

Ατομική Διπλωματική Εργασία

Υλοποίηση της MashQL στο πλαίσιο του πλοηγού Firefox

Αντρέας Μανώλη

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Δεκέμβριος 2009

ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Υλοποίηση της MashQL στο πλαίσιο του πλοηγού Firefox

Αντρέας Μανώλη

Επιβλέπων Καθηγητής

Μάριος Δ. Δικαιάκος

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Δεκέμβριος 2009

ΕΥΧΑΡΙΣΤΙΕΣ

Θέλω να εκφράσω τις ευχαριστίες μου στον καθηγητή του Πανεπιστημίου Κύπρου και επιβλέπων καθηγητή Δρ. Μάριο Μ. Δικαιάκο, και στον Δρ Mustafa Jarrar για την καθοδήγηση που μου έδωσαν, για την υπομονή που επέδειξαν σε όλη την διάρκεια της συνεργασίας μας αλλά κυρίως για τον τρόπο αντιμετώπισης, σεβασμό και εμπιστοσύνη που επέδειξαν απέναντι στο άτομο μου.

Επίσης θέλω να ευχαριστήσω ιδιαίτερα τους φίλους μου για την έμπρακτη βοήθειά τους και τη ψυχολογική συμπαράστασή τους.

Τέλος, θέλω να ευχαριστήσω τους γονείς μου που με την υποστήριξη τους με βοήθησαν σε όλες τις προσπάθειες μου κατά την διάρκεια της φοίτησης μου στο Πανεπιστήμιο Κύπρου.

Πρόλογος.

Σήμερα το διαδίκτυο αποτελεί τη μεγαλύτερη πηγή πληροφοριών. Μέσω του διαδικτύου ο κάθε άνθρωπος έχει τη δυνατότητα να εντοπίσει πληροφορίες για τα πάντα. Το πρόβλημα που υφίσταται για την αναζήτηση δεδομένων στο διαδίκτυο είναι ότι τα περισσότερα δεδομένα βρίσκονται σε μη δομημένη μορφή. Το semantic web έχει υλοποιήσει αρκετά μοντέλα που έχουν τη δυνατότητα να χρησιμοποιηθούν στο διαδίκτυο για την αναπαράσταση με δομημένο τρόπο τα δεδομένα. Ένα από τα δημοφιλέστερα μοντέλα είναι το RDF.

Στα κεφάλαια που ακολουθούν παρουσιάζετε μια εφαρμογή της MashQL το MashQL Firefox add-on. Η MashQL είναι μια νέα τεχνολογία βασισμένη στα Mashups και έχει ως σκοπό της να βοηθά τους χρήστες του διαδικτύου να αναζητούν πληροφορίες στα RDF αρχεία που είναι αναρτημένα στο διαδίκτυο . Μέσα από τα κεφάλαια που ακολουθούν παρουσιάζεται η δομή της MashQL και του MashQL Firefox add-on. Επιπλέον επεξηγούνται τα RDF αρχεία, η Sparql και η μέθοδος δημιουργίας Firefox add-on. Στο τέλος επισυνάπτονται τα αποτελέσματα από το usability test και επεξηγούνται.

Περιεχόμενα

Κεφάλαιο 1: Εισαγωγή.	1
Κεφάλαιο 2: Τεχνολογίες Διαδικτύου.	3
2.1 Semantic Web.	3
2.2 RDF.	4
2.2.1 Εισαγωγή στα Resource Description Framework (RDF)	4
2.2.2 Προέλευση του RDF	4
2.2.3 Δήλωση βασικών εννοιών.	5
2.2.4 Η βασική δομή του RDF.	5
2.2.5 Το μοντέλο RDF	7
2.2.6 Η XML σύνταξη για το RDF: RDF/XML	9
2.3 Sparql.	10
2.3.1 Εισαγωγή στη Sparql	10
2.3.2 Δημιουργία και επεξήγηση ενός απλού Query στη Sparql	10
2.3.3 Δημιουργία και επεξήγηση ενός Query στη Sparql χρησιμοποιώντας το τελεστή AND και OR.	12
2.3.4 Δημιουργία και επεξήγηση ενός Query στη Sparql χρησιμοποιώντας το OPTIONAL.	14
2.3.5 Δημιουργία και επεξήγηση ενός Query στη Sparql χρησιμοποιώντας το UNION.	15
2.3.6 Δημιουργία και επεξήγηση Query στη Sparql χρησιμοποιώντας FILTER.	16
2.4 Mashup.	17
2.4.1 Τεχνολογία Mashup.	17
2.4.2 Τεχνολογίες που χρησιμοποιούν τα Mashups	18
Κεφάλαιο 3. MashQL	20
3.1 Εισαγωγή στη τεχνολογία MashQL	20
3.2. Ιστορικό και Κίνητρα [5]	20
3.3. Επεξήγηση της δημιουργίας ενός επερώτηματος στη MashQL.	22
3.4. Δυνατότητες της MashQL.	25
3.4.1 Μεταβλητή στη MashQL:	26
3.4.2 Ετικέτα στη MashQL:	26
3.4.3 Φίλτρα στη MashQL:	27

3.4.5 Ο τελεστής AND στη MashQL:.....	27
3.4.6 UNION στη MashQL:.....	29
3.4.7 Maybe στη MashQL:.....	30
3.4.8 WITHOUT στη MashQL:.....	30
Κεφάλαιο 4. MashQL Firefox add-on.....	31
4.1. Εισαγωγή στο MashQL Firefox add-on.	31
4.2 Εισαγωγή στο Firefox.....	31
4.2.1 Mozilla Firefox Web browser.	31
4.2.2 Ιστορία του Mozilla Firefox Web browser.....	32
4.2.3 Mozilla Firefox add-on.....	33
4.3. Η λειτουργικότητα και οι κύριοι αλγόριθμοι που χρησιμοποιεί το MashQL Firefox add-on.	34
4.3.1. Εισαγωγή URI και εξόρυξη των αρχικών Subject.	35
4.3.2. Επιλογή Subject και εξαγωγή των ανάλογων predicate.	38
4.3.1. Επιλογή predicate.....	41
4.3.1. Επιλογή Object και δημιουργία φίλτρων.....	47
Κεφάλαιο 5. MashQL usability test.....	53
5.1. Εισαγωγή.	53
5.2. Περίληψη.....	54
5.3. Μεθοδολογία	54
5.4. Αποτελέσματα	56
Κεφάλαιο 6: Συμπεράσματα.....	67
6.1. Γενικά συμπεράσματα.	67
6.2. Μελλοντική Εργασία.....	68
Βιβλιογραφία.....	69
Παράρτημα Α.....	A-1
Σενάριο και ερωτηματολόγιο.	A-1
RDF αρχείο για πωλητές αυτοκινήτων.	A-10
RDF αρχείο για μηχανικούς αυτοκινήτων.	A-20

Παράρτημα Β.....	B-1
-------------------------	------------

Σχεδιάγραμμα Λειτουργιών	B-1
--------------------------------	-----

Κεφάλαιο 1: Εισαγωγή.

Τα mashups αναδεικνύουν ένα τρόπο με τον οποίο μπορεί να δημιουργηθεί μια νέα εφαρμογή Ιστού με συνδυασμό υφιστάμενων πόρων από το Web, όπως δεδομένα και Web APIs. Ο σκοπός των mashups δηλαδή, είναι η διανομή και η συνάθροιση πληροφοριών για την υποστήριξη δημοσίευσης περιεχομένου, με σκοπό την ανάπτυξη μιας νέας γενιάς εφαρμογών Ιστού. Τα mashups συνδυάζουν υπάρχουσες τεχνολογίες με νέα δεδομένα και περιεχόμενα καθώς και data management και semantic technologies, όπως RSS, RDFa, Sparql και RDF query Language.

Το Resource Description Framework (RDF) είναι μια γλώσσα αναπαράστασης πληροφοριών στο World Wide Web. Προορίζεται ιδιαίτερα για την αντιπροσώπευση των δεδομένων που εμφανίζονται στο Web, όπως ο τίτλος, ο συντάκτης, η ημερομηνία τροποποίησης τις ιστοσελίδας, τα πνευματικά δικαιώματα και οι σχετικές πληροφορίες για τη χορήγησης άδειας χρήσης . Εντούτοις, με τη γενίκευση της έννοιας ενός «Web recourses», το RDF μπορεί να χρησιμοποιηθεί για να αντιπροσωπεύσει πληροφορίες και δεδομένα που είναι προσδιορισμένα στο Web, ακόμα και όταν δεν μπορούν να είναι άμεσα ανακτημένα.

Αν και το RDF έχει τυποποιηθεί από το W3C για την αναπαράσταση μεταδεδομένων, αρκετές κορυφαίες επιχειρήσεις προσπαθούν να εκμεταλλευτούν τη δομή του για διάφορες άλλες εφαρμογές. Η υποστήριξη από τις κορυφαίες επιχειρήσεις είναι πράγματι ένα μεγάλο βήμα για την υιοθέτηση των RDF ως κύρια γλώσσα μεταδεδομένων. Επομένως, τα mashups που θα επεξεργάζονται RDF με τη βοήθεια της SPARQL θα έχουν σημαντικό ρόλο στο κοντινό μέλλον. Το πρόβλημα είναι ότι οι χρήστες θα πρέπει να είναι καλοί γνώστες των πιο πάνω τεχνολογιών για να δημιουργήσουν τα δικά τους mashups.

Για τη διευκόλυνση των χρηστών που δεν διαθέτουν ειδικευμένες γνώσεις για RDF και Sparql προτείνεται από Dr. Mustafa Jarrar και το Dr. Μάριο Δικαϊάκο στο άρθρο MashQL: A Query-by-Diagram Language -Towards Semantic Data Mashups μια νέα μορφή της τεχνολογίας των mashups. Μέσα από αυτό το άρθρο οι συγγραφείς

επιθυμούν να γενικεύσουν την ιδέα των mashups, θεωρώντας το Διαδίκτυο ως βάση δεδομένων και ένα mashup σαν μια επερώτηση προς τη βάση. Επιπλέον υποθέτουν ότι τα δεδομένα που βρίσκονται στο διαδίκτυο είναι της μορφής RDF. Για τη διατύπωση επερωτήσεων σε RDF αρχεία χρησιμοποιούμε τη γλώσσα εκτέλεσης επερωτήσεων SPARQL.

Η MashQL έχει ως απώτερο στόχο να επιτρέψει στους χρήστες να δημιουργούν εύκολα και χωρίς ιδιαίτερες γνώσεις, επερωτήσεις επάνω στα υπάρχοντα δεδομένα στο διαδίκτυο. Στο υπόβαθρο, οι επερωτήσεις MashQL μεταφράζονται αυτόματα επερωτήσεις SPARQL, δίνοντας τη δυνατότητα στους χρήστες να μπορούν να χτίσουν στοιχεία mashups χωρίς να πρέπει να είναι γνωστό σ' αυτούς η δομή ή τεχνικές λεπτομέρειες των δεδομένων.

Κεφάλαιο 2: Τεχνολογίες Διαδικτύου

2.1 Semantic Web.	3
2.2 RDF.	4
2.3 Sparql.	10
2.4 Mashup.	17

2.1 Semantic Web.

Το World wide web υλοποιήθηκε για να παρέχει μη δομημένες πληροφορίες στους ανθρώπους. Αυτές οι πληροφορίες είναι κατανοητές από τον άνθρωπο αλλά όχι από τον υπολογιστή. Με λίγα λόγια είναι αδύνατο να κατανοήσει και να επεξεργαστεί ορθά όλες τις αδόμητες πληροφορίες ένα συγκεκριμένο λογισμικό. Η λύση στο πιο πάνω πρόβλημα είναι το Semantic Web.

Μιλώντας για την τεχνολογία Semantic Web αναφερόμαστε σε δεδομένα δομημένης μορφής τα οποία βρίσκονται στο δίκτυο, και δηλώνουν τα web resources δια μέσου των RDF αρχείων . Η έννοια Semantic Web ήταν ιδέα του Tim Berners-Lee, εφευρέτη του WWW, URIs, HTTP και του HTML. Την ιδέα αυτή ασπάστηκε η κοινοπραξία World Wide Web consortium με απώτερο σκοπό να βελτιώσουν, να επεκτείνουν και να τυποποιήσουν το υπάρχον σύστημα. Εντούτοις, το Semantic Web είναι ακόμα υπό ανάπτυξη και αναμένεται ότι θα επεκταθεί παρά πολύ[4].

2.2 RDF.

2.2.1 Εισαγωγή στα Resource Description Framework (RDF)

Το Resource Description Framework (RDF) είναι γλώσσα αναπαράστασης πληροφοριών στο World Wide Web. Προορίζεται ιδιαίτερα για την αντιπροσώπευση των δεδομένων που εμφανίζονται στο Web, όπως ο τίτλος, ο συντάκτης, η ημερομηνία τροποποίησης τις ιστοσελίδας, τα πνευματικά δικαιώματα και οι σχετικές πληροφορίες για τη χορήγησης άδειας χρήσης . Εντούτοις, με τη γενίκευση της έννοιας ενός «Web resource», το RDF μπορεί να χρησιμοποιηθεί για να αντιπροσωπεύσει πληροφορίες και δεδομένα που είναι προσδιορισμένα στον Web, ακόμα και όταν δεν μπορούν να είναι άμεσα ανακτημένα[2].

Με την εισαγωγή του RDF ένα νέο μεγάλο κεφαλαίο άνοιξε για το World Wide Web, οι πληροφορίες που είναι δομημένες σε ένα RDF αρχείο είναι εκμεταλλεύσιμες. Η τεχνολογία RDF παρέχει ένα κοινό πλαίσιο για την αναπαράσταση των πληροφοριών, καθιστώντας καθολική τη δομή των πληροφοριών. Η καθολικότητα τις δομής των RDF επιτρέπει στις εφαρμογές να επεξεργάζονται όλες τις πληροφορίες που αναπαριστούνται από το συγκεκριμένο μοντέλο.

