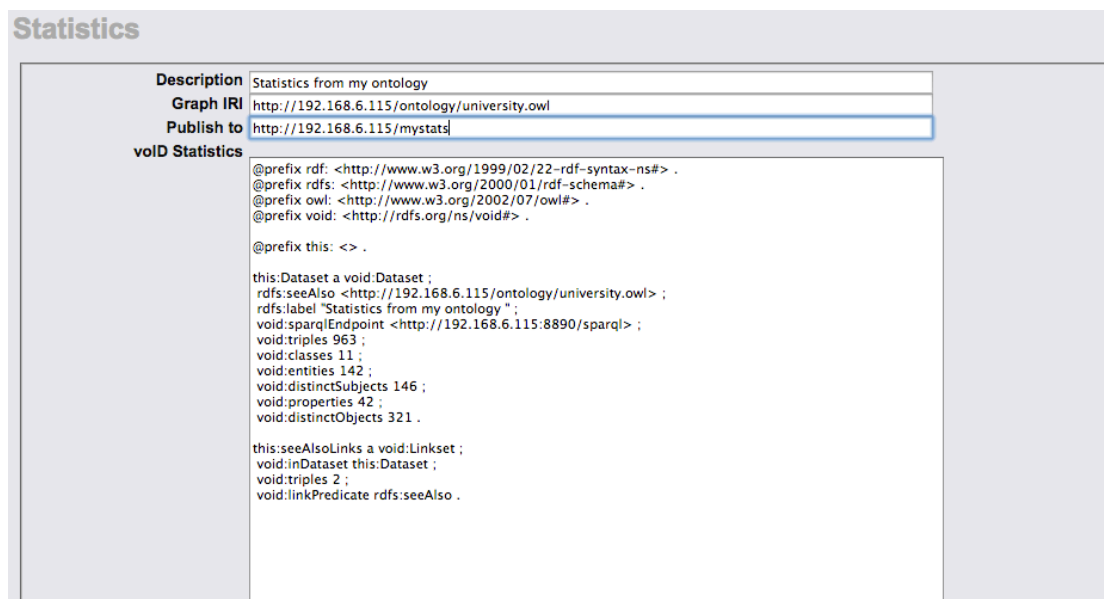
Εικόνα 43 : Virtuoso Conductor – SPARQL Execution – Linked Data

Το Sponger είναι ένα RDF middleware που σκοπός του είναι η μετατροπή non-RDF δεδομένα σε RDF [50]. Στο Sponger, μπορούμε να συσχετίσουμε URIs με βασικές ρυθμίσεις για τη σύνδεση σε αυτά όπως API Key, meta tags κοκ.



Εικόνα 44 : Virtuoso Conductor – Sponger, Linked Data Cartridges Extractor

Στην υποκατηγορία Statistics, μας δίνονται στατιστικά για συγκεκριμένα GRAPH IRI :



Statistics

Description Statistics from my ontology

Graph IRI http://192.168.6.115/ontology/university.owl

Publish to http://192.168.6.115/mystats

void Statistics

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl: <http://www.w3.org/2002/07/owl#> .
@prefix void: <http://rdfs.org/ns/void#> .

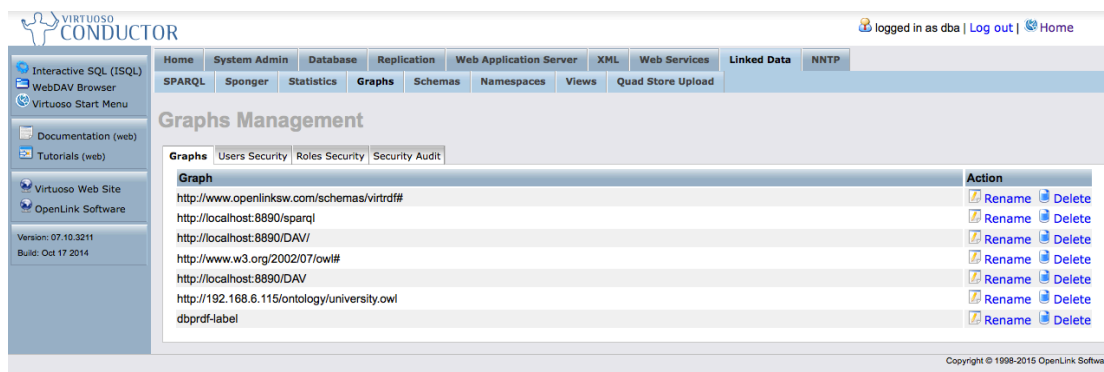
@prefix this: <> .

this:Dataset a void:Dataset ;
rdfs:seeAlso <http://192.168.6.115/ontology/university.owl> ;
rdfs:label "Statistics from my ontology " ;
void:sparqlEndpoint <http://192.168.6.115:8890/sparql> ;
void:triples 963 ;
void:classes 11 ;
void:entities 142 ;
void:distinctSubjects 146 ;
void:properties 42 ;
void:distinctObjects 321 .