Η τεχνολογία RDF προσδιορίζει τα δεδομένα σαν Web identifiers (*URIs*). Αυτό επιτρέπει την αντιπροσώπευση των δεδομένων σαν γράφο που παρέχει πληροφορίες για τους πόρους, τις ιδιότητες και τις τιμές των δεδομένων.

2.2.2 Προέλευση του RDF

Υπήρξαν διάφοροι πρόγονοι του W3C RDF. Τεχνικά ο πιο στενός ήταν ο MCF, που άρχισε από το Ramanathan V. Guha και συνεχίστηκε από τη εταιρία Apple Computer με τη συνεισφορά του Tim Bray . Οι ιδέες που διαμορφώθηκαν έπαιξαν βασικό ρολό στη διαμόρφωση της κατεύθυνσης του RDF. Η W3C δημοσίευσε το RDF το 1999. Σε μετέπειτα στάδιο και μετά από ανασκόπηση της αρχικής έκδοσης, το 2004 αναδημοσιεύτηκε το οριστικό μοντέλο[2] .

2.2.3 Δήλωση βασικών εννοιών

Literal: σε ένα RDF γράφημα περιέχονται ένα ή δύο ονομασμένα συστατικά (λέξεις). Όλα τα literals έχουν μια λεξιλογική μορφή, τη Unicode string. Τα plain literals έχουν μια λεξιλογική μορφή και προαιρετικά μια ετικέτα(language tag). Τα typed literals έχουν και αυτά μια λεξιλογική μορφή και ένα datatype URI, που είναι ένα datatype URI reference. Δύο literals είναι ίσα εάν και μόνο ευσταθούν τα παρακάτω: α) τα strings των δύο λεξιλογικών μορφών συγκρίνονται ίσα, χαρακτήρα με χαρακτήρα, β) είτε και τα δύο ή κανένα έχουν language tag, γ) τα language tags, εάν υπάρχουν, συγκρίνονται ίσα, δ) είτε και τα δύο ή κανένα, έχουν datatype URIs, ε) τα δύο datatype URIs, εάν υπάρχουν, συγκρίνονται ίσα[3].

Blank nodes: είναι RDF graphs που προέρχονται από ένα απροσδιόριστο σύνολο(set), το οποίο είναι αυθαίρετο. Το RDF δεν κάνει καμία αναφορά σε οποιαδήποτε εσωτερική δομή των blank nodes[3].

2.2.4 Η βασική δομή του RDF

Όπως προαναφέραμε στη τεχνολογία RDF οι πληροφορίες δηλώνονται βάση ενός πλαισίου έτσι ώστε να είναι επεξεργάσιμες και κατανοητές από λογισμικά που είναι σχεδιασμένα βάση του RDF πρωτόκολλου. Πιο κάτω ακολουθεί μια προσέγγιση του πλαισίου και τις μεθοδολογίας που χρησιμοποιεί η τεχνολογία RDF.

Στο παράδειγμα 2.1 παρουσιάζεται η δήλωση μιας πρότασης στη καθομιλουμένη που έχει ως στόχο να μας ενημερώσει ότι το άτομο `cs05am1` είναι ο δημιουργός της ιστοσελίδας <http://www.cs.ucy.ac.cy/~cs05am1>.

`http://www.cs.ucy.ac.cy/~cs05am1` has a **creator whose value is **cs05am1****

Παράδειγμα 2.1

`http://www.cs.ucy.ac.cy/~cs05am1` has a **creation-year whose value is **2005****

Παράδειγμα 2.2

http://www.cs.ucy.ac.cy/~cs05am1 has a language whose value is English

Παράδειγμα 2.3

Με λίγα λόγια η βασική δομή του RDF αποτελείται από τρία πεδία, το subject , το predicate και το object. Το subject ονομάζεται το μέρος το οποίο αναγνωρίζει το αντικείμενο στο οποίο αναφέρεται η δήλωση. Το μέρος που αναγνωρίζει την ιδιοκτησία ή το χαρακτηριστικό του subject που η δήλωση συγκεκριμενοποιεί ονομάζεται predicate. Το μέρος που αναγνωρίζει την αξία αυτής της ιδιοκτησίας ονομάζεται object. Για το Παράδειγμα 2.1 μπορούμε να διαπιστώσουμε ότι:

- το subject = http://www.cs.ucy.ac.cy/~cs05am1
- το predicate = creator
- το object = cs05am1

Για το Παράδειγμα 2.2 μπορούμε να διαπιστώσουμε ότι:

- το subject = http://www.cs.ucy.ac.cy/~cs05am1
- το predicate = creation-year
- το object = 2005

Για το Παράδειγμα 2.3 μπορούμε να διαπιστώσουμε ότι:

- το subject = http://www.cs.ucy.ac.cy/~cs05am1
- το predicate = language
- το object = English

Εντούτοις, η καθομιλουμένη αν και είναι κατανοητή στους ανθρώπους δεν είναι κατανοητή από τον υπολογιστή. Κάτω από τις πιο πάνω προϋποθέσεις οι δημιουργοί του RDF αποφάσισαν ότι το RDF για να είναι επεξεργάσιμο πρέπει να ακολουθεί τους κανόνες της τεχνολογίας XML. Το RDF καθορίζεται από ένα συγκεκριμένο μοντέλο XML, που ονομάζεται ως *RDF/XML* όπως παρουσιάζεται στο Παράδειγμα 2.4.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
```

```

<rdf:Description rdf:about="http://www.cs.ucy.ac.cy/~cs05am1">
  <foaf:creator>cs05am1</foaf:creator>
  <foaf:language>English</foaf:language>
  <foaf:creation-year>2005</foaf:creation-year>
</rdf:Description>
</rdf:RDF>

```

Παράδειγμα 2.4

2.2.5 Το μοντέλο RDF

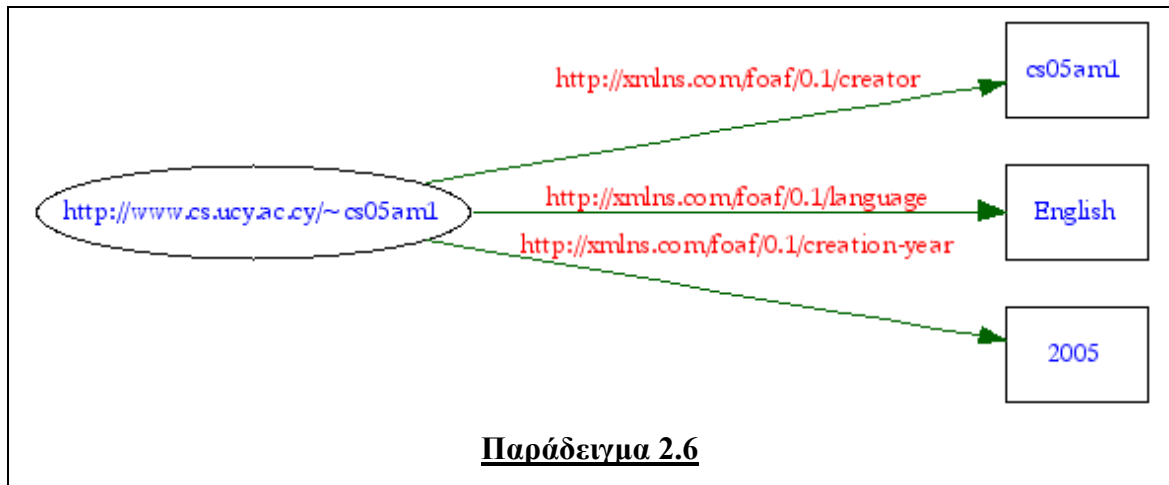
Πιο πάνω διατυπώθηκε η βασική δομή του RDF που αποτελείται από τα τρία πεδία το subject, το predicate και το object. Επίσης αναφέρθηκε γιατί το RDF είναι δομημένο σε RDF/XML μορφή. Σε αυτό το υποκεφάλαιο θα επεξηγηθεί πως στο RDF χρησιμοποιούνται τα URIs για την δήλωση των πόρων. Στο παράδειγμα 2.5 που ακολουθεί εμφανίζονται τα τρία πεδία που προκύπτουν από το παράδειγμα 2.4.

Subject	Predicate	Object
http://www.cs.ucy.ac.cy/~cs05am1	http://xmlns.com/foaf/0.1/creator	"cs05am1"
http://www.cs.ucy.ac.cy/~cs05am1	http://xmlns.com/foaf/0.1/language	"English"
http://www.cs.ucy.ac.cy/~cs05am1	http://xmlns.com/foaf/0.1/creation-year	"2005"

Παράδειγμα 2.5

Επιπλέον τα URI δεν χρησιμοποιούνται μόνο για να προσδιορίσουν το subject της αρχικής δήλωσης, αλλά και για το προσδιορισμό του predicate και του object. Όπως μπορούμε να διακρίνουμε και στο παράδειγμα 2.5 τα πεδία που περιέχουν τα predicate από το RDF/XML αρχείο του παραδείγματος 2.4 παρουσιάζονται σαν URI. Με την χρήση των URI δημιουργείται ο γράφος που βασικά συμπεριλαμβάνει τα δεδομένα ενός RDF αρχείου και τις σχέσεις των δεδομένων. Ο γράφος που δημιουργείται απεικονίζει

με κόμβους το subject και το object και με ένα τόξο, που κατευθύνεται από το subject στο object, το predicate. Στο παράδειγμα 2.6 παρουσιάζεται μια απλή δομή του γράφου για το RDF/XML αρχείο του παραδείγματος 2.4.



Στο παράδειγμα 2.6 διευκρινίζεται ότι τα object στις δηλώσεις RDF μπορούν να είναι είτε URI, είτε σταθερές τιμές (literals), προκειμένου να αντιπροσωπεύσουν ορισμένα είδη τιμών ιδιοκτησίας. Επιπλέον τα literals δεν μπορούν να χρησιμοποιηθούν ως subject ή predicate στις RDF δηλώσεις. Στις RDF γράφους, οι κόμβοι που αποτελούνται από URI παρουσιάζονται ως κυκλικά σχήματα, ενώ οι κόμβοι που αποτελούνται από literals παρουσιάζονται ως ορθογώνια.

```
<http://xmlns.com/foaf/0.1/creator>
```

Παράδειγμα 2.7

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:foaf="http://xmlns.com/foaf/0.1/">
```

Παράδειγμα 2.8

```
<foaf:creator>
```

Παράδειγμα 2.9

Τα URI επιβάλλεται να είναι διατυπωμένα με την μέθοδο που παρουσιάζεται στο παράδειγμα 2.7. Επιπλέον μπορούμε να μειώσουμε το μέγεθος ενός URI με τον ορισμό του στην αρχή του αρχείου ως Prefix όπως παρουσιάζεται στο παράδειγμα 2.8 και να διατυπωθεί όπως παρουσιάζεται στο παράδειγμα 2.9.

2.2.6 Η XML σύνταξη για το RDF: RDF/XML

Στο υποκεφάλαιο που ακολουθεί επεξηγείται η RDF/XML με βάση τη δομή του RDF αρχείου για το παράδειγμα 2.10.

```
1.<?xml version="1.0"?>
2.<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3.  xmlns:foaf="http://xmlns.com/foaf/0.1/">
4. <rdf:Description rdf:about="http://www.cs.ucy.ac.cy/~cs05am1">
5.     <foaf:creation-year>2005</foaf:creation-year>
6. </rdf:Description>
7. </rdf:RDF>
```

Παράδειγμα 2.10

Στη γραμμή με αριθμό 1, η δήλωση `<?xml version="1.0"?>` δηλώνει ότι η μορφή του περιεχομένου του αρχείου είναι XML και επιπλέον δηλώνει και τον αριθμό της έκδοσης της XML που χρησιμοποιείται.

Έπειτα η γραμμή 2 αρχίζει με το πρόθεμα `rdf: RDF` που απεικονίζει ότι το ακόλουθο περιεχόμενο XML αντιπροσωπεύει RDF. Στην ίδια γραμμή ακολουθεί η δήλωση ενός XML namespace με την ονομασία `rdf: RDF`, επιπλέον στην ίδια γραμμή και στην επομένη εμφανίζονται οι `xmlns` δηλώσεις με την ονομασία `rdf` και `foaf` που αντιπροσωπεύουν τα URI. Στις γραμμές από 1 μέχρι 3 δηλώνονται τα απαραίτητα στοιχεία που χρειάζεται για να είναι κατανοητό ότι το αρχείο είναι RDF/XML μορφής,

και προσδιορίζονται τα namespaces που χρησιμοποιούνται μέσα στο περιεχόμενο RDF/XML. Με το τέλος των δηλώσεων για τα xmlns τοποθετούμε το χαρακτήρα '>' για να δείξουμε το τέλος της ετικέτας ,και στο τέλος του αρχείου τοποθετούμε την ετικέτα </rdf:RDF> που συμβολίζει το τέλος του RDF αρχείου. Τέλος οι γραμμές 4 έως 6 είναι δομημένες βάση της σύνταξης subject ,predicate και object που ορίζεται από τη τεχνολογία RDF και προσδιορίζει τα δεδομένα[3].

2.3 Sparql.

2.3.1 Εισαγωγή στη Sparql

Η SPARQL query language, που είναι σχεδιασμένη με βάση τη δομή του RDF, μπορεί να επεξεργάζεται τα query και τις απαιτήσεις που αναγνωρίζονται από το RDF Data Access Working Group[1].

2.3.2 Δημιουργία και επεξήγηση ενός απλού Query στη Sparql

Στο υποκεφάλαιο που ακολουθεί θα επεξηγηθεί μέσω παραδειγμάτων η δομή απλών Query σε Sparql. Τα Query που θα επεξηγηθούν θα εξάγουν δεδομένα από το RDF αρχείο που παρουσιάζεται στο παράδειγμα 2.11. Το συγκεκριμένο αρχείο περιέχει δεδομένα για τον κατασκευαστή της ιστοσελίδας "http://www.newPage.com" και για τον κατασκευαστή της ιστοσελίδας "<http://www.newPage.com>". Για τον κατασκευαστή της ιστοσελίδας "http://www.newPage.com" το αρχείο περιέχει συγκεκριμένα το πλήρες όνομα του κατασκευαστή (Bill Smith), τη προσωπική του ιστοσελίδα με ονομασία "http://www.BillSmith.com" και τη επαγγελματική του ιστοσελίδα με ονομασία "http://www.creatorHomePage.com". Επιπλέον το RDF αρχείο περιέχει το όνομα του κατασκευαστή και το επίθετο σαν δύο ξεχωριστές εγγραφές. Για το κατασκευαστή της ιστοσελίδας "http://www.newPage2.com" το

αρχείο περιέχει συγκεκριμένα το πλήρες όνομα του κατασκευαστή (John Smith), τη προσωπική του ιστοσελίδα με ονομασία "http://www. JohnSmith.com" και τη επαγγελματική του ιστοσελίδα με ονομασία "http://www.creatorHomePage2.com" αλλά το αρχείο δεν περιέχει το όνομα του κατασκευαστή και το επίθετο σαν δύο ξεχωριστές εγγραφές.

creator.rdf

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF xmlns:foaf="http://xmlns.com/foaf/0.1/"
          xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <foaf:Creator rdf:about="http://www.newPage.com">
    <foaf:name>Bill Smith</foaf:name>
    <foaf:firstName> Bill</foaf:firstName>
    <foaf:surname>Smith</foaf:surname>
    <foaf:homepage rdf:resource="http://www.creatorHomePage.com"/>
    <foaf:personalHomepage rdf:resource="http://www.BillSmith.com"/>
  </foaf:Creator>

  <foaf:Creator rdf:about="http://www.newPage2.com">
    <foaf:name>John Smith</foaf:name>
    <foaf:homepage rdf:resource="http://www.creatorHomePage2.com"/>
    <foaf:personalHomepage rdf:resource="http://www.JohnSmith.com"/>
  </foaf:Creator>

</rdf:RDF>
```

Παράδειγμα 2.11

Όπως προαναφέρθηκε στο παράδειγμα 2.11 παρουσιάζονται οι πληροφορίες για τον κατασκευαστή μιας ιστοσελίδας. Επιπλέον οι πληροφορίες είναι δομημένες σε ένα

RDF αρχείο βάση της RDF/XML μορφής. Πιο κάτω παρουσιάζεται το Query 2.1 που επιστρέφει το όνομα του κατασκευαστή.

```
SELECT ?name
FROM < creator.rdf>
WHERE {
    ?s <http://xmlns.com/foaf/0.1/name> ?name .
}
```

Query 2.1

Το συγκεκριμένο Query αποτελείται από τρία πεδία, το SELECT, το FROM και το WHERE, στο πεδίο SELECT καθορίζονται οι μεταβλητές που επιστρέφουν τα επιθυμητά αποτελέσματα και στο πεδίο FROM καθορίζετε το path του RDF αρχείου που περιέχει τα δεδομένα. Τέλος στο πεδίο WHERE ορίζονται τα τρία πεδία subject, object και predicate. Συγκεκριμένα το subject αντιπροσωπεύεται από την μεταβλητή ?s, το object από το URI <http://xmlns.com/foaf/0.1/name> και το predicate από την μεταβλητή ?name . Το αποτέλεσμα αυτού του ερωτήματος θα είναι Bill Smith και John Smith.

2.3.3 Δημιουργία και επεξήγηση ενός Query στη Sparql χρησιμοποιώντας το τελεστή AND και OR.

Επίσης στη Sparql μπορούμε να υλοποιήσουμε και πιο σύνθετα Query με το τρόπο που συνθέτουμε τις τριπλέτες εντός του επερωτήματος. Στο Query 2.2 που ακολουθεί παρουσιάζεται η μέθοδος συγγραφής ενός επερωτήματος που θα εκτελεί το λογικό τελεστή AND. Αυτό επιταχύνεται με το να δώσουμε την ίδια ονομασία στις μεταβλητές που αναπαριστούν το subject στις δυο τριπλέτες. Επιπλέον με την ίδια λογική μπορούμε να υλοποιήσουμε ένα επερώτημα που θα χρησιμοποιούμε το τελεστή OR. Η διαφορά των δυο επερωτημάτων είναι ότι θα πρέπει να δώσουμε διαφορετική ονομασία στις μεταβλητές που αναπαριστούν το subject στις δυο τριπλέτες. Στο Query 2.3 παρουσιάζεται παράδειγμα επερωτήματος που υλοποιεί το λογικό τελεστή OR.

```

SELECT DISTINCT    ?firstname    ?surname
FROM < creator.rdf>
WHERE
{
?s    <http://xmlns.com/foaf/0.1/name> ?firstname .
?s <http://xmlns.com/foaf/0.1/surname> ?surname .    }

```

Query 2.2

Το συγκεκριμένο Query αποτελείται από τρία πεδία, το SELECT, το FROM και το WHERE, στο πεδίο SELECT καθορίζονται οι μεταβλητές που επιστρέφουν τα επιθυμητά αποτελέσματα και στο πεδίο FROM καθορίζεται το path του RDF αρχείου που περιέχει τα δεδομένα. Τέλος το πεδίο WHERE αποτελείται από δυο τριπλέτες, που ορίζονται τα τρία πεδία subject, object και predicate. Συγκεκριμένα το subject αντιπροσωπεύεται από την μεταβλητή ?s και στις δυο τριπλέτες, το object από το URI <http://xmlns.com/foaf/0.1/name> για τη πρώτη τριπλέτα και το URI <http://xmlns.com/foaf/0.1/surname> για τη δεύτερη. Τέλος το predicate για τη πρώτη τριπλέτα αναπαριστάται από τη μεταβλητή ?name και για τη δεύτερη τριπλέτα αναπαριστάται από τη μεταβλητή ?surname. Το αποτέλεσμα αυτού του ερωτήματος θα είναι ?name= Bill Smith και ?surname= Smith.

```

SELECT DISTINCT    ?firstname    ?surname
FROM < creator.rdf>
WHERE
{
?s1    <http://xmlns.com/foaf/0.1/firstName> ?name .
?s2    <http://xmlns.com/foaf/0.1/surname> ?surname .    }

```

Query 2.3

Το συγκεκριμένο Query αποτελείται από τρία πεδία, το SELECT, το FROM και το WHERE, στο πεδίο SELECT καθορίζονται οι μεταβλητές που επιστρέφουν τα επιθυμητά αποτελέσματα και στο πεδίο FROM καθορίζεται το path του RDF αρχείου που περιέχει τα δεδομένα. Τέλος στο πεδίο WHERE αποτελείται από δυο τριπλέτες, ορίζονται τα τρία πεδία subject, object και predicate. Συγκεκριμένα το subject αντιπροσωπεύεται από την μεταβλητή ?s1 για τη πρώτη τριπλέτα και ?s2 για τη δεύτερη τριπλέτα, το object από το URI <http://xmlns.com/foaf/0.1/name> για τη πρώτη τριπλέτα και το URI <http://xmlns.com/foaf/0.1/surname> για τη δεύτερη. Τέλος το predicate για τη πρώτη τριπλέτα αναπαρίσταται από τη μεταβλητή ?name και για τη δεύτερη τριπλέτα αναπαρίσταται από τη μεταβλητή ?surname. Το αποτέλεσμα αυτού του ερωτήματος θα είναι, ?name= Bill Smith και John Smith, και , ?surname= Smith.

2.3.4 Δημιουργία και επεξήγηση ενός Query στη Sparql χρησιμοποιώντας το OPTIONAL.

Η Sparql επιπλέον υποστηρίζει και τον τελεστή OPTIONAL, η έννοια του συγκεκριμένου τελεστή μπορεί να δοθεί πλήρως με τη λέξη 'εάν'. Στο Query 2.4 που ακολουθεί χρησιμοποιείται ο τελεστής OPTIONAL και το επερώτημα επιστρέφει όλα τα πλήρες ονόματα που περιέχει το RDF αρχείο και το επώνυμο εάν η πληροφορία αυτή περιέχεται στο αρχείο.

```
SELECT DISTINCT    ?name    ?surname
FROM <creator.rdf>
WHERE
{?s <http://xmlns.com/foaf/0.1/name> ?name
Optional
{ ?s <http://xmlns.com/foaf/0.1/surname> ?surname . }
}
```

Query 2.4

Το συγκεκριμένο Query αποτελείται από τρία πεδία, το SELECT, το FROM και το WHERE, στο πεδίο SELECT καθορίζονται οι μεταβλητές που επιστρέφουν τα επιθυμητά αποτελέσματα και στο πεδίο FROM καθορίζεται το path του RDF αρχείου που περιέχει τα δεδομένα. Τέλος στο πεδίο WHERE αποτελείται από δυο τριπλέτες, ορίζονται τα τρία πεδία subject, object και predicate. Συγκεκριμένα το subject αντιπροσωπεύεται από την μεταβλητή ?s και στις δυο τριπλέτες, το object από το URI <http://xmlns.com/foaf/0.1/name> για τη πρώτη τριπλέτα και το URI <http://xmlns.com/foaf/0.1/surname> για τη δεύτερη. Τέλος το predicate για τη πρώτη τριπλέτα αναπαριστάται από τη μεταβλητή ?name και για τη δεύτερη τριπλέτα αναπαριστάται από τη μεταβλητή ?surname. Το αποτέλεσμα αυτού του ερωτήματος θα είναι, ?name= Bill Smith και John Smith, και ?surname= Smith.

2.3.5 Δημιουργία και επεξήγηση ενός Query στη Sparql χρησιμοποιώντας το UNION.

Στη γλώσσά εκτέλεσης επερωτημάτων Sparql συναντούμε και την έννοια της ένωσης (UNION). Στο παράδειγμα που ακολουθεί επεξηγείται πως είναι δυνατό να δημιουργηθεί ένα επερώτημα χρησιμοποιώντας την ένωση. Σημαντικό σημείο που πρέπει να προσέξουμε είναι ότι οι τριπλέτες που θα χρησιμοποιηθούν στη σύνθεση του μέρους του επερώτηματος που θα γίνει η ένωση, πρέπει να έχουν κοινή μεταβλητή τόσο στο subject όσο και στο predicate.

```
SELECT DISTINCT    ?firstNameAndName
FROM < creator.rdf>
WHERE {    {    ?s <http://xmlns.com/foaf/0.1/name> ?firstNameAndName    . }
UNION
{    ?s <http://xmlns.com/foaf/0.1/firstName> ?firstNameAndName    . }    }
```

Query 2.5

Το συγκεκριμένο Query αποτελείται από τρία πεδία, το SELECT, το FROM και το WHERE, στο πεδίο SELECT καθορίζονται οι μεταβλητές που επιστρέφουν τα επιθυμητά αποτελέσματα και στο πεδίο FROM καθορίζεται το path του RDF αρχείου που περιέχει τα δεδομένα. Τέλος στο πεδίο WHERE αποτελείται από δυο τριπλέτες, ορίζονται τα τρία πεδία subject, object και predicate. Συγκεκριμένα το subject αντιπροσωπεύεται από την μεταβλητή ?s και στις δυο τριπλέτες, το object από το URI <http://xmlns.com/foaf/0.1/name> για τη πρώτη τριπλέτα και το URI <http://xmlns.com/foaf/0.1/ firstName> για τη δεύτερη. Τέλος το predicate και για τις δυο τριπλέτες αναπαρίσταται από τη μεταβλητή ?firstNameAndName. Το αποτέλεσμα αυτού του ερωτήματος θα είναι, ?firstNameAndName = Bill Smith, John Smith και Bill.

2.3.6 Δημιουργία και επεξήγηση Query στη Sparql χρησιμοποιώντας FILTER.

Το πιο κάτω παράδειγμα παρουσιάζει τη χρήση φίλτρων στη Sparql. Το πιο κάτω ερωτήματα επιστρέφει όλα τα πλήρες ονόματα των δημιουργών με το ενδεχόμενο ότι το μικρό τους όνομα είναι Bill. Η συνάρτησης FILTER δέχεται σαν παράμετρο τη σύγκριση. Στη Sparql επιτρέπεται η χρήση όλων των τελεστών όπως της ισότητας (=). Επιπλέον, υποστηρίζει και τη σύγκριση συμβολοσειρών και αυτό είναι εφικτό με τη βοήθεια της συνάρτησης str().