this:seeAlsoLinks a void:Linkset ;
void:inDataset this:Dataset ;
void:triples 2 ;
void:linkPredicate rdfs:seeAlso .
```

Εικόνα 45 : Virtuoso Conductor – Linked Data GRAPH Statistics

Υποκατηγορία Graphs, για διαχείριση των προκαθορισμένων SPARQL Graphs όπως και ρυθμίσεις ασφαλείας για αυτά.



VIRTUOSO CONDUCTOR logged in as dba | Log out | Home

Home System Admin Database Replication Web Application Server XML Web Services **Linked Data** NNTP

SPARQL Sponger Statistics **Graphs** Schemas Namespaces Views Quad Store Upload

Graphs Management

Graphs Users Security Roles Security Security Audit

Graph	Action
http://www.openlinksw.com/schemas/virtrdf#	Rename Delete
http://localhost:8890/sparql	Rename Delete
http://localhost:8890/DAV/	Rename Delete
http://www.w3.org/2002/07/owl#	Rename Delete
http://localhost:8890/DAV	Rename Delete
http://192.168.6.115/ontology/university.owl	Rename Delete
dbprdf-label	Rename Delete

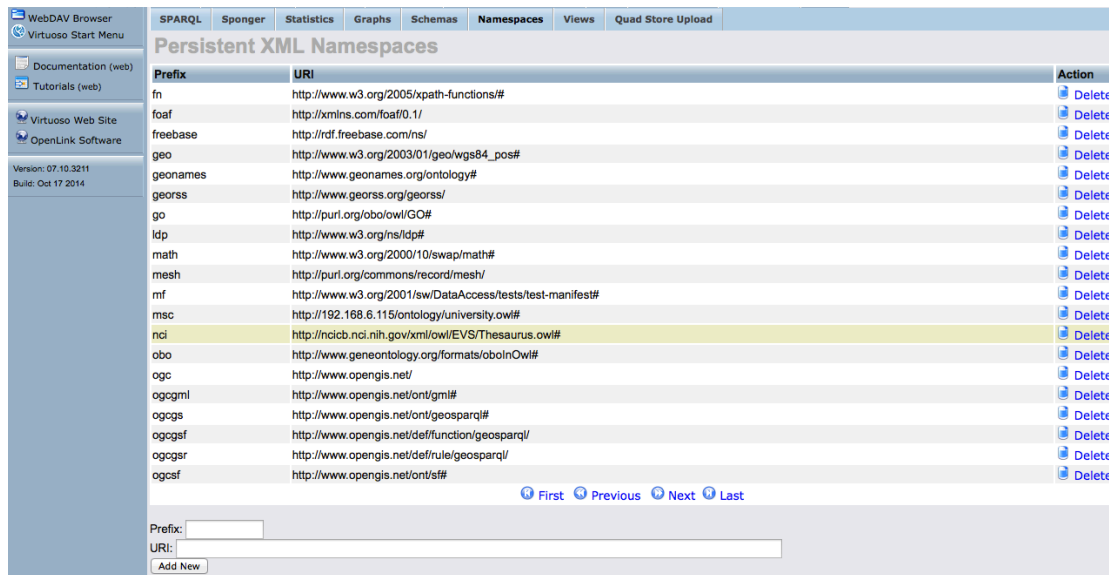
Version: 07.10.3211
Build: Oct 17 2014

Copyright © 1998-2015 OpenLink Software

Εικόνα 46 : Virtuoso Conductor – Linked Data – Graphs Managements

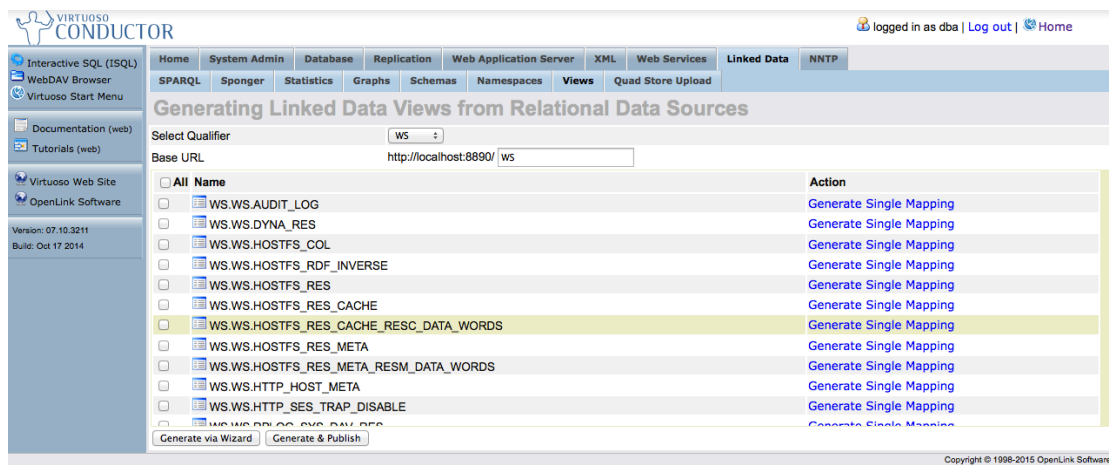
Υποκατηγορία Schemas, μπορούμε να εισαγάγουμε δεδομένα απο εξωτερικά Schema IRI.

Στην υποκατηγορία Namespaces, μπορούμε να προσθέσουμε Namespaces για σύνδεση μέσω SERVICE SPARQL είτε να θέσουμε το προκαθορισμένο namespace για τα δικά μας SPARQL ερωτήματα.



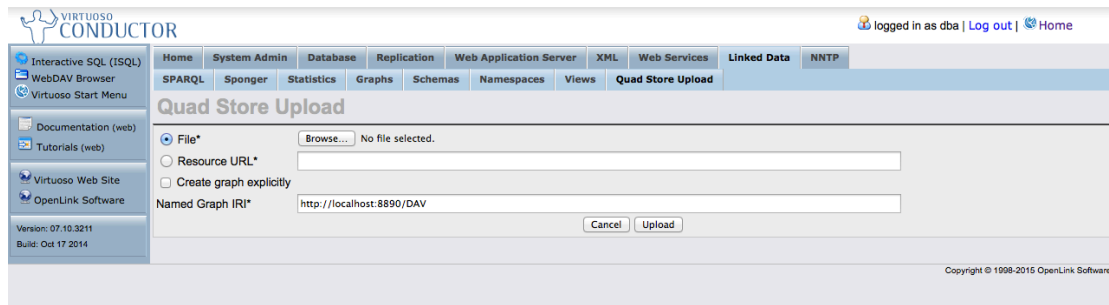
Εικόνα 47 : Virtuoso Conductor – Namespaces

Υποκατηγορία Views, δημιουργία linked data από τη εσωτερική σχεσιακή βάση δεδομένων. Αποτελεί ισχυρό χαρακτηριστικό του virtuoso universal server, που μπορεί να εξαγάγει πληροφορία απο RDBMS σε RDF dataset (R2RML).



Εικόνα 48 : Virtuoso Conductor – Linked Data Views with RDBMS Sources