```
SELECT ?name
FROM < creator.rdf>
WHERE
{
    ?s rdf:type <http://xmlns.com/foaf/0.1/Creator> .
    ?s <http://xmlns.com/foaf/0.1/firstName> ?o1 .
    FILTER (str( ?o1 )="Bill") .
    ?s <http://xmlns.com/foaf/0.1/name> ?name . }

```

Query 2.6

Το συγκεκριμένο Query αποτελείται από τρία πεδία, το SELECT, το FROM και το WHERE, στο πεδίο SELECT καθορίζονται οι μεταβλητές που επιστρέφουν τα επιθυμητά αποτελέσματα και στο πεδίο FROM καθορίζεται το path του RDF αρχείου που περιέχει τα δεδομένα. Τέλος στο πεδίο WHERE αποτελείται από δυο τριπλέτες, ορίζονται τα τρία πεδία subject, object και predicate. Συγκεκριμένα το subject αντιπροσωπεύεται από την μεταβλητή ?s και στις δυο τριπλέτες, το object από το URI <http://xmlns.com/foaf/0.1/firstName> για τη πρώτη τριπλέτα και το URI <http://xmlns.com/foaf/0.1/ name> για τη δεύτερη. Τέλος το predicate για τη πρώτη τριπλέτα αναπαριστάται από τη μεταβλητή ?o1 και για τη δεύτερη τριπλέτα αναπαριστάται από τη μεταβλητή ?name. Το αποτέλεσμα αυτού του ερωτήματος θα είναι, ?name= Bill Smith.

2.4 Mashup.

2.4.1 Τεχνολογία Mashup.

Τα mashups αναδεικνύουν ένα τρόπο με τον οποίο μπορεί να δημιουργηθεί μια νέα εφαρμογή Ιστού με συνδυασμό υφιστάμενων πόρων από το Web, όπως δεδομένα και Web APIs. Ο σκοπός των mashups δηλαδή, είναι η διανομή και η συνάθροιση πληροφοριών για την υποστήριξη δημοσίευσης περιεχομένου με σκοπό την ανάπτυξη μιας νέας γενιάς εφαρμογών Ιστού.

Τα mashups συνδυάζουν υπάρχουσες τεχνολογίες με νέα δεδομένα και περιεχόμενα καθώς και data management και semantic technologies, όπως RSS, RDFa, Sparql και RDF query Language. Φυσικά ο συνδυασμός των πιο πάνω εφαρμογών, υπηρεσιών και περιεχομένου είναι μια πρόκληση για την επιστημονική κοινότητα. Φυσικά έχουν είδη αναπτυχθεί ειδικά εργαλεία για την σύνθεση των εφαρμογών αυτών από τους χρήστες[7].

Παραδείγματα κάποιων εργαλείων είναι τα εξής:

- IBM's QEDWiki (<http://services.alphaworks.ibm.com/qedwiki/>)

- Yahoo Pipes (<http://pipes.yahoo.com>)
- Google Mashup Editor (<http://code.google.com/gme/>) **Εκτός Λειτουργίας**
- Microsoft's Popfly (www.popfly.com) **Εκτός Λειτουργίας**

Υπάρχουν διάφορα είδη mashups όπως consumer, data και enterprise mashups με τον πιο συνηθισμένο τύπο τα consumer mashups οι εφαρμογές των οποίων στοχεύουν το ευρύτερο κοινό. Τα data mashups επίσης είναι αρκετά διαδεδομένα τα οποία συνδυάζουν παρόμοιους τύπους πληροφοριών και μέσων από πολλαπλές πηγές σε μια καινούργια αναπαράσταση[7].

Επίσης σημαντικό είναι να αναφερθεί πως αρχιτεκτονικά υπάρχουν δυο είδη mashups, τα Web based και τα server based. Τα web based χρησιμοποιούν τον φυλλομετρητή του χρήστη για να συνδυάζουν τα δεδομένα, ενώ τα server based αναλύουν τα δεδομένα σε κάποιον απομακρυσμένο server και μεταδίδουν τα τροποποιημένα δεδομένα στον χρήστη στην τελική τους μορφή[7].

2.4.2 Τεχνολογίες που χρησιμοποιούν τα Mashups

Κάποιες από τις τεχνολογίες που χρησιμοποιούν και συνδυάζουν τα mashups για να παράγουν τα τελικά αποτελέσματα είναι οι τεχνολογίες **RSS, RDF και XML**.

Ο όρος RSS προέρχεται από το αγγλικό Really Simple Syndication. Είναι ένα format ανταλλαγής περιεχομένου βασισμένο σε γλώσσα XML. Το RSS (Really Simple Syndication) είναι μια τεχνολογία που επιτρέπει στους χρήστες του Διαδικτύου να λαμβάνουν άμεσα πληροφορίες από πολλές και στοχευμένες πηγές. Οι πηγές RSS προσφέρουν λίστες άρθρων από συγκεκριμένες ιστοσελίδες ή από ενότητες ιστοσελίδων. Όταν επέλθει μια προσθήκη ή ενημέρωση στις λίστες αυτές, οι χρήστες λαμβάνουν άμεση ενημέρωση.

Το Resource Description Framework (RDF) είναι μια οικογένεια των προδιαγραφών του World Wide Web (W3C) που σχεδιάζονται αρχικά ως πρότυπο για

στοιχεία μεταδεδομένων. Χρησιμοποιείται ως γενική μέθοδος για την εννοιολογική περιγραφή ή τη διαμόρφωση των πληροφοριών που εφαρμόζονται στους πόρους Ιστού με την χρησιμοποίηση διαφόρων ειδών σύνταξης.

Η XML είναι ένα σύνολο κανόνων για την δημιουργία και σύνταξη ηλεκτρονικών εγγράφων. Είναι στη ουσία ένα υποσύνολο της SGML. Ο σχεδιασμός της XML έχει σαν στόχο την απλότητα , τη γενικότητα και τη χρησιμοποίηση της στο διαδίκτυο.

Κεφάλαιο 3. MashQL

3.1 Εισαγωγή στη τεχνολογία MashQL	20
3.2. Ιστορικό και Κίνητρα	20
3.3. Επεξήγηση της δημιουργίας ενός επερώτηματος στη MashQL.	22
3.4. Δυνατότητες της MashQL.	25

3.1 Εισαγωγή στη τεχνολογία MashQL

Στο κεφάλαιο που επακολουθεί καθορίζεται η σημασία της MashQL, η οποία είναι μια διαγραμματική γλώσσα εκτέλεσης ερωτημάτων με απώτερο στόχο να επιτρέπει στους ανθρώπους να υλοποιήσουν mashups. Η καινοτομία της MashQL είναι ότι επιτρέπει την αναζήτηση σε μία πηγή δεδομένων ανεξάρτητα εάν είναι γνωστό το σχήμα ή η δομή της. Επίσης, οι χρήστες δεν χρειάζεται να διαθέτουν γνώσεις για RDF και SPARQL.

3.2. Ιστορικό και Κίνητρα [5]

Η ταχεία ανάπτυξη του ιστού έχει δημιουργήσει την ανάγκη να καταστεί το περιεχόμενο που περιέχετε στο διαδίκτυο επαναχρησιμοποιήσιμο. Αρκετές επιχειρήσεις όπως η Google, η Yahoo, η Microsoft, η Amazon, το eBay, η LinkedIn, και η Wikipedia, έχουν κάνει το περιεχόμενό τους δημόσια προσιτό μέσω APIs. Η επαναχρησιμοποίηση των περιεχομένων δίνει την ευκαιρία να χρησιμοποιηθούν δεδομένα από πολλές πηγές για τη δημιουργία μιας εντελώς καινούργιας διαδικτυακής εφαρμογής. Για παράδειγμα, κάποιος μπορεί να δημιουργήσει μια εφαρμογή που θα έχει πρόσβαση στο περιεχόμενο της ιστοσελίδας www.craigslist.org/, η οποία περιέχει

πληροφορίες για αγορά οικίας. Η εφαρμογή αυτή θα συνδυάζει πληροφορίες από το Google Maps για να προσδιορίζει την ακριβή θέση της οικίας με άμεσο αποτέλεσμα τη δημιουργία μιας καινούργιας υπηρεσίας που δεν προσφέρονταν αρχικά από τις δυο προϋπάρχοντες εφαρμογές .

Οι εφαρμογές στο διαδίκτυο που επαναχρησιμοποιούν περιεχόμενα που προέρχονται από υπάρχουσες εφαρμογές ονομάζονται Mashups. Αρκετές εταιρίες υλοποίησαν διάφορες εφαρμογές για τη εύκολη δημιουργία mashups με πιο γνωστά το Google Mashup, το Microsoft's Popfly, το IBM's Smash και το Yahoo Pipes. Το Yahoo Pipes έχουν λάβει τη μέγιστη προσοχή χάρη στην απλότητά του και στη δυνατότητα να επιτρέπει στους χρήστες να συνδυάσουν διαφορετικές πηγές δεδομένων και να υλοποιήσουν τα δικά τους mashups, μέσα από ένα γραφικό και φιλικό προς το χρήστη περιβάλλον.

Τα Yahoo Pipes γενικεύουν την ιδέα του mashup, παρέχοντας τη δυνατότητα στο χρήστη να εκμεταλλευτεί στοιχεία από RSS, Atom και RDF feed. Ο συνδυασμός δεδομένων από διάφορες πηγές και με τη χρήση διάφορων φίλτρων ο χρήστης έχει τη δυνατότητα να εξάγει τα δεδομένα που αναζητά σε μια καινούργια ιστοσελίδα. Το πιο ενδιαφέρον μέρος σε αυτό, είναι ότι ένας χρήστης δεν πρέπει πραγματικά να έχει την τεχνική εμπειρία.

Εντούτοις, όλες οι εκδόσεις που έχουν δημιουργήσει οι εταιρίες για υλοποίηση mashup περιορίζονται στη επεξεργασία δεδομένων που προέρχονται από RSS, Atom και RDF feed. Αυτά τα σχήματα είναι ικανά μόνο για αντιπροσώπευση των news items και όχι των data items τα οποία κωδικοποιούνται σε RDF και XML αρχεία. Για την επεξεργασία των data items δεν έχει υλοποιηθεί καμία εφαρμογή, έτσι δεν έχουν όλοι οι χρήστες του διαδικτύου τη δυνατότητα να χρησιμοποιήσουν τα data items που βρίσκονται στο διαδίκτυο.

Για τη διευκόλυνση των χρηστών που δεν διαθέτουν ειδικευμένες γνώσεις για RDF και Sparql προτείνεται από Dr. Mustafa Jarrar και το Dr. Μάριο Δικαϊάκο στο άρθρο MashQL: A Query-by-Diagram Language -Towards Semantic Data Mashups μια νέα μορφή της τεχνολογίας των mashups. Μέσα από αυτό το άρθρο οι συγγραφείς επιθυμούν να γενικεύσουν την ιδέα των mashups, θεωρώντας το Διαδίκτυο ως βάση

δεδομένων και ένα mashup σαν μια επερώτηση προς τη βάση. Επιπλέον υποθέτουν ότι τα δεδομένα που βρίσκονται στο διαδίκτυο είναι της μορφής RDF. Για τη διατύπωση επερωτήσεων σε RDF αρχεία χρησιμοποιούμε τη γλώσσα εκτέλεσης επερωτήσεων SPARQL.

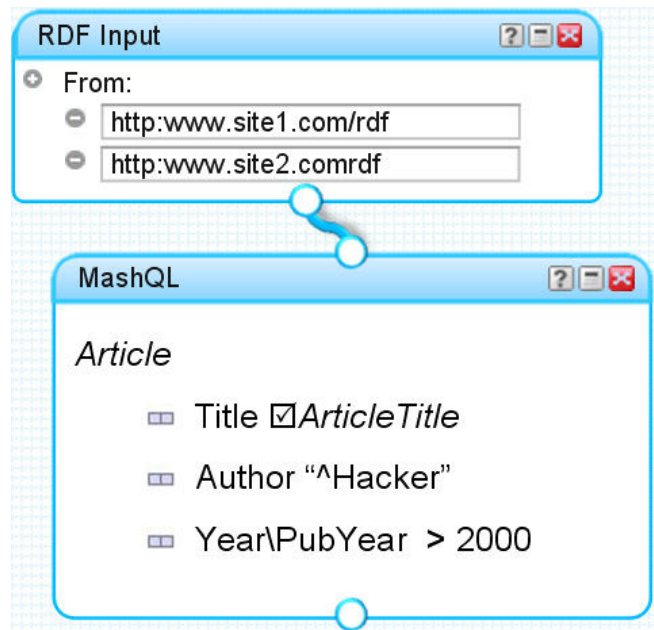
Αν και το RDF έχει τυποποιηθεί από W3C για την αναπαράσταση μεταδεδομένων οι συγγραφείς τονίζουν πως αρκετές κορυφαίες επιχειρήσεις προσπαθούν να εκμεταλλευτούν τη δομή του για διάφορες άλλες εφαρμογές. Παραδείγματος χάριν, το Yahoo ανήγγειλε ότι η επόμενη γενεά μηχανών αναζήτησης θα έχει την ικανότητα να κατανοεί το semantics web μέσω των RDF αρχείων. Το MySpace ανήγγειλε ότι θα υιοθετήσει την τεχνολογία semantics web και ότι θα χρησιμοποιήσει RDF αρχεία στη δομή του. Επιπλέον η Oracle 11g έχει την ικανότητα να επεξεργαστεί RDF αρχεία και να εκτελέσει επερωτήματα με SPARQL.

Η υποστήριξη από τις κορυφαίες επιχειρήσεις είναι πράγματι ένα μεγάλο βήμα για την υιοθέτηση των RDF ως κύρια γλώσσα μεταδομένων. Επομένως, τα mashups που θα επεξεργάζονται RDF με τη βοήθεια της SPARQL θα έχουν σημαντικό ρόλο στο κοντινό μέλλον. Το πρόβλημα είναι ότι οι χρήστες θα πρέπει να είναι καλοί γνώστες των πιο πάνω τεχνολογιών για να δημιουργήσουν τα δικά του mashups. Συνδυάζοντας τα πιο πάνω δεδομένα το άρθρο εισηγείται ένα νέο εργαλείο για τη δημιουργία mashups, τη MashQL.

3.3. Επεξήγηση της δημιουργίας ενός επερωτήματος στη MashQL.

Η MashQL έχει ως απώτερο στόχο να επιτρέψει στους χρήστες να δημιουργούν εύκολα και χωρίς ιδιαίτερες γνώσεις, επερωτήσεις επάνω στα υπάρχοντα δεδομένα στο διαδίκτυο. Στο υπόβαθρο, οι επερωτήσεις MashQL μεταφράζονται αυτόματα επερωτήσεις SPARQL, δίνοντας τη δυνατότητα στους χρήστες να μπορούν να χτίσουν στοιχεία mashups χωρίς να πρέπει να είναι γνωστό σ' αυτούς η δομή ή τεχνικές λεπτομέρειες των δεδομένων.

Στο παράδειγμα 3.1 που ακολουθεί παρουσιάζεται μια απλή υλοποίηση mashup με τη MashQL. Στο πεδίο RDF Input εισάγονται τα resources και στο πεδίο MashQL υλοποιείται το επερώτημα. Το συγκεκριμένο επερώτημα αναζητά τους τίτλους όλων των άρθρων που ο συγγραφέας τους είναι ο Hacker και η χρονολογία δημοσίευσής του ή συγγραφής του είναι μετά το 2000[5].



Παράδειγμα 3.1

Στο παράδειγμα 3.2 παρουσιάζεται ακόμα μια απλή υλοποίηση mashup με τη MashQL. Το επερώτημα υλοποιείται με γραφικό τρόπο και παρουσιάζεται σαν ένα δέντρο. Η ρίζα αυτού του δέντρου αποτελεί το θέμα του επερώτηματος (π.χ. άρθρο), το οποίο είναι το περιεχόμενο που ερευνάται. Κάθε παιδί της ρίζας του δέντρου αποτελεί ένα περιορισμό για το επερώτημα και χρησιμοποιείται για να περιορίσει το μέγεθος των αποτελεσμάτων. Επιπλέον κάθε παιδί της ρίζας μπορεί να έχει και αυτό παιδιά που θα περιορίζουν το μέγεθος του και έτσι να δημιουργούν τα δικά τους υποδέντρα. Το επερώτημα του παραδείγματος 3.2 ανακτά τα πρόσφατα άρθρα από τη Κύπρο, δηλ. κάθε άρθρο που γράφτηκε από οποιοδήποτε συντάκτη, ο οποίος έχει μια διεύθυνση που ανήκει στη Κύπρο, και το άρθρο δημοσιεύεται μετά από το 2000[5].



Παράδειγμα 3.2

Όπως παρατηρούμε από τα δυο πιο πάνω παραδείγματα η MashQL έχει ως σκοπό να απαλλάξει το χρήστη από τη γνώση και τη πολυπλοκότητα της δημιουργίας Sparql . Η MashQL δίνει τη δυνατότητα στους χρηστές να δημιουργούν τα δικά τους επερώτημα επιλέγοντας από τους εξελισσόμενους καταλόγους που παράγονται δυναμικά, ανάλογα με τις προηγούμενες επιλογές του χρήστη και από τα δεδομένα που περιέχουν τα RDF αρχεία. Πιο κάτω θα προσπαθήσω να εξηγήσω την πορεία που ακολουθεί ο χρήστης βήμα προς βήμα για να δημιουργήσει τα δικά του ερωτήματα στη MashQL.

Καταρχάς ο χρήστης πρέπει να τοποθετήσει στο πεδίο Input τα URI που βρίσκονται τα RDF αρχεία που εμπεριέχουν τα δεδομένα που επιθυμεί να επεξεργαστεί. Η MashQL όπως μπορούμε να διακρίνουμε και στο παράδειγμα 3.1 παρέχει τη δυνατότητα στο χρήστη να εισάγει ένα ή περισσότερα RDF αρχεία. Αφότου τοποθετηθούν τα URI που αναπαριστούν τα RDF αρχεία, ξεκινά η διαδικασία διατύπωση μιας επερώτησης στη MashQL. Μετέπειτα ο χρήστης πρέπει να επιλέξει ένα ή περισσότερα θέματα μέσα από τον εξελισσόμενο κατάλογο που δημιουργείται δυναμικά. Ο εξελισσόμενος κατάλογος περιέχει όλα τα θέματα που εμπεριέχονται στα RDF αρχεία και που δόθηκαν στην είσοδο από το χρήστη. Στο παράδειγμα 3.1 και 3.2 το θέμα που επιλέχτηκε από το χρήστη είναι ο συγγραφέας. Όπως προαναφέρθηκε ο χρήστης έχει τη δυνατότητα να επιλέξει περισσότερα από ένα θέματα, αυτή η δυνατότητα του δίνεται με τη χρήση UNION (‘\’) που αναφέρουμε στο επόμενο υποκεφάλαιο. Επιπλέον εάν ο χρήστης θέλει να δημιουργήσει ένα επερώτημα που θα αναφέρεται σε όλα τα θέματα που του προσφέρονται έχει τη δυνατότητα να μην επιλέξει κάποιο θέμα και να δώσει μια μεταβλητή στη θέση του θέματος. Η μεταβλητή θα εμπεριέχει όλες τις δυνατές τιμές που προσφέρονται από τα RDF αρχεία[6].

Συνεχίζοντας έχει τη δυνατότητα να προσθέσει ένα περιορισμό στο επιλεγμένο θέμα, από ένα νέο εξελισσόμενο κατάλογο που θα τοποθετηθεί κάτω από το θέμα που προεπιλέγετε. Ο κατάλογος θα εμπεριέχει όλες τις ιδιότητες του θέματος. Στο παράδειγμα 3.1 και 3.2 ο κατάλογος περιείχε μαζί με άλλες ιδιότητες και τις ιδιότητες συγγραφέας, τίτλος, χρονολογία και χρονολογία δημοσίευσης. Όπως μπορείτε να διαπιστώσετε στα δυο πιο πάνω παραδείγματα ο χρήστης μετά από τη επιλογή του θέματος επέλεξε από το κατάλογο την ιδιότητα τίτλος. Επιπλέον ο χρήστης έχει τη

δυνατότητα να χρησιμοποιήσει και σε αυτό το πεδίο τον τελεστή UNION(‘\’) ή να μην επιλέξει κάποια τιμή και να τοποθετήσει μεταβλητή. Εφόσον ο χρήστης επέλεξε ιδιότητα, του δίνεται η δυνατότητα να επιλέξει το βάρος της ιδιότητας. Εάν η ιδιότητα είναι αναγκαστική θα επιλέξει το χαρακτηριστικό “with” , εάν η ιδιότητα δεν πρέπει να υπάρχει θα επιλέξει το χαρακτηριστικό ”without” και τέλος εάν η ιδιότητα δεν είναι αναγκαστική θα επιλέξει το χαρακτηριστικό ”maybe”[6].

Μετέπειτα οι χρήστες διαθέτουν τη δυνατότητα να επιλέξουν ένα φίλτρο (ισότητα, περιέχουν, δεν περιέχουν, ένας από, μεταξύ, περισσότερο από κ.λπ.) που θα μειώσει και θα περιορίσει το εύρος των αποτελεσμάτων. Επιπλέον δίνεται η δυνατότητα στο χρήστη να επιλέξει ένα αντικείμενο από έναν κατάλογο, ο οποίος παράγεται ανάλογα με τις προηγούμενες επιλογές του χρήστη (θέμα και κατηγορία) . Τέλος οι χρήστες διαθέτουν τη δυνατότητα να δημιουργήσουν περιορισμό για τον περιορισμό που έχουν ήδη δημιουργήσει επιλέγοντας το εικονίδιο + που βρίσκετε στη αριστερή πλευρά του περιορισμού. Το θέμα για το νέο περιορισμό θα προκύπτει από τον αρχικό περιορισμό. Έτσι δημιουργείτε ένα δέντρο όπως φαίνεται και στο παράδειγμα 3.2[6].

Τέλος ο χρήστης έχει τη δυνατότητα να δημιουργήσει περισσότερους περιορισμούς για το θέμα που επέλεξε ακολουθώντας την ίδια μέθοδο που προαναφέραμε. Τα αποτελέσματα που θα επιστρέψει το επερώτημα για το θέμα που επιλέξαμε θα πρέπει να ικανοποιεί όλους τους περιορισμούς που έθεσε ο χρήστης ανάλογα με τη βαρύτητα τους. Επιπλέον ο χρήστης μπορεί να θέσει ετικέτες στα αποτελέσματα του όπως φαίνεται στο παράδειγμα 3.2. Εάν ο χρήστης επιλέξει την ετικέτα τότε θα εμφανιστεί και θα παρουσιάσει τα αποτελέσματα που αντιπροσωπεύει. [6]

3.4. Δυνατότητες της MashQL.

Η MashQL προσφέρει αρκετές δυνατότητες στους χρηστές, επιτρέποντας τους να δημιουργούν εύκολα από πολύ απλά μέχρι αρκετά σύνθετα ερωτήματα. Σε αυτό το

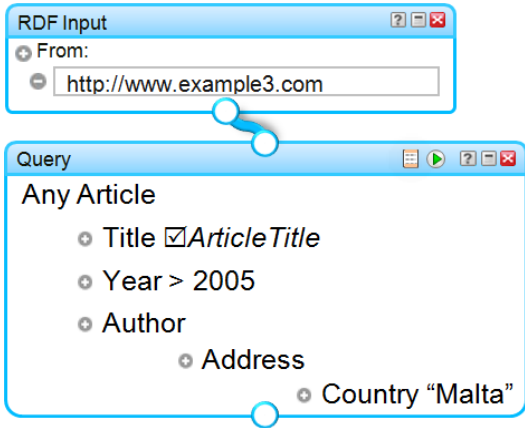
υποκεφάλαιο θα παρουσιάζονται οι δυνατότητες της MashQL μέσω παραδειγμάτων. Επιπλέον θα υποδείξουμε και το SPARQL ερώτημα που θα δημιουργήσει δυναμικά η MashQL.

3.4.1 Μεταβλητή στη MashQL:

Ο χρήστης έχει την δυνατότητα εάν θέλει να δημιουργήσει ένα ερωτήματα που θα αναφέρεται σε όλα τα θέματα που του προσφέρονται. Αυτό το επιτυγχάνει με το να μην επιλέξει κάποιο θέμα και να δώσει μια μεταβλητή στη θέση του θέματος. Η μεταβλητή θα εμπεριέχει όλες τις δυνατές τιμές που προσφέρονται από τα RDF αρχεία. Στο παράδειγμα 3.3 χρησιμοποιείται η μεταβλητή Any Article.

3.4.2 Ετικέτα στη MashQL:

Η MashQL δίνει τη δυνατότητα στο χρήστη να καθορίσει την ονομασία που θέλει να δώσει για τα συγκεκριμένα αποτελέσματα. Στο παράδειγμα 3.3 που ακολουθεί διακρίνουμε τη ετικέτα ArticleTitle. Δίπλα από την ετικέτα διακρίνουμε ένα checkbox, εάν το checkbox είναι επιλεγμένο τότε θα επιστρέψει τα αποτελέσματα που αναπαριστά η συγκεκριμένη μεταβλητή. Εάν δεν επιλεγεί τότε η ετικέτα θα έχει την έννοια της μεταβλητής. (SPARQL{?ArticleTitle.}) [6]



```
PREFIX S1: <http://www.example.com>

SELECT ?ArticleTitle
FROM <http://www.example.com?
WHERE {
    ?X1 rdf:type :Article.
    ?X1 S1:Title ?ArticleTitle.
    ?X1 S1:Year ?X2.
    FILTER (?X2 > 2005).
```

?X1 S1:Author ?X3.
 ?X3 S1:Address ?X4.
 ?X4 S1:Country ?X5.
 FILTER regex(?X5, "Malta")}

Results:

ArticleTitle (Ετικέτα)
Web 2.0

Παράδειγμα 3.3

3.4.3 Φίλτρα στη MashQL:

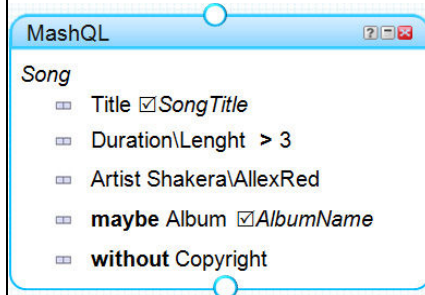
Επιπλέον η MashQL δίνει τη δυνατότητα στο χρήστη να χρησιμοποιεί φίλτρα στη υλοποίηση των ερωτημάτων του. Η δυνατότητα αυτή βοηθά το χρήστη να εκτελεί εξειδικευμένα επερωτήματα στα δεδομένα και επιπλέον να περιορίζει το εύρος των αποτελεσμάτων που θα του επιστρέψει ένα επερώτημα. Στο παράδειγμα 3.3 χρησιμοποιούνται δυο φορές τα φίλτρα. Τη πρώτη φορά τα χρησιμοποιεί για να συγκεκριμενοποιήσει τη χρονολογία του άρθρου (FILTER (?X2 > 2005)) και τη δεύτερη για να συγκεκριμενοποιήσει την καταγωγή του συγγραφέα (FILTER regex(?X5, "Malta")).[6]

3.4.5 Ο τελεστής AND στη MashQL:

Επιπλέον στο παράδειγμα 3.3 μπορούμε να παρατηρήσουμε ότι χρησιμοποιείται ο τελεστής AND. Ο τελεστής AND πραγματοποιείται με τον ορισμό περισσότερο του ενός περιορισμού στο θέμα μας. Στο παράδειγμα 3.3 δημιουργείται

αυτόματα η λογική εξίσωση Title && Year && Author (SPARQL{?X1 rdf:type :Article. ?X1 S1:Title ?ArticleTitle. ?X1 S1:Year ?X2. X1 S1:Author ?X3.}) και για να ισχύει το επερώτημα πρέπει να διαθέτει και τα τρία χαρακτηριστικά. [6]

Το παράδειγμα 3.4 που ακολουθεί περιέχει ακόμη περισσότερες δυνατότητες της MashQL. Το συγκεκριμένο παράδειγμα αναζητά όλους τους τίτλους τραγουδιών και εάν υπάρχουν οι συλλογές της τραγουδίστριας Shakira και του τραγουδιστή AxelleRed. Τα τραγούδια ή οι συλλογές πρέπει να έχουν διάρκεια μεγαλύτερη τριών λεπτών και επιπλέον τα αποτελέσματα δεν πρέπει να διαθέτουν Copyright. [6]