Υποκατηγορία Quad Store, το σημείο εισαγωγής δεδομένων/οντολογίας στο δικό μας Endpoint σε συγκεκριμένο Graph. Για τη δημιουργία, θα πρέπει να ανεβάσουμε το αρχείο δεδομένων στο WebDAV, και συγκεκριμένα στο tmp ούτως ώστε να είναι ορατό απο όλους για παράδειγμα http://sparql_endpoint_ip:8890/DAV/tmp. Η μέθοδος αποτελεί τον δεύτερο τρόπο εισαγωγής dataset στο σύστημα Virtuoso.



Εικόνα 49 : Virtuoso Conductor – Linked Data - Quad Store

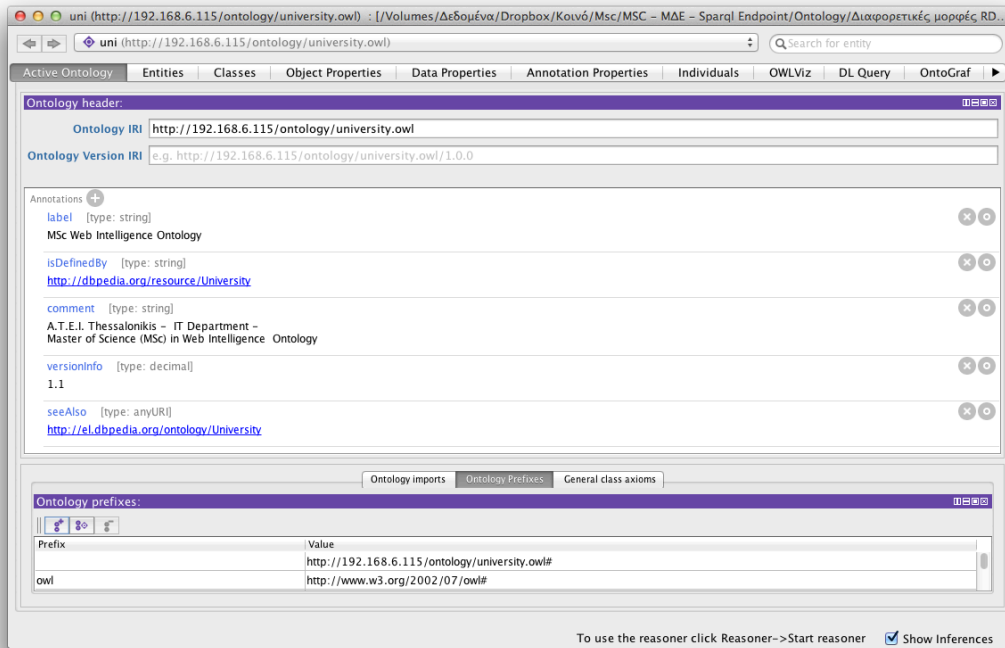
6.2 Protégé

Πρόκειται για ένα ολοκληρωμένο περιβάλλον δημιουργίας οντολογιών με πλούσια χαρακτηριστικά και πληθώρα προσθέτων (plugins). Υφίσταται σε δυο (2) εκδόσεις, την native desktop έκδοση [80] που είναι υλοποιημένη σε γλώσσα java και λειτουργεί σε πληθώρα λειτουργικών συστημάτων και την on-line έκδοση web protégé που προσφέρεται για συνεργατική συγγραφή με ομάδα συμμετέχοντων.

Περιέχει συντάκτη οντολογιών για όλα τα γνωστά μορφότυπα οντολογιών OWL [1] [2]. Διαθέτει plug-ins για νέες λειτουργίες όπως για την αυτοματοποιημένη συλλογιστική (reasoning), για τον έλεγχο της ορθότητας της οντολογίας που δημιουργούμε. Δημοφιλείς plug-ins reasoners είναι οι FaCT++, Hermit, Pellet [1] [69].

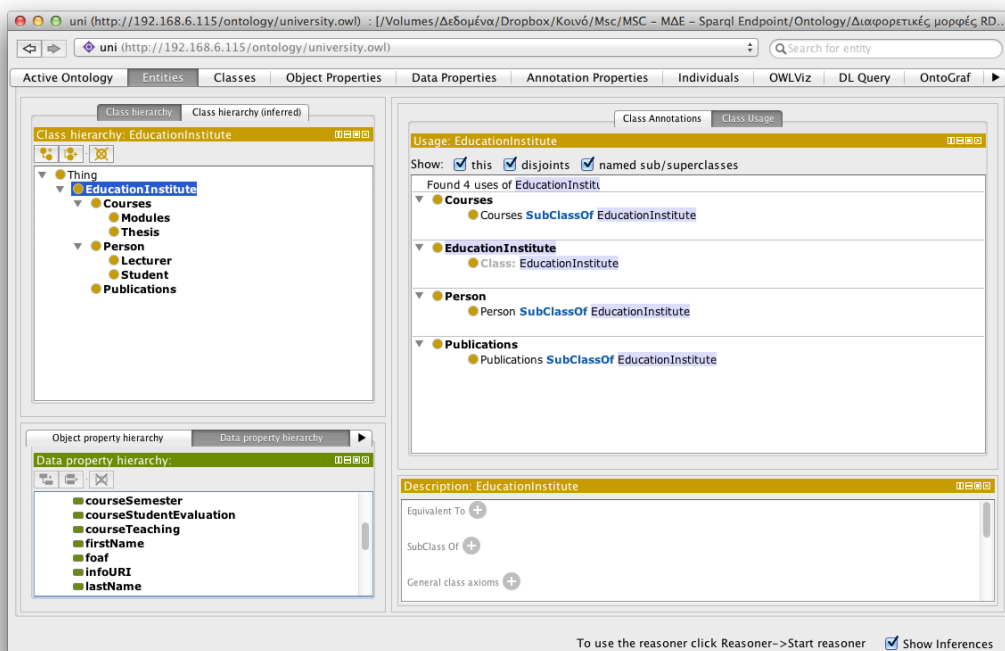
Η διεπαφή μας παρέχει μια σειρά από καρτέλες με διακριτές λειτουργίες, σημαντικότερες είναι οι εξής :

- Active Ontology – Εδώ δίνουμε τις πληροφορίες για το σχηματισμό της κεφαλίδας όπως η έκδοση, ο τίτλος της οντολογίας, το URI, seeAlso. Δίνεται επίσης η δυνατότητα να κάνουμε εισαγωγή (Ontology import) μιας άλλης εξωτερικής οντολογίας για να συσχετιστεί με την δική μας οντολογία (διασυνδεδεμένα δεδομένα). Να εισαγάγουμε νέα Ontology Prefixes που θα χρειαστεί η οντολογία και να δημιουργήσουμε δικά μας αξιώματα. Στιγμιότυπο από την οντολογία φαίνεται στην εικόνα 50 :



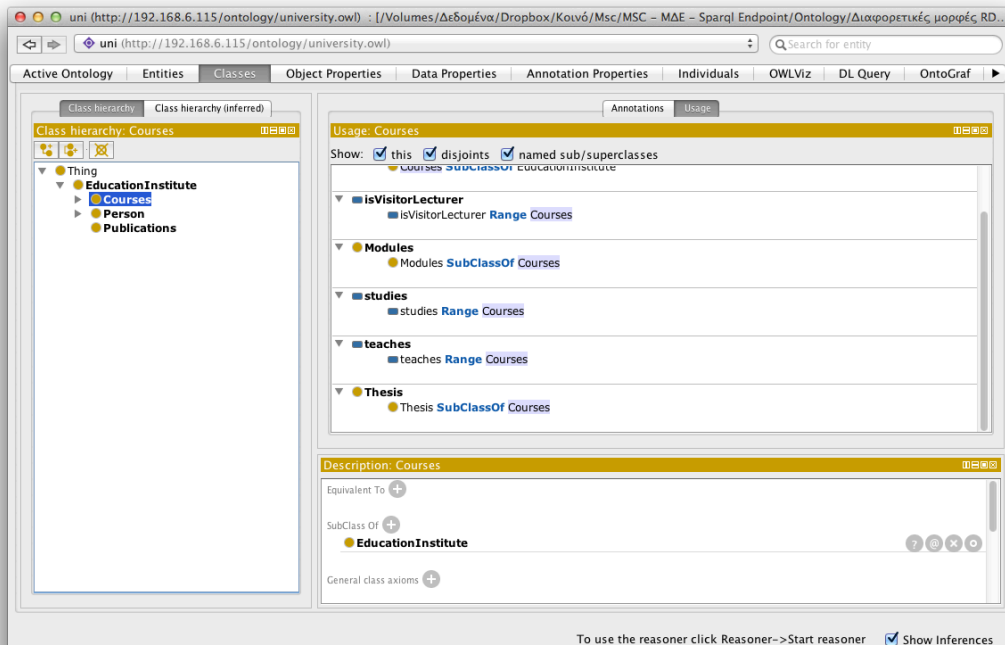
Εικόνα 50 : protégé – Active Ontology

- Entities – Απεικόνιση των οντοτήτων σε ιεραρχικό μοντέλο, συνοψίζοντας και τις πληροφορίες αυτών.



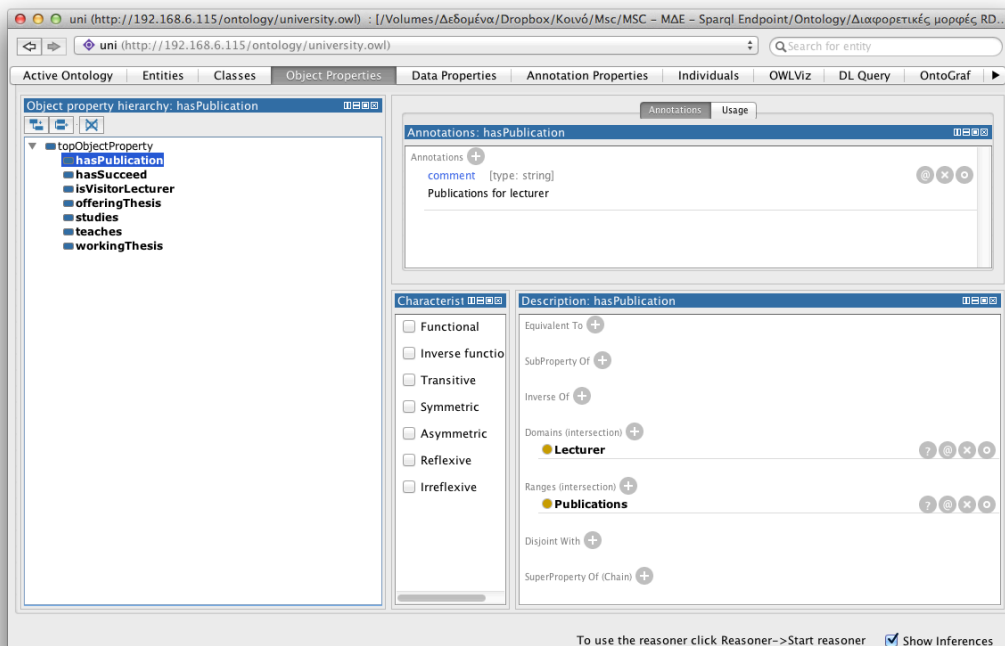
Εικόνα 51 : protégé – Entities

- Classes – Απεικόνιση των υπάρχουσών κλάσεων, μαζί με τα object properties



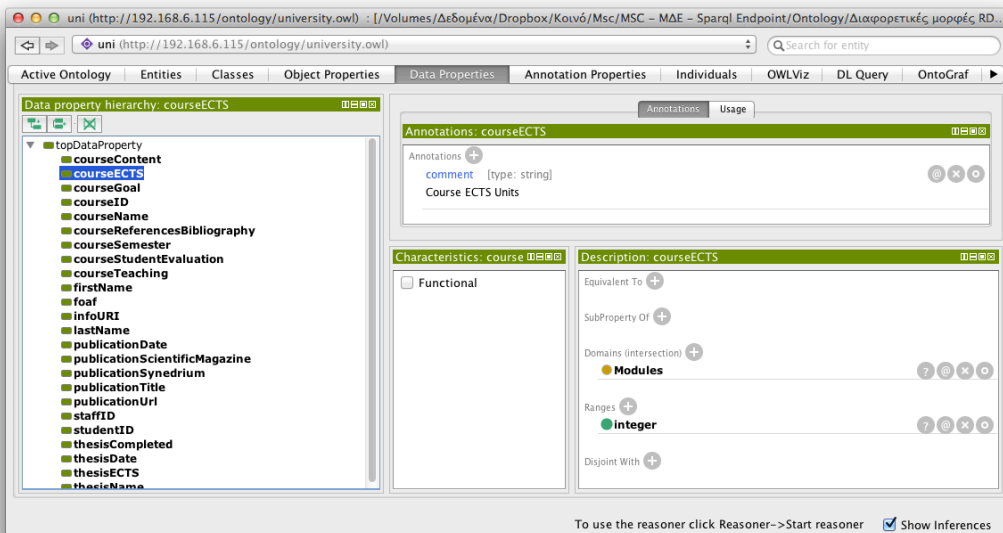
Εικόνα 52 : protégé – Classes

- Object Properties – Απεικόνιση των ιδιοτήτων αντικειμένων



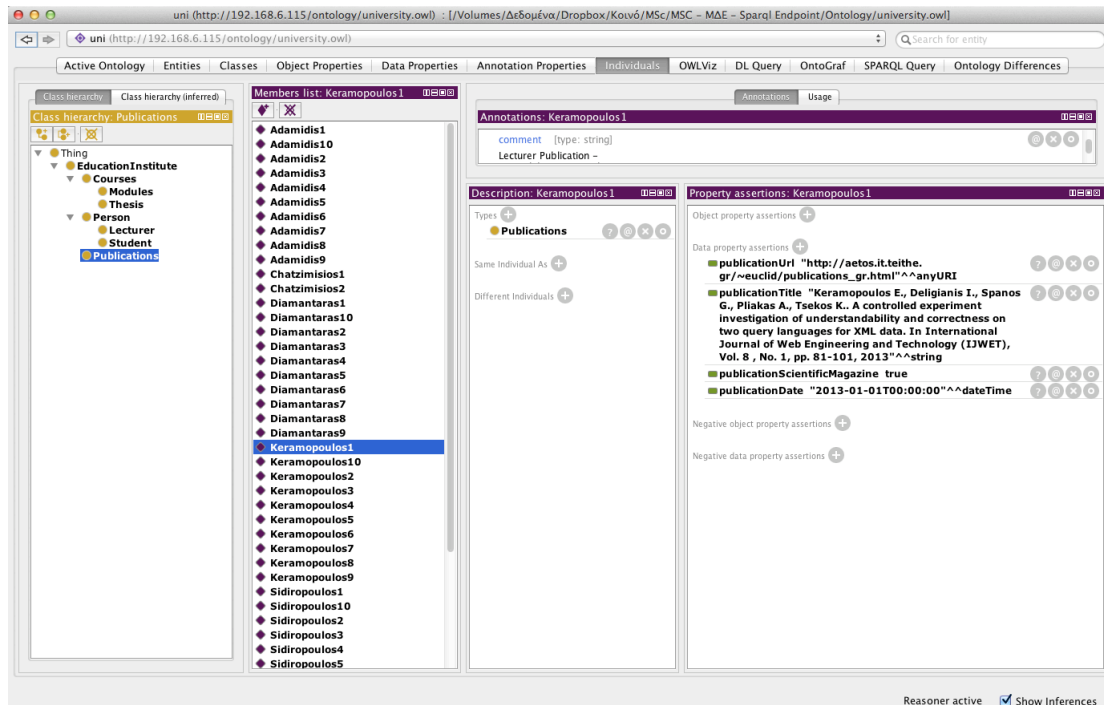
Εικόνα 53 : protégé – Object Properties

- Data Properties – Απεικόνιση ιδιοτήτων τύπου δεδομένων



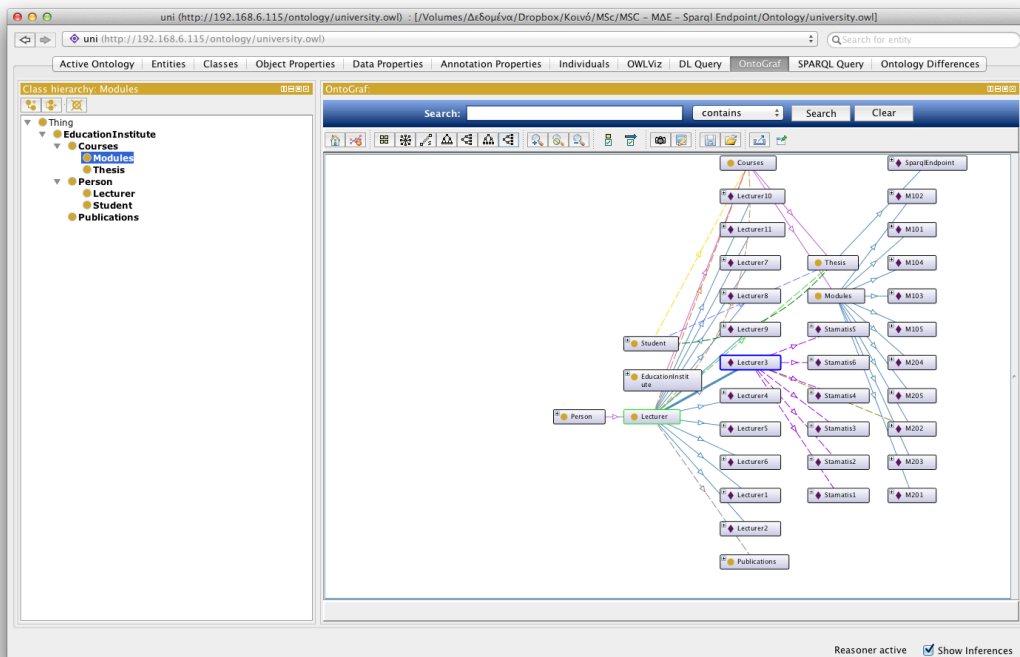
Εικόνα 54 : protégé – Data Properties

- Individuals – Στιγμιότυπα της οντολογίας μαζί με τους τύπους δεδομένων



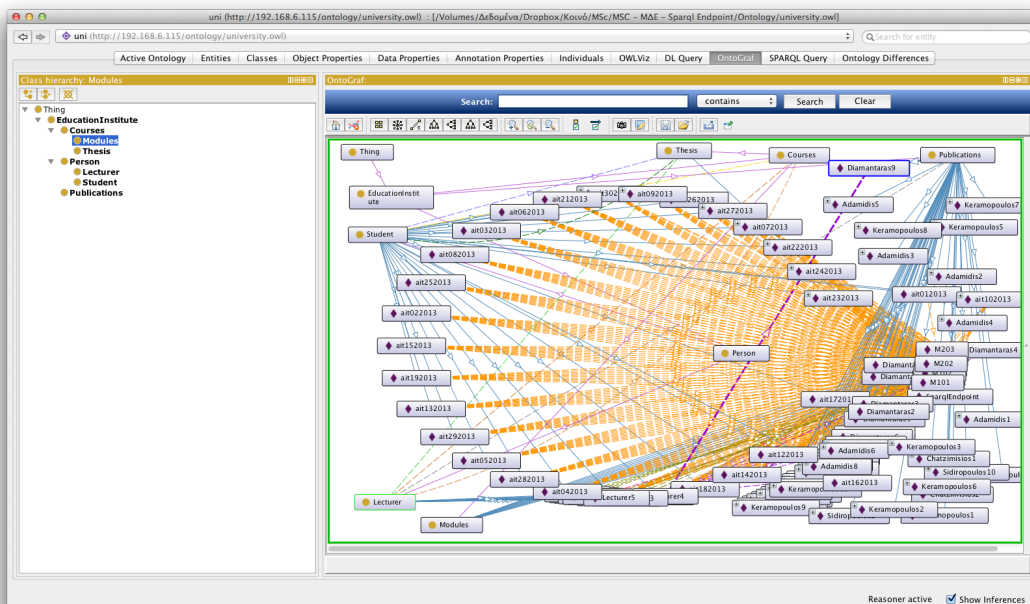
Εικόνα 55 : protégé - Individuals

- **Ontograf** – Απεικόνιση επιλεγμένων κλάσεων με τις συσχετίσεις των



Εικόνα 56 : *protégé* - *Ontograf*

και μια απεικόνιση της δημιουργηθείσας οντολογίας



Εικόνα 57 : *protégé* - *Ontograf* συνολική οπτική απεικόνιση

6.3 Web Εφαρμογή

Πρόκειται για μια web εφαρμογή, που σκοπός της είναι ο έλεγχος της ορθότητας λειτουργίας της οντολογίας και του SPARQL Endpoint.

Χρησιμοποιεί τις τεχνολογίες PHP, XML, HTML [90], BOOTSTRAP CSS.

Τα επιθυμητά SPARQL ερωτήματα είναι αποθηκευμένα σε XML μορφή, διαβάζονται από την PHP και διαμορφώνονται για αισθητικούς λόγους (indent, colorize) στην εξαγόμενη HTML μορφή. Με την εκτέλεση του ερωτήματος, καλείται το SPARQL Endpoint και μας αποδίδει τη πληροφορία στην προεπιλεγμένη της μορφή. Η μορφή του web interface φαίνεται στις παρακάτω εικόνες,

SPARQL + Ontology

SPARQL Endpoint ISPARQL Endpoint Conductor

PREFIX msc: <http://192.168.6.115/ontology/university.owl#>

1) Ποια είναι τα μαθήματα για το MSc ;