```

PREFIX a: <http://www.example.nam.com>
PREFIX S1: <http://www.example.si.com>
SELECT ?SongTitle, ?AlbumName
FROM <http://www.example.si.com>
WHERE {
    ?Song S1:Title ?SongTitle.
    {{?Song S1:Duration ?X1}
    UNION {?Song a:Length ?X1}}
    FILTER (?X1 > 3).
    {
        ?Song S1:Artist S1:Shakira}
        UNION {?Song S1:Artist
        S1:AxelleRed}
    }
    OPTIONAL {?Song S1:Album ?AlbumName}.
    OPTIONAL {?Song S1:Copyright ?X2}.
    FILTER (!Bound(?X2)).}

```

Παράδειγμα 3.4

3.4.6 UNION στη MashQL:

Η MashQL δίνει τη δυνατότητα να εκτελεστεί ένα ερωτήματα σε ένα ή περισσότερα θέματα. Αυτή η δυνατότητα προέρχεται με τη χρήση του τελεστή UNION. Ο τελεστής UNION συνδέει τα δυο θέματα και δημιουργεί ένα θέμα. Μετέπειτα ο χρήστης δημιουργώντας ένα ερωτήματα εξασφαλίζει την εκτέλεση του και στα δυο θέματα. Στο παράδειγμα 3.4 χρησιμοποιεί τον τελεστή UNION που στη MashQL αντιπροσωπεύεται από το χαρακτήρα (‘\’) για να δημιουργήσει ένα κοινό

επερώτημα για του τραγουδιστές Shakira και AxelleRed. [6] Στη SPARQL αντιπροσωπεύεται από τον παραγόμενο κώδικα :

```
{?Song S1:Artist S1:Shakira}  
UNION {?Song S1:Artist S1:AxelleRed}
```

3.4.7 Maybe στη MashQL:

Υπάρχουν περιπτώσεις που ο χρήστης θέλει να αναζητήσει πεδία από δεδομένα που πιθανώς να υπάρχουν αλλά παράλληλα θα ήθελε να εμφανιστούν και τα δεδομένα που δεν διαθέτουν αυτό το πεδίο. Η MashQL για να επιλύσει αυτό το πρόβλημα εισάγει την έννοια του Maybe. Στο παράδειγμα 3.4 χρησιμοποιείται το Maybe για να θέσει ως μη αναγκαστικό το πεδίο Album. [6] Στη SPARQL αντιπροσωπεύεται από το παραγόμενο κώδικα :

```
OPTIONAL {?Song S1:Album ?AlbumName}.
```

3.4.8 WITHOUT στη MashQL:

Τέλος με τη χρήση του WITHOUT ο χρήστης έχει τη δυνατότητα να αναζητήσει δεδομένα που δεν διαθέτουν ένα πεδίο. Στο παράδειγμα 3.4 χρησιμοποιεί το WITHOUT για να μην διαθέτουν τα αποτελέσματα του το Copyright. [6] Στη SPARQL αντιπροσωπεύεται από το παραγόμενο κώδικα :

```
OPTIONAL {?Song S1:Copyright ?X2}. FILTER (!Bound(?X2)).}
```

Κεφάλαιο 4. MashQL Firefox add-on.

4.1. Εισαγωγή στο MashQL Firefox add-on.	31
4.2 Εισαγωγή στο Firefox	31
4.3. Η λειτουργικότητα και οι κύριοι αλγόριθμοι που χρησιμοποιεί το MashQL Firefox add-on.	34

4.1. Εισαγωγή στο MashQL Firefox add-on.

Στο υποκεφάλαιο που ακολουθεί παρουσιάζεται το MashQL Firefox add-on. Με λίγα λόγια σ' αυτό το υποκεφάλαιο θα εξηγηθεί η μεθοδολογία για τη δημιουργία Firefox add-on αλλά και η δομή και οι σημαντικότεροι αλγόριθμοι για την υλοποίηση του MashQL Firefox add-on.

4.2 Εισαγωγή στο Firefox

4.2.1 Mozilla Firefox Web browser.

Ο Mozilla Firefox είναι ανοικτού κώδικα Web browser που προήλθε από το λογισμικό Mozilla Application Suite και η διαχείριση της ανάπτυξής του γίνεται από την Mozilla Corporation. Για την απεικόνιση των ιστοσελίδων, ο Firefox χρησιμοποιεί την μηχανή διάταξης Gecko, η οποία εφαρμόζει ορισμένα από τα σημερινά πρότυπα του Web συν μερικά χαρακτηριστικά που σκοπό έχουν να προλάβουν πιθανές προσθήκες των πρότυπων. Ο Firefox περιλαμβάνει αναδυόμενα παράθυρα, περιήγηση με καρτέλες, ορθογραφικό έλεγχο, επιμέρους εύρεση, ενεργούς σελιδοδείκτες, διαχείριση των μεταφορτώσεων και ένα ενσωματωμένο πεδίο αναζήτησης με δυνατότητα επιλογής της επιθυμητής μηχανής αναζήτησης. Περαιτέρω λειτουργίες

ενεργοποιούνται μέσω extensions που αναπτύχθηκαν από τρίτους. Ο Firefox λειτουργεί σε αρκετές εκδόσεις των Microsoft Windows, στο Mac OS X, στο Linux, και σε πολλά λειτουργικά συστήματα που προήλθαν από το Unix. Ο πηγαίος κώδικας του Firefox είναι ελεύθερο λογισμικό και διακινείται ελεύθερα με την τριπλή αδειοδότηση GPL/LGPL/MPL.

4.2.2 Ιστορία του Mozilla Firefox Web browser.

Ο Dave Hyatt και ο Blake Ross άρχισαν να αναπτύσσουν τον Firefox ως έναν πειραματικό κλάδο του Mozilla project. Προσφέροντας μια εναλλακτική πρόταση στο απαιτητικό σε πόρους Mozilla Suite, ανέπτυξαν παράλληλα έναν ελαφρύ, γρήγορο και εύκολα επεκτάσιμο browser, με τον οποίο ήλπιζαν ότι θα αλλάξουν τον κύριο αναπτυξιακό προσανατολισμό του Mozilla Foundation. Στις 3 Απριλίου 2003, το Mozilla Foundation ανακοίνωσε την αλλαγή πλευσης και από το Mozilla Suite, έστρεψαν την προσοχή τους στον Firefox και το Thunderbird.

Ο Firefox είχε αρκετές διαφορετικές ονομασίες κατά τα πρώιμα στάδια της ανάπτυξής του. Αρχικά ονομαζόταν Phoenix αλλά αργότερα μετονομάστηκε επειδή η λέξη αυτή ήταν δεσμευμένο εμπορικό σήμα της Phoenix Technologies. Η νέα ονομασία, Firebird, προκάλεσε νέες έντονες αντιδράσεις από την κοινότητα ανάπτυξης ελεύθερων βάσεων δεδομένων με το ίδιο όνομα. Οι συνεχόμενες πιέσεις όμως από την πλευρά της κοινότητας ανάπτυξης των βάσεων δεδομένων οδήγησαν σε άλλη μια αλλαγή ονομασίας. Στις 9 Φεβρουαρίου 2004, ανακοινώθηκε το όνομα που έχει ακόμη σήμερα, Mozilla Firefox ή απλά Firefox όπως συχνά αναφέρεται. Αναφορά στον Firefox γίνεται και με την συντόμευση FF, αν και η Mozilla προτιμά την συντόμευση Fx ή fx.

Το αναπτυξιακό σχέδιο του Firefox είχε πολλές εκδόσεις πριν την έκδοση 1.0 που κυκλοφόρησε στις 9 Νοεμβρίου 2004. Μετά την βελτίωση μιας αλληλουχίας στοιχείων σταθερότητας και ασφάλειας, η Mozilla κυκλοφόρησε, στις 29 Νοεμβρίου

2005, την πρώτη σημαντική αναβάθμιση, τον Firefox 1.5. Στις 24 Οκτωβρίου 2006, η Mozilla κυκλοφόρησε τον Firefox 2. Αυτή η έκδοση περιείχε αναβαθμίσεις για το «σερφόρισμα» με καρτέλες, την διαχείριση επεκτάσεων, το γραφικό περιβάλλον, την λειτουργία εύρεσης και αναζήτησης και την λειτουργία αναβάθμισης του λογισμικού. Τον Δεκέμβριο του 2007, τέθηκε σε λειτουργία το Firefox Live Chat που δίνει την δυνατότητα σε χρήστες να θέτουν ερωτήματα υποστήριξης σε εθελοντές. Λειτουργεί σε πραγματικό χρόνο καθορισμένες ώρες καθημερινά, ενώ υπάρχει και η πιθανότητα βοήθειας και πέραν αυτών.

4.2.3 Mozilla Firefox add-on.

Με τη εισαγωγή ενός add-on στο Firefox εισαγάγουμε επιπλέον functionality στο Mozilla application. Τα add-on δημιουργήθηκαν για να μπορεί ο κάθε χρήστης να προσθέσει λειτουργικότητες που θα ικανοποιούν τις ανάγκες του[8]. Η δομή ενός add-on είναι όπως παρουσιάζεται στο παράδειγμα 4.1

exampleExt.xpi:

```
/install.rdf  
/chrome.manifest  
/chrome/  
/chrome/content/
```

Παράδειγμα 4.1

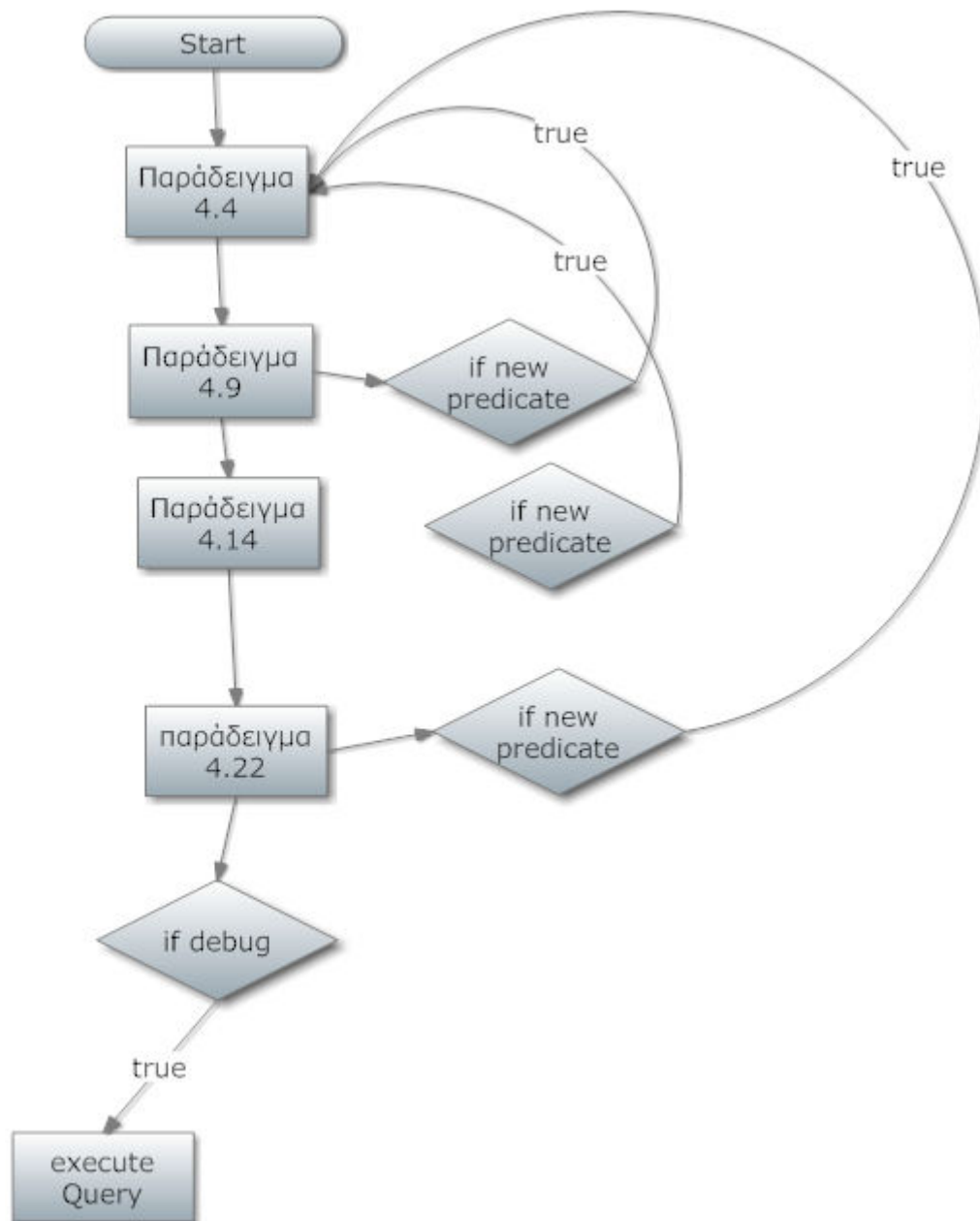
Ένα add-on Firefox έχει επέκταση xpi. Η κωδικοποίηση xpi είναι η ίδια κωδικοποίηση που χρησιμοποιεί η τεχνολογία zip. Εντός του xpi εντοπίζουμε το chrome.manifest αρχείο το οποίο περιέχει οδηγίες εγκατάστασης και τα δεδομένα που πρόκειται να εγκατασταθούν. Επιπλέον εντοπίζουμε το install.rdf που περιέχει μεταδεδομένα όπως για παράδειγμα τις πληροφορίες για το ποιος το δημιούργησε και με ποιες εκδόσεις του Firefox είναι συμβατό κτλ. Τέλος εντοπίζουμε το φάκελο chrome

που περιέχει το φάκελο content που περιέχει τα .xul, τα .js και όλα τα υπόλοιπα συμβατά αρχεία.

. Η XUL είναι μια cross-platform γλώσσα η οποία χρησιμοποιείται για περιγραφή των user interfaces των εφαρμογών. Η λειτουργικότητα τους καθορίζεται από τα JavaScript αρχεία. Ένα XUL αρχείο μπορεί να έχει ένα ή περισσότερα script αρχεία που συσχετίζεται με αυτό/ά. Η Mozilla χειρίζεται την XUL ακριβώς με τον ίδιο τρόπο όπως την HTML. Επιπλέον παρέχει τη δυνατότητα για δημιουργία σχεδόν όλων των element που χρησιμοποιούνται στα μοντέρνα graphical interfaces όπως Textbox, Checkboxes, Toolbars, Pop up menu , Tabbed dialog και πολλά άλλα. Η JavaScript είναι μια αντικειμενοστραφής scripting γλώσσα προγραμματισμού η οποία τρέχει σε πολλές πλατφόρμες. Στα Firefox add-on η XUL είναι υπεύθυνη για το user interface και η JavaScript για τη δημιουργία.

4.3. Η λειτουργικότητα και οι κύριοι αλγόριθμοι που χρησιμοποιεί το MashQL Firefox add-on.

Στο υποκεφάλαιο που ακολουθεί ορίζεται η δομή του MashQL Firefox add-on καθώς καθορίζονται οι κύριοι αλγόριθμοι που υλοποιούνται. Στο παράδειγμα 4.2 που ακολουθεί ορίζεται το διάγραμμα ροής για το MashQL Firefox add-on. Στη συνέχεια θα επεξηγηθούν οι αλγόριθμοι για κάθε μέρος του πιο κάτω διαγράμματος ροής καθώς και για το ίδιο το διάγραμμα.

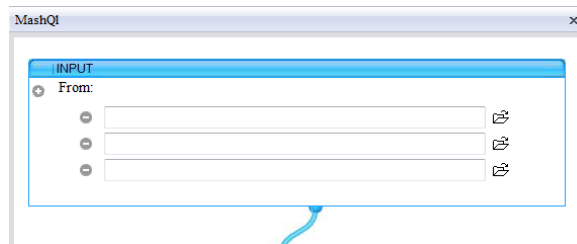


Παράδειγμα 4.2

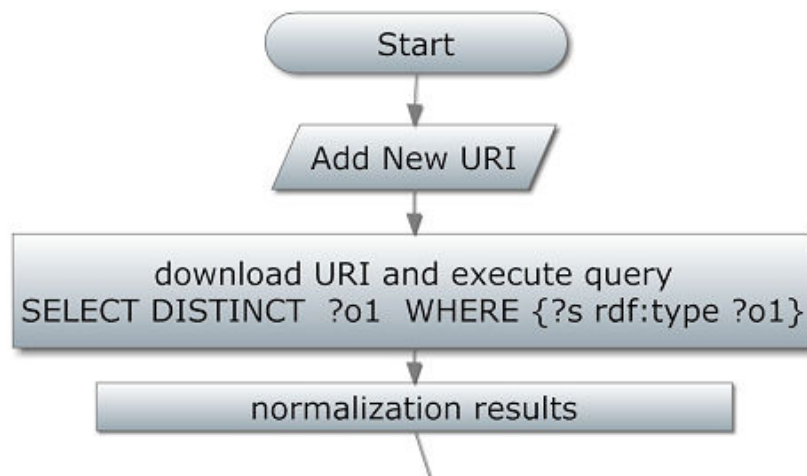
4.3.1. Εισαγωγή URI και εξόρυξη των αρχικών Subject.

Το πρώτο βήμα για τη λειτουργία MashQL Firefox add-on είναι να τοποθετήσουμε τα URI των RDF αρχείων. Στο παράδειγμα 4.3 παρουσιάζεται το μέρος του add-on που τοποθετεί ο χρήστης τα URI που αντιπροσωπεύουν τα RDF. Εάν ο

χρήστης χρειάζεται περισσότερα από τρία URI τότε του δίνεται η δυνατότητα να αυξήσει τον αριθμό τους επιλέγοντας το εικονίδιο δίπλα από το From. Εάν ο χρήστης θέλει να διαγράψει ένα URI τότε πρέπει να επιλέξει το εικονίδιο που βρίσκεται στην αριστερή πλευρά του textbox. Τέλος ο χρήστης έχει την ικανότητα να εντοπίσει το URI του RDF που βρίσκεται στο προσωπικό του υπολογιστή επιλέγοντας το εικονίδιο που βρίσκεται στη δεξιά πλευρά του textbox. Η πιο πάνω διαδικασία αντιπροσωπεύεται στο διάγραμμα ροής από το Add New URI όπως φαίνεται και στο παράδειγμα 4.4.



Παράδειγμα 4.3



Παράδειγμα 4.4