```
SELECT
  str(?e) as ?course ?ECST ?semester
WHERE
  { ?s msc:courseECTS ?ECST. ?s msc:courseName ?e. ?s msc:courseSemester ?semester. }
```

Εκτέλεση SPARQL Ερωτήματος

Εικόνα 58 : web app – SPARQL queries listing

course	ECST	semester
Research Methods	6	1
Software Engineering for Web Applications	6	1
Internet Security	6	1
Mobile and Ubiquitous Computing Systems	6	1
Social Networking	6	1
Data Warehousing and Data Mining	6	2
Artificial Intelligence Techniques – Agents	6	2
Machine Learning	6	2
Internet Information Retrieval	6	2
Semantic Web	6	2

Εικόνα 59 : web app – output via virtuoso Endpoint

5) Περισσότερα στοιχεία για τους διδάσκοντες :

```
PREFIX msc: <http://192.168.6.115/ontology/university.owl#>
SELECT
  distinct str(?lastname) as ?surname str(?firstname) as ?name ?id ?homepage ?foafLink GROUP_CONCAT(
    distinct ?publication; separator = ", "
  ) as ?publications GROUP_CONCAT(
    distinct ?courseName; separator = ", "
  ) as ?teaches ?thesis
WHERE
  { ?s msc:firstName ?firstname. ?s msc:teaches ?courseName. ?s msc:lastName ?lastname. ?s msc:staffID ?id.
  OPTIONAL { ?s msc:inFoURI ?url. } ?s ?e msc:Lecturer.msc:teaches ?ae ?course.
  OPTIONAL { ?s msc:foaf ?foaf. }
  OPTIONAL { ?s msc:offeringThesis ?thesis }
  OPTIONAL { ?s msc:hasPublication ?publication }
  bind(
    xsd:string(?foaf) as ?foafLink
  )
  bind(
    xsd:string(?url) as ?homepage
  )
  bind(
    xsd:string(?thesis) as ?thesis
  )
}
ORDER BY
  ?lastname
```

Εκτέλεση SPARQL Ερωτήματος

Εικόνα 60 : web app – SPARQL queries listing

καθώς επίσης και στιγμιότυπα μερικών αποτελεσμάτων.

Diamantaras	Konstantinos	4566	http://aetos.it.teithe.gr/~kdiamant/		http://192.168.6.115/ontology/university.owl#Diamantaras4.http://192.168.6.115/ontology/university.owl#Diamantaras5.http://192.168.6.115/ontology/university.owl#Diamantaras6.http://192.168.6.115/ontology/university.owl#Diamantaras7.http://192.168.6.115/ontology/university.owl#Diamantaras8.http://192.168.6.115/ontology/university.owl#Diamantaras9	http://192.168.6.115/ontology/university.owl#M203	
Ilioudis	Christos	1111	http://aetos.it.teithe.gr/~iliou/			http://192.168.6.115/ontology/university.owl#M103	
Keramopoulos	Euclid	4951	http://aetos.it.teithe.gr/~euclid/	http://dblp.13s.de/d2r/resource/authors/Euclid_Keramopoulos	http://192.168.6.115/ontology/university.owl#Keramopoulos1.http://192.168.6.115/ontology/university.owl#Keramopoulos10.http://192.168.6.115/ontology/university.owl#Keramopoulos2.http://192.168.6.115/ontology/university.owl#Keramopoulos3.http://192.168.6.115/ontology/university.owl#Keramopoulos4.http://192.168.6.115/ontology/university.owl#Keramopoulos5.http://192.168.6.115/ontology/university.owl#Keramopoulos6.http://192.168.6.115/ontology/university.owl#Keramopoulos7	http://192.168.6.115/ontology/university.owl#M201 , http://192.168.6.115/ontology/university.owl#M205	http://192.168.6.115/ontology/university.owl#SparqlEndpoint

Εικόνα 61 : web app – output via virtuoso Endpoint

Δίνεται το αρχείο ερωτημάτων [SparqlQueries.xml](#) σε RAW μορφή:

```

<!--
Document created by: Constantinos M. Melisidis, SPARQL Queries for our Endpoint
-->
<SPARQL_Queries>
  <Query>
    <Info>Ποια είναι τα μαθήματα για το MSc ;</Info>
    <Sparql>
      SELECT str(?e) as ?course ?ECST ?semester WHERE { ?s msc:courseECTS ?ECST. ?s msc:courseName ?e. ?s msc:courseSemester ?semester . }
    </Sparql>
  </Query>
  <Query>
    <Info>Υπάρχουν σχόλια περιγραφής , μέσα στην οντολογία ;</Info>
    <Sparql>
      SELECT ?uri ?labelS WHERE { ?uri rdfs:comment ?label. bind( xsd:string(?label) as ?labelIS ) FILTER( CONTAINS( LCASE( str(?uri) ), "http://192.168.6.115/ontology" ) ) } GROUP BY ?uri
    </Sparql>
  </Query>
  <Query>
    <Info>Πόσοι είναι οι διδάσκοντες ;</Info>
    <Sparql>
      PREFIX msc: <http://192.168.6.115/ontology/university.owl#> SELECT count(?s) WHERE { ?s msc:firstName ?firstname. ?s msc:lastName ?lastname. ?s msc:staffID ?id. }
    </Sparql>
  </Query>

```

Εικόνα 62 : XML μορφότυπο για SPARQL ερωτήματα

7

Επίλογος

7.1 Σύνοψη και συμπεράσματα

Στα πλαίσια της μεταπτυχιακής διπλωματικής εργασίας, δημιουργήθηκε η οντολογία που περιγράφει το πρόγραμμα μεταπτυχιακών σπουδών. Δημιουργήθηκε μια πύλη διενέργειας ερωτημάτων SPARQL, με σκοπό τη δημοσίευση της οντολογίας για χρήση στο σημασιολογικό ιστό. Η κατανάλωση αυτής της πληροφορίας μπορεί να γίνει πλέον, είτε από ανθρώπους είτε από μηχανές [6] [95]. Το σενάριο που υιοθετήθηκε, περιγράφει μια μιναλιστική έκδοση της οντολογίας ως προς τις ενθυλακωμένες οντότητες. Η δομή της οντολογίας, προέκυψε ύστερα από ανάλυση της πληροφορίας που διατίθεται στο επίσημο ιστότοπο του ΠΜΣ. Για τη δοκιμή της λειτουργίας, αναπτύχθηκε μια web εφαρμογή με προκαθορισμένα SPARQL ερωτήματα, όπως αναφέρθηκε στον ενότητα 6.3.

Το virtuoso universal server, αποτελεί την απαραίτητη πλατφόρμα για τη συμμετοχή και ολοκλήρωση των δεδομένων μας στο σημασιολογικό ιστό. Πέραν της σημασιολογικής προσέγγισης των δεδομένων από την πλατφόρμα, παρέχει και μια πληθώρα απαραίτητων υπηρεσιών όπως DBMS, LDAP, MAIL, WebDAV [9] [82].

Η ενασχόληση με τη συγκεκριμένη πλατφόρμα virtuoso, μας διδάσκει την ποιοτική υπεροχή που μπορεί να υπάρξει στο λογισμικό όταν συνδέονται de facto web υπηρεσίες ως μια ενοποιημένη υπερ-υπηρεσία [9].

7.2 Μελλοντικές επεκτάσεις

Η κατανάλωση πληροφορίας από ένα SPARQL Endpoint, είναι ελεύθερη και ανοικτή προς όλους εξ ορισμού [6] [78].

Η υλοποίηση ενός μηχανισμού αυθεντικοποίησης για την επιλεκτική προώθηση συγκεκριμένης πληροφορίας για πιο ευαίσθητα δεδομένα, που συγχρόνως να μην αντιτίθεται στον ορισμό των διασυνδεδεμένων δεδομένων [6] [88]. Παραπλήσια λειτουργία ενός DBMS στο λογικό επίπεδο.

Έμμεση υλοποίηση αποτελεί όταν έχουμε ανεβάσει την οντολογία στο WebDAV μέσω Quad Store (όχι στο δημόσιο DAV/tmp) και θα πρέπει να συνδεθούμε μέσω iSPARQL για την εκτέλεση του ερωτήματος. Φυσικά υπάρχουν και διάφορες άλλες υλοποιήσεις σε SPARQL Endpoint, που μπορούν να υποστηρίξουν σύνδεση μέσω openid πιστοποίησης [16] [89].

Υπάρχουν μεγάλα περιθώρια βελτίωσης της οντολογίας μας. Για τη δημιουργία των individuals, χρησιμοποιήθηκε το Protégé και σχεδιάστηκε ξεχωριστά κάθε οντότητα.

Η συνεχόμενη δημιουργία των individuals απαιτεί διαφορετική προσέγγιση, όπως η αυτοματοποιημένη δημιουργία individuals από τις υπάρχουσες βάσεις δεδομένων του μητρώου των φοιτητών. Κάτι τέτοιο θα μπορούσε να επιτευχθεί με την δημιουργία μιας υπηρεσίας που θα μπορούσε να συλλέξει το μητρώο καθενός και την εξαγωγή αυτού να την πρόσθετε στην οντολογία.

Διαφορετική λύση, θα μπορούσε να είναι η υιοθέτηση του virtuoso universal server ως το κατεξοχήν σύστημα βάσης δεδομένων για την εφαρμογή του μητρώου, και μετέπειτα η αυτοματοποιημένη εξαγωγή της ΒΔ σε γράφους RDF.

Υιοθέτηση μηχανήματος μεγαλύτερων δυνατοτήτων με αποκλειστική χρήση ως SPARQL Endpoint. Αυτό θα επιφέρει ταχύτερη απόκριση για κάθε λειτουργία και αποσυμφόρηση των περίπλοκων ερωτημάτων από ταυτόχρονα ερωτήματα.

Μετατροπή και σύνδεση της οντολογίας με άλλες συναφείς οντολογίες για τη δημοσίευση μέρους ή ολόκληρης της οντολογίας στο LOD cloud [9].

Βελτιώσεις στην web εφαρμογή, για σύστημα διαχείρισης χρηστών και διαχείρισης πολλαπλών αρχείων ερωτημάτων XML.

Σύνδεση της web εφαρμογής, με το LDAP server του virtuoso για την αποθήκευση των αρχείων ερωτημάτων.

Βιβλιογραφία

- [1] Γρηγόρης Αντωνίου, Frank van Harmelen, 2009. Εισαγωγή στο σημασιολογικό ιστό, Δεύτερη Αμερικάνικη Έκδοση. Εκδόσεις Κλειδάριθμος.
- [2] Steffen Staab, Rudi Studer, 2009. «Handbook on Ontologies», 2nd Edition, Springer
- [3] SPARQL Query Language for RDF, URI : <http://www.w3.org/TR/sparql11-query/> (Οκτώβριος 2014)
- [4] W3C SPARQL, URI: <http://www.w3c.gr/press/pressreleases/2008/01/sparql-pressrelease.el.html> (Οκτώβριος 2014)
- [5] What the Semantic Web can represent, URI : <http://www.w3.org/DesignIssues/RDFnot.html> (Οκτώβριος 2014)
- [6] Linked Data, URI : <http://www.w3.org/DesignIssues/LinkedData.html> (Οκτώβριος 2014)
- [7] The Enduring Myth of the SPARQL Endpoint, URI: <http://daverog.wordpress.com/2013/06/04/the-enduring-myth-of-the-sparql-endpoint/> (Οκτώβριος 2014)
- [8] Ανάπτυξη Εφαρμογών με Ανοικτά Δεδομένα και RESTful APIs: Η υπηρεσία OpenDataCloud, URI: <https://ellak.gr/2014/11/anaptixi-efarmogon-me-anikta-dedomena-ke-restful-apis-ipiresia-opensdatacloud/> (Οκτώβριος 2014)
- [9] Virtuoso Universal Server, URI : <http://virtuoso.openlinksw.com/whitepapers/A-Super-Platform.html> (Οκτώβριος 2014)
- [10] Virtuoso Universal Server, URI : <http://www.w3.org/wiki/VirtuosoUniversalServer> (Οκτώβριος 2014)
- [11] Virtuoso Universal Server White Papers, URI: <http://virtuoso.openlinksw.com/white-papers/> (Οκτώβριος 2014)
- [12] Virtuoso Open-Source Edition, URI:

- <http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/> (Οκτώβριος 2014)