```

PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT DISTINCT    ?o1
FROM <URI1>
.....
FROM <URI n>
WHERE {    ?s rdf:type ?o1. }
  
```

Παράδειγμα 4.5

Εφόσον ο χρήστης εισάγει τα URI τότε το add-on αναλαμβάνει να εντοπίσει τα Subject που είναι διαθέσιμα στα αρχεία. Καταρχήν, για τον εντοπισμό των δεδομένων αυτοματοποιημένα δημιουργείτε το επερώτημα του παραδείγματος 4.5, του οποίου τα αποτελέσματα από την εκτέλεση του επερώτηματος είναι τα Subject . Η δημιουργία του επερώτηματος γίνεται από το αλγόριθμο που καθορίζει το παράδειγμα 4.6. Με το τέλος του αλγόριθμου που υλοποιείται από το παράδειγμα 4.6. το επερώτημα εκτελείται με τη βοήθεια της βιβλιοθήκης Jena που είναι υλοποιημένη σε java.

```
var query= 'PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> '  
query+= ' SELECT DISTINCT    ?o1 '  
foreach( URI )  
    query+= 'FROM ' + URI;  
query+= ' WHERE '  
query+= ' {    ?s rdf:type ?o1. }'
```

Παράδειγμα 4.6

Τέλος τα αποτελέσματα που πιθανώς είναι URI πρέπει να κανονικοποιηθούν για να είναι πιο κατανοητά για το χρήστη. Ένα URI αποτελείται από δυο μέρη το namespace και το prefix. Με το τέλος της κανονικοποίησης το θεμιτό αποτέλεσμα είναι το prefix. Για περισσότερη ακρίβεια το add-on περιέχει ένα αρχείο με τα πιο δημοφιλή URI που εντοπίζουμε στα RDF όπως το <http://www.w3.org/1999/02/22-rdf-syntax-ns#>. Εάν το URI δεν υπάρχει στη λίστα των πιο δημοφιλή URI τότε εφαρμόζεται ο αλγόριθμος του παραδείγματος 4.7 που έχει σκοπό να εντοπίσει το prefix. Ο συγκεκριμένος αλγόριθμος παίρνει την τελευταία συμβολοσειρά από ένα URI μετά από τους χαρακτήρες '#' ή '/' . Εάν η συμβολοσειρά είναι μικρότερη των 3

χαρακτήρων τότε χρησιμοποιούμε τη αμέσως προηγούμενη συμβολοσειρά μέχρι να βρούμε κάποια συμβολοσειρά που να ικανοποιεί την πιο πάνω συνθήκη. Το συγκεκριμένο μέρος είναι εμφανές στο διάγραμμα ροής του παραδείγματος 4.4.

```
foreach( result in x){
    temp=false;
    foreach(prefixList in i)
        if(prefixList[i]== result[x].URI){
            result[x].prefix= prefixList[i];
            temp=true;
        }

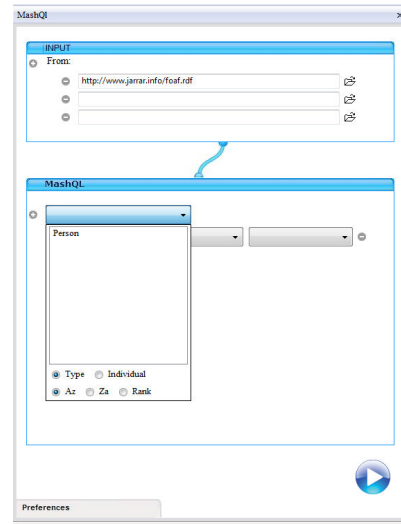
        array=result[x].URI.split("#");
        for(array.lenght in z && !temp)
            if(array[array.lenght-z].lenght>3){
                result[x].prefix= array[array.lenght-z];
                temp=true;
            }
    If(!temp)
        result[x].prefix= result[x].URI;
}
```

Παράδειγμα 4.7

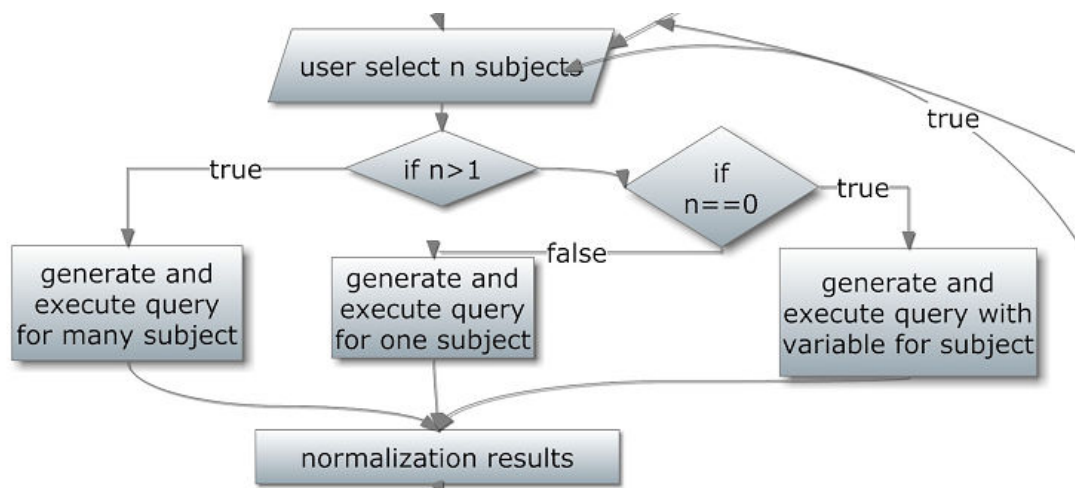
4.3.2. Επιλογή Subject και εξαγωγή των ανάλογων predicate.

Στη συνέχεια ο χρήστης έχει τη δυνατότητα να επιλέξει η Subject συνεχίζοντας τη δημιουργία του επρωτήματος που τον ενδιαφέρει.. Στο διάγραμμα ροής που παρουσιάζεται από το παράδειγμα 4.9 υποδεικνύεται η μέθοδος λειτουργίας του add-on

μετά από την επιλογή Subject. Στο παράδειγμα 4.8 εμφανίζεται η λίστα με τα υπάρχοντα Subject που μπορεί ο χρήστης να επιλέξει. Τέλος ανάλογα με τον αριθμό των Subject που θα επιλεγούν θα ενεργοποιηθεί ο ανάλογος αλγόριθμος. Πιο κάτω επεξηγούνται οι τρεις πιθανοί αλγόριθμοι επιλογής θέματος.



Παράδειγμα 4.8



Παράδειγμα 4.9

Εάν ο χρήστης επιλέξει ένα Subject τότε για τη παραγωγή του ερωτήματος θα εκτελεστεί ο αλγόριθμος που εμφανίζεται στο παράδειγμα 4.10. Ο συγκεκριμένος αλγόριθμος επεκτείνει το String Query που περιέχει το μέχρι τώρα υλοποιημένο ερώτημα. Συγκεκριμένα αντικαθιστά το υφιστάμενο object με το επιλεγμένο subject και τέλος προσθέτει μια νέα τριπλέτα στο ερωτήμα που αποτελείται από τη μεταβλητή “?s” που αντιπροσωπεύει το subject, τη μεταβλητή “?p” που αντιπροσωπεύει το

predicate και τέλος τη μεταβλητή που αντιπροσώπευε το προηγούμενο object που αντικαταστάθηκε με το επιλεγμένο subject.

```
addNewSubject(var Query, var lastObjectVariable, var newSubject){  
    Query=Query.replace(lastObjectVariable,newSubject)+ “?s ?p ?”+  
        lastObjectVariable+”. ” ;  
    return Query;  
}
```

Παράδειγμα 4.10

Εάν ο χρήστης δεν επιλέξει Subject τότε για τη παραγωγή του επερωτήματος θα εκτελεστεί ο αλγόριθμος που εμφανίζεται στο παράδειγμα 4.11. Ο συγκεκριμένος αλγόριθμος επεκτείνει το String Query που περιέχει το μέχρι τώρα υλοποιημένο ερώτημα. Συγκεκριμένα προσθέτει μια νέα τριπλέττα στο επερώτημα που αποτελείται από τη μεταβλητή “?s” που αντιπροσωπεύει το subject, τη μεταβλητή “?p” που αντιπροσωπεύει το predicate και τέλος αναδημιουργεί μια νέα μεταβλητή βασισμένη στη μεταβλητή που αντιπροσώπευε το προηγούμενο object συγκεκριμένα εάν η προηγούμενη μεταβλητή είναι ?o1 η νέα μεταβλητή θα είναι ?o2.