- [13] Αποθετήριο Virtuoso Open-Source Edition, URI:
<https://github.com/openlink/virtuoso-opensource> (Οκτώβριος 2014)
- [14] WSO2 Data Services Server, URI: <http://wso2.com/products/data-services-server/> (Οκτώβριος 2014)
- [15] TopBraid Live, URI: <http://www.topquadrant.com/products/topbraid-live/> (Οκτώβριος 2014)
- [16] Authenticating SPARQL client and server, URI:
<http://dig.csail.mit.edu/2008/sparql-server/> (Οκτώβριος 2014)
- [17] SPARQL Endpoints Status - Availability, URI:
<http://sparqls.okfn.org/availability> (Δεκέμβριος 2014)
- [18] Natalya F. Noy and Deborah L. McGuinness, “Ontology Development 101: A Guide to Creating Your First Ontology”, Stanford University
- [19] Gruber T.R. 1993. “A Translation Approach to Portable Ontology Specifications”, Knowledge Acquisition, 5:199–220
- [20] Gruber. Σ, “Toward Principles for the Design of Ontologies Used for Knowledge Sharing”. In: International Journal Human-Computer Studies, 43(5-6):907-928, 1995
- [21] Malviya, Naveen, Nishchol Mishra, and Santosh Sahu. "Developing university ontology using protégé owl tool: Process and reasoning." International Journal of Scientific & Engineering Research 2.9 (2011): 1-8.
- [22] Ling Zeng et al. - Study on Construction of University Course Ontology: Content, Method and Process| IEEE-2009.
- [23] OWL Web Ontology Language, URI: <http://www.w3.org/TR/owl-features> (Νοέμβριος 2014)
- [24] Jeff Heflin —University Ontology —version 1.0, URI:
<http://www.cs.umd.edu/projects/plus/SHOE/onts/univ1.0.html> (Δεκέμβριος 2014)
- [25] Gil, Richard, et al. "Improving ontologies through ontology learning: a university case." Computer Science and Information Engineering, 2009 WRI World Congress on. Vol. 4. IEEE, 2009.
- [26] METHONTOLOGY, URI : <http://semanticweb.org/wiki/METHONTOLOGY> (Ιανουάριος 2015)
- [27] Fensel, Dieter, et al. "On-to-knowledge: Ontology-based tools for knowledge

- management." Proceedings of the eBusiness and eWork. 2000.
- [28] Jones, Dean, Trevor Bench-Capon, and Pepijn Visser. Methodologies for ontology development. na, 1998.
- [29] WebODE, URI : <http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/old-technologies/60-webode> (Ιανουάριος 2015)
- [30] Chimaera, URI: <http://www.ksl.stanford.edu/software/chimaera/> (Ιανουάριος 2015)
- [31] Berners-Lee, T., Fischetti, M. (1999). Weaving the Web, 1st Edition, Harper San Francisco
- [32] Allan M. Collins, A (1969). «Retrieval time from semantic memory». Journal of verbal learning and verbal behavior : 240–247
- [33] Quillian, MR (1967). «Word concepts. A theory and simulation of some basic semantic capabilities». Behavioral Science : 410–430.
- [34] Semantic Web, URI: <http://www.w3.org/standards/semanticweb/> (Ιανουάριος 2014)
- [35] Teresa Wu : Web 3.0. vs Web 2.0., (2013), URI: <http://cdol.lacoe.edu/blog/web-3-0-vs-web-2-0/> (Δεκέμβριος 2014).
- [36] Bagley, Philip R. Extension of programming language concepts. UNIVERSITY CITY SCIENCE CENTER PHILADELPHIA PA, 1968.
- [37] Metadata, URI: http://en.wikipedia.org/wiki/Metadata#Metadata_types (Ιανουάριος 2015)
- [38] Turner, Thomas P., and Lise Brackbill. "Rising to the top: evaluating the use of the HTML meta tag to improve retrieval of World Wide Web documents through Internet search engines." Library resources & technical services 42.4 (1998): 258-271.
- [39] Page, Lawrence, et al. "The PageRank citation ranking: Bringing order to the web." (1999).
- [40] Weibel, Stuart, et al. "Dublin core metadata for resource discovery." Internet Engineering Task Force RFC 2413.222 (1998): 132.
- [41] Dublin Core Metadata, URI : <http://dublincore.org/> (Δεκέμβριος 2014)
- [42] ACR2, URI: <http://www.library.yale.edu/cataloging/toolsaacr2.htm> (Δεκέμβριος 2014)
- [43] GILS, URI: <http://www.gils.net/element2.html>, (Δεκέμβριος 2014)
- [44] A. L. Symeonidis, K. C. Chatzidimitriou, I. N. Athanasiadis and P. A. Mitkas,

"Data mining for agent reasoning: A synergy for training intelligent agents", Engineering Applications of Artificial Intelligence, vol. 20, no. 8, pp. 1097-1111, December 2007.

- [45] P. S. Sajja and R. Akerkar, "Agent-based Web", in Intelligent Technologies for Web Applications, CRC Press, Boca Raton, 2012
- [46] Bizer, Christian, Tom Heath, and Tim Berners-Lee. "Linked data-the story so far." (2009).
- [47] Five ways open data can boost democracy around the world, URI : <http://www.theguardian.com/public-leaders-network/2015/feb/20/open-data-day-fairer-taxes> (Φεβρουάριος 2015)
- [48] Datahub, URI:<http://datahub.io/el/organization/country-gr> (Φεβρουάριος 2015)
- [49] Open Data Index, URI: <http://openindex.gr/> (Φεβρουάριος 2015)
- [50] Ανοικτή Διακυβέρνηση, URI: <http://www.opengov.gr/home/> (Φεβρουάριος 2015)
- [51] Ανοικτά Δεδομένα, URI: <https://diavgeia.gov.gr/blog/wp-content/uploads/2014/11/nomos-4305-2014-anoixta-dedomena-klp.pdf> (Φεβρουάριος 2015)
- [52] Journal European Union, URI: <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2013:175:0001:0008:EN:PDF> (Φεβρουάριος 2015)
- [53] Linked Data web crawler, URI : <https://code.google.com/p/ldspider/> (Φεβρουάριος 2015)
- [54] Creating knowledge out of interlinked data, URI : <http://lod2.eu> (Φεβρουάριος 2015)
- [55] Open Data Institute, URI: <http://opendatainstitute.org/> (Φεβρουάριος 2015)
- [56] ODI ΕΛ/ΛΑΚ, URI: <https://odi.ellak.gr/> (Φεβρουάριος 2015)
- [57] ODI GREECE, URI: <http://theodi.org/nodes/odi-athens> (Φεβρουάριος 2015)
- [58] Open Data Day, URI: <http://opendataday.org/> (Φεβρουάριος 2015)
- [59] Thomas R. Gruber. A translation approach to portable ontology specifications. Knowledge Acquisition, 5(2):199–220, 1993
- [60] SPARQL-RW Framework, URI <http://www.dblab.ntua.gr/~bikakis/SPARQL-RW.html> (Φεβρουάριος 2015)
- [61] Uschold, Mike, and Michael Gruninger. "Ontologies: Principles, methods and applications." The knowledge engineering review 11.02 (1996): 93-136.

- [62] O. Corcho, M. Fernández-López, and A. Gómez-Pérez, ‘Methodologies, tools and languages for building ontologies. Where is their meeting point?’, *Data & Knowledge Engineering*, vol. 46, no. 1, pp. 41 – 64, 2003.
- [63] Jeff Heflin, James Hendler, and Sean Luke, ‘SHOE: A Prototype Language for the Semantic Web’. 12-Dec-2000.
- [64] X. Su and L. Hebrekke, ‘A Comparative Study of Ontology Languages and Tools’, in *Advanced Information Systems Engineering*, vol. 2348, A. Pidduck, M. T. Ozsu, J. Mylopoulos, and C. Woo, Eds. Springer Berlin Heidelberg, 2002, pp. 761–765
- [65] W3C, ‘RDF Primer’, URI: <http://www.w3.org/TR/2004/REC-rdf-primer-20040210/> (Φεβρουάριος 2015)
- [66] W3C, "RDF Schema 1.1", URI: <http://www.w3.org/TR/rdf-schema/> (Φεβρουάριος 2015)
- [67] D. Fensel, F. van Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider, “OIL: an ontology infrastructure for the Semantic Web,” *Intelligent Systems*, IEEE, vol. 16, no. 2, pp. 38–45, Mar. 2001.
- [68] URI: http://www.researchgate.net/profile/Michel_Klein2/publication/2412399_On-To-Knowledge_Ontology-based_Tools_for_Knowledge_Management/file/79e4150ba1a54c08c1.pdf (Δεκέμβριος 2014)
- [69] Sirin, Evren, et al. "Pellet: A practical owl-dl reasoner." *Web Semantics: science, services and agents on the World Wide Web 5.2* (2007): 51-53.
- [70] W3C, "SPARQL Query Language for RDF", URI : <http://www.w3.org/TR/rdf-sparql-query/> (Φεβρουάριος 2015)
- [71] Prud’Hommeaux, Eric, and Andy Seaborne. "SPARQL query language for RDF." W3C recommendation 15 (2008).
- [72] Auer, Sören, et al. *Dbpedia: A nucleus for a web of open data*. Springer Berlin Heidelberg, 2007.
- [73] SPARQL and Scalable Inference on Demand, URI : [http://virtuoso.openlinksw.com/whitepapers/SPARQL and Scalable Inference on Demand.html](http://virtuoso.openlinksw.com/whitepapers/SPARQL_and_Scalable_Inference_on_Demand.html) (Φεβρουάριος 2015)
- [74] SPARQL 1.1, URI : <http://www.w3.org/TR/sparql11-overview/> (Φεβρουάριος 2015)
- [75] Bizer, Christian, and Andreas Schultz. "Benchmarking the performance of

- storage systems that expose SPARQL endpoints." World Wide Web Internet And Web Information Systems (2008).
- [76] Proceedings of the ISWC 2014 Posters & Demonstrations Track a track within the 13th International Semantic Web Conference (ISWC 2014), URI: <http://iswc2014.semanticweb.org> (Φεβρουάριος 2015)
 - [77] Bischof, Stefan, et al. "Schema-Agnostic Query Rewriting in SPARQL 1.1." The Semantic Web–ISWC 2014. Springer International Publishing, 2014. 584-600.
 - [78] SPARQL Endpoint, URI : http://semanticweb.org/wiki/SPARQL_endpoint (Δεκέμβριος 2014)
 - [79] Virtuoso Sponger, URI : <http://virtuoso.openlinksw.com/dataspace/dav/wiki/Main/VirtSpongerWhitePaper?skin=PDFSkin#AncWhat> (Φεβρουάριος 2015)
 - [80] Protégé, URI: <http://protege.stanford.edu/> (Φεβρουάριος 2015)
 - [81] Virtuoso Virtual Database, URI: (Φεβρουάριος 2015)
[http://virtuoso.openlinksw.com/whitepapers/virtuoso virtual database vdb.html](http://virtuoso.openlinksw.com/whitepapers/virtuoso_virtual_database_vdb.html)
 - [82] OpenLink Virtuoso Universal Server: Documentation, URI : <http://docs.openlinksw.com/pdf/virtdocs.pdf> (Ιανουάριος 2015)
 - [83] Git, URI: [http://en.wikipedia.org/wiki/Git_\(software\)](http://en.wikipedia.org/wiki/Git_(software)) (Ιανουάριος 2015)
 - [84] Vload, URI: <https://github.com/timrdf/csv2rdf4lod-automation> (Ιανουάριος 2015)
 - [85] Unix Runlevels, URI: <http://en.wikipedia.org/wiki/Runlevel> (Ιανουάριος 2015)
 - [86] Virtuoso logging trace, URI: http://docs.openlinksw.com/virtuoso/fn_trace_on.html (Ιανουάριος 2015)
 - [87] Dbpedia VAD, URI: <https://github.com/dbpedia/dbpedia-vad-i18n> (Φεβρουάριος 2015)
 - [88] Papakonstantinou, V., et al. "Securing Access to Sensitive RDF Data." The Semantic Web: ESWC 2014 Satellite Events. Springer International Publishing, 2014. 455-460.
 - [89] VAL – The Virtuoso Authentication and Authorization Layer, URI: <https://trueg.wordpress.com/projects/> (Μάρτιος 2015)
 - [90] RDFa Core Initial Context, URI : <http://www.w3.org/2011/rdfa-context/rdfa-1.1> (Μάρτιος 2015)
 - [91] Bulk Loading RDF Source Files into one or more Graph IRIs, URI : <http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtBulkRDFLoad>

er (Μάρτιος 2015)

- [92] RDF Performance Tuning, URI :
<http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtRDFPerformanceTuning> (Μάρτιος 2015)
- [93] RDF Insert Methods in Virtuoso, URI :
<http://shadok.enst.fr:8890/doc/html/rdfinsertmethods.html> (Μάρτιος 2015)
- [94] SEO, URI: http://en.wikipedia.org/wiki/Search_engine_optimization (Μάρτιος 2015)
- [95] Zaveri, Amrapali, et al. "Using Linked Data to Evaluate the Impact of Research and Development in Europe: A Structural Equation Model." The Semantic Web–ISWC 2013. Springer Berlin Heidelberg, 2013. 244-259.

Παράρτημα Α

Πηγές αναφοράς εικόνων - URIs

- 1 Κεντριοποιημένη Αρχιτεκτονική, http://espa.lib.ntua.gr/images/3_1_arxitektoniki.jpeg
- 2 Η διαστρωμάτωση του Σημειολογικού Ιστού Tim Berners-Lee, http://de.wikipedia.org/wiki/Datei:SW_layercake_2006.svg
- 3 Linked Open Data (LOD) Community - Αύγουστος 2014, <http://datahub.io/organization/lodcloud2014>
- 4-8 Virtuoso Open-Source Edition 7.1, Στιγμιότυπα SPARQL Query Editor
- 9-18 Virtuoso Open-Source Edition 7.1, Στιγμιότυπα iSPARQL Query Editor
- 19 Αρχιτεκτονική Virtuoso Universal Server / Open-Source Edition, <http://virtuoso.openlinksw.com/images/varch625.jpg>
- 20 Αρχιτεκτονική Openlink Virtuoso Product, <http://virtuoso.openlinksw.com:8889/doc/images/varch32.jpg>
- 21-49 Virtuoso Open-Source Edition 7.1, Στιγμιότυπα Conductor
- 50-67 Protégé 5.0.0, Δημιουργία οντολογίας