```
addNewVariableAsSubject(var Query, var lastObjectVariable){  
    Query+= “?s ?p ?”+generateNewVariable( lastObjectVariable)+”. ” ;  
    return Query;  
}
```

Παράδειγμα 4.11

Εάν ο χρήστης επιλέξει περισσότερα από ένα Subject τότε για τη παραγωγή του επερωτήματος θα εκτελεστεί ο αλγόριθμος που εμφανίζεται στο παράδειγμα 4.12. Ο συγκεκριμένος αλγόριθμος επεκτείνει το String Query που περιέχει μέσα το μέχρι τώρα

υλοποιημένο ερώτημα. Συγκεκριμένα διασπά τη συμβολοσειρά με το χαρακτήρα '\'. Μετέπειτα χρησιμοποιούμε το αλγόριθμο του παραδείγματος 4.10 για κάθε μια από τις νέες μεταβλητές. Τέλος δημιουργούμε μια καινούργια συμβολοσειρά συνδυάζοντας τα αποτελέσματα που θα πάρουμε από τη συνάρτηση. Για τη δημιουργία της καινούργιας συμβολοσειράς όπως αναφέραμε χρησιμοποιούμε τα αποτελέσματα που μας δίνονται από το αλγόριθμο του παραδείγματος 4.10 τα οποία συνδέει με τον τελεστή UNION.

Τέλος εκτελείται το ερώτημα και τα αποτελέσματα του κανονικοποιούνται. Για να κανονικοποιηθούν τα αποτελέσματα μας χρησιμοποιούμε τον αλγόριθμο που επεξηγήσαμε στο παράδειγμα 4.7.

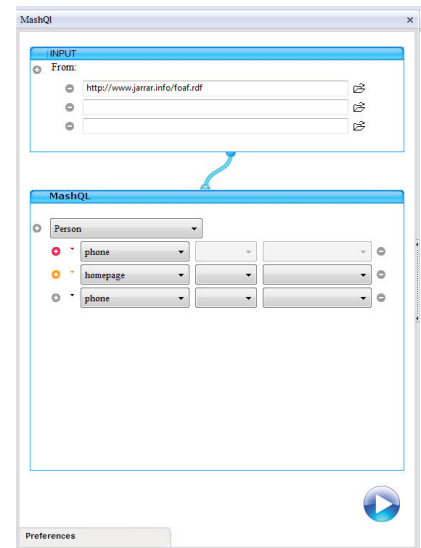
```
addManySubject(var Query, var lastObjectVariable, var newSubject){  
    var temp=null;  
    var _newSubject= newSubject.split("\\");  
    foreach(_newSubject in i){  
        if(temp==null)  
            temp="{ "+addNewSubject(Query, lastObjectVariable, _newSubject[i])+"}";  
        else  
            temp="UNION { "+addNewSubject(Query, lastObjectVariable,  
                _newSubject[i])+"}";  
    }  
    return temp;  
}
```

Παράδειγμα 4.12

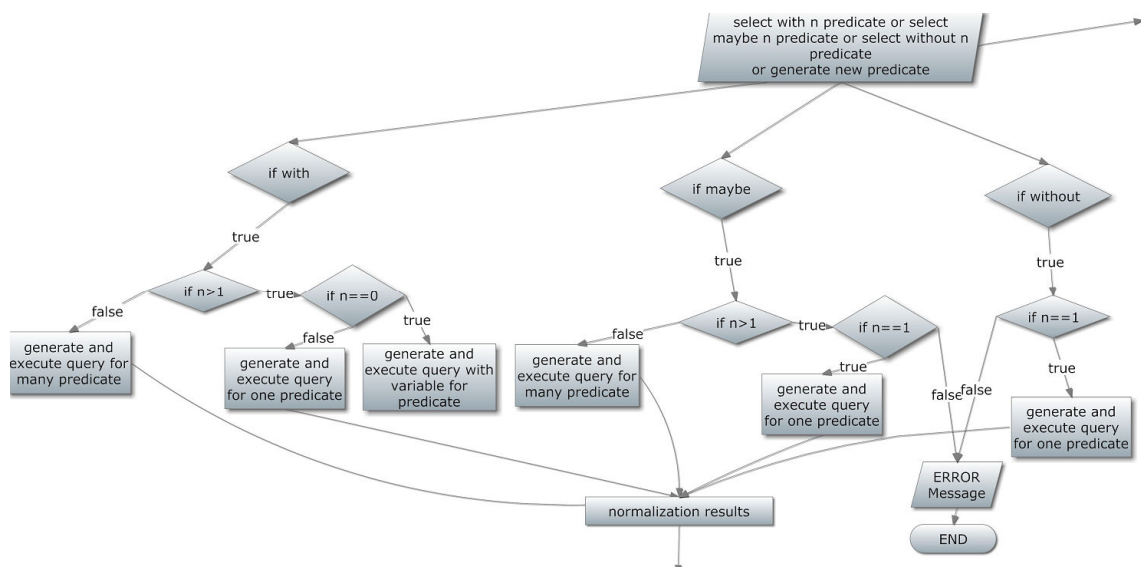
4.3.1. Επιλογή predicate.

Στη συνέχεια ο χρήστης έχει τη δυνατότητα να επιλέξει η predicate συνεχίζοντας τη δημιουργία του επερωτήματος που τον ενδιαφέρει. Επιπλέον δίνεται η δυνατότητα στο χρήστη να επιλέξει ιδιότητα του επιλεγμένου predicate. Εάν η ιδιότητα είναι αναγκαστική θα επιλέξει το χαρακτηριστικό “with”, εάν η ιδιότητα δεν πρέπει να υπάρχει θα επιλέξει το χαρακτηριστικό “without” και τέλος εάν η ιδιότητα δεν είναι αναγκαστική θα επιλέξει το χαρακτηριστικό “maybe”. Στο διάγραμμα ροής που

παρουσιάζεται από το παράδειγμα 4.14 υποδεικνύεται η μέθοδος λειτουργίας του add-on κατά τη διάρκεια και μετά την επιλογή η predicate. Τέλος ανάλογα με τον αριθμό των predicate και το βάρος που θα επιλεγούν θα ενεργοποιηθεί ο ανάλογος αλγόριθμος. Στο παράδειγμα 4.13 παρουσιάζεται η μέθοδος εισαγωγής ιδιοτήτων στο ερώτημα. Το κόκκινο κουμπί αναπαριστά το “without” το πορτοκαλί το “maybe” και το γκρίζο αναπαριστά το “with”. Πιο κάτω επεξηγούνται οι έξι πιθανοί αλγόριθμοι επιλογής predicate.



Παράδειγμα 4.13



Παράδειγμα 4.14

Εάν ο χρήστης επιλέξει ένα predicate και θέσει σαν αναγκαστικό τότε για τη παραγωγή του επερωτήματος θα εκτελεστεί ο αλγόριθμος που εμφανίζεται στο παράδειγμα 4.15. Ο συγκεκριμένος αλγόριθμος επεκτείνει το String Query που περιέχει το μέχρι τώρα υλοποιημένο ερώτημα. Συγκεκριμένα αντικαθιστά την υφιστάμενη μεταβλητή που αναπαριστά το predicate με το επιλεγμένο predicate.

```
addNewPredicate (var Query, var lastPredicateVariable, var newPredicate){  
    Query=Query.replace(lastPredicateVariable,newPredicate)+"'." ;  
    return Query;  
}
```

Παράδειγμα 4.15

Επιπλέον ο χρήστης έχει τη δυνατότητα να θέσει το predicate σαν αναγκαστικό και να μην επιλέξει predicate θέτοντας μια μεταβλητή. Τότε για τη παραγωγή του επερωτήματος θα εκτελεστεί ο αλγόριθμος που εμφανίζεται στο παράδειγμα 4.16. Ο συγκεκριμένος αλγόριθμος αντικαθιστά στο String Query που περιέχει το μέχρι τώρα υλοποιημένο ερώτημα, την μεταβλητή που αναπαριστά το predicate με την μεταβλητή που έδωσε ο χρήστης.

```
addNewVariableAsPredicate(var Query, var lastPredicateVariable, var  
                           newPredicateVariable){  
    Query=Query.replace(lastPredicateVariable,newPredicateVariable)+"'." ;  
    return Query;  
}
```

Παράδειγμα 4.16

Εάν ο χρήστης επιλέξει περισσότερα από ένα predicate και θέσει το predicate σαν αναγκαστικό τότε για τη παραγωγή του επερωτήματος θα εκτελεστεί ο αλγόριθμος

που εμφανίζεται στο παράδειγμα 4.17. Ο συγκεκριμένος αλγόριθμος αντικαθιστά στο String Query που περιέχει μέσα το μέχρι τώρα υλοποιημένο ερώτημα. Συγκεκριμένα διασπά τη συμβολοσειρά με το χαρακτήρα '\'. Μετέπειτα χρησιμοποιούμε το αλγόριθμο του παραδείγματος 4.15 για κάθε μια από τις νέες μεταβλητές. Τέλος δημιουργούμε μια καινούργια συμβολοσειρά συνδυάζοντας τα αποτελέσματα που θα πάρουμε από τη συνάρτηση. Για τη δημιουργία της καινούργιας συμβολοσειράς όπως αναφέραμε χρησιμοποιούμε τα αποτελέσματα που μας δίνονται από το αλγόριθμο του παραδείγματος 4.15 συνδέοντας τα με το τελεστή UNION.

```
addMany Predicate (var Query, var last PredicateVariable, var new Predicate){  
    var temp=null;  
    var _new Predicate = new Predicate.split("\\");  
    foreach(_new Predicate in i){  
        if(temp==null)  
            temp="{ "+addNew Predicate (Query, last PredicateVariable,  
                _newPredicate [i])+"}";  
        else  
            temp="UNION { "+addNewPredicate (Query, lastPredicateVariable,  
                _newPredicate[i])+"}";  
    }  
    return temp; }
```

Παράδειγμα 4.17

Όπως προαναφέρθηκε με την επιλογή των predicate δίνεται η δυνατότητα στο χρήστη να επιλέξει ιδιότητα του επιλεγμένου predicate. Στις προηγούμενες παραγράφους περιγράφεται η εισαγωγή αναγκαστικού predicate, στις παραγράφους που ακολουθούν θα εξηγηθεί πως υλοποιείται η εισαγωγή προαιρετικού predicate. Η κύρια διαφορά των δυο ιδιοτήτων είναι η χρήση του τελεστή Optional. Επιπλέον θέτοντας το predicate σαν προαιρετικό ο χρήστης επιβάλλεται να επιλέξει ένα ή

περισσότερα predicate. Κατ' ακρίβεια είναι άσκοπο ο χρήστης να θέσει μεταβλητή η να μην επιλέξει predicate που ουσιαστικά θέτει αυτόματα το predicate σαν μεταβλητή.

Εάν ο χρήστης επιλέξει ένα predicate και το θέσει σαν προαιρετικό τότε για την παραγωγή του ερωτήματος θα εκτελεστεί ο αλγόριθμος που εμφανίζεται στο παράδειγμα 4.18. Ο συγκεκριμένος αλγόριθμος επεκτείνει το String Query που περιέχει το μέχρι τώρα υλοποιημένο ερώτημα. Συγκεκριμένα αντικαθιστά την υφιστάμενη μεταβλητή που αναπαριστά το predicate με το επιλεγμένο predicate και προσθέτει το τελεστή Optional που καθιστά το συγκεκριμένο μέρος του ερωτήματος σαν προαιρετικό. Για εύρεση της θέσης που θα τοποθετηθεί ο τελεστής Optional πρέπει να εντοπιστεί η τελευταία τριπλέτα για να αντικατασταθεί με τη νέα τριπλέτα που θα εμπεριέχει το τελεστή και το επιλεγμένο predicate. Η τελευταία τριπλέτα που προστέθηκε στο ερώτημα είναι αποθηκευμένη στη μεταβλητή με την ονομασία lastTripleVariable.

```
addNewPredicateMaybe (var Query, var lastPredicateVariable ,var  
                        lastTripleVariable, var newPredicate){  
    var temp= lastPredicateVariable;  
    temp=" OPTIONAL {"'+temp.replace(lastPredicateVariable,newPredicate)+"'."};  
    Query=Query.replace(lastTripleVariable, temp)+"'." ;  
    return Query;  
}
```

Παράδειγμα 4.18

Όπως προαναφέρθηκε ο χρήστης έχει τη δυνατότητα να επιλέξει περισσότερα από ένα predicate όταν θέσει το predicate σαν προαιρετικό. Ο αλγόριθμος που θα εκτελέσει την παραγωγή του ερωτήματος εμφανίζεται στο παράδειγμα 4.19. Ο συγκεκριμένος αλγόριθμος αντικαθιστά στο String Query που περιέχει μέσα το μέχρι

τώρα υλοποιημένο ερώτημα. Συγκεκριμένα διασπά τη συμβολοσειρά με το χαρακτήρα ‘\’. Μετέπειτα χρησιμοποιούμε το αλγόριθμο του παραδείγματος 4.18 για κάθε μια από τις νέες μεταβλητές. Τέλος δημιουργούμε μια καινούργια συμβολοσειρά συνδυάζοντας τα αποτελέσματα που θα πάρουμε από τη συνάρτηση. Για τη δημιουργία της καινούργιας συμβολοσειράς όπως αναφέραμε χρησιμοποιεί τα αποτελέσματα που δίνονται από το αλγόριθμο του παραδείγματος 4.18 τα οποία συνδέει τα με τον τελεστή UNION.

```

addMany PredicateMaybe (var Query, var lastPredicateVariable ,var
                                lastTripleVariable, var newPredicate){

    var temp=null;
    var _new Predicate = new Predicate.split("\");
    foreach(_new Predicate in i){
        if(temp==null)
            temp="{" + addNewPredicateMaybe (Query, last PredicateVariable,
                                                lastTripleVariable, _newPredicate [i])+"}";
        else
            temp="UNION {" + addNewPredicateMaybe (Query, lastPredicateVariable,
                                                lastTripleVariable ,_newPredicate[i])+"}";
    }
    return temp; }

```

Παράδειγμα 4.19

Κλείνοντας αυτό το υποκεφάλαιο πρέπει να αναφερθεί και ο αλγόριθμος επιλογής predicate που δεν πρέπει να υπάρχει. Ο συγκεκριμένος αλγόριθμος χρησιμοποιεί τη συνάρτηση addNewPredicateMaybe από το παράδειγμα 4.18 και προσθέτει στο τέλος το φίλτρο FILTER(!Bound(‘?object’)). Το συγκεκριμένο φίλτρο αφαιρεί όλα τα object που περιλαμβάνονται στη μεταβλητή που δέχεται. Με αυτό το τρόπο τα αποτελέσματα του ερωτήματος που βρίσκονται εντός της μεταβλητής δεν θα παρουσιαστούν στα τελικά αποτελέσματα. Ο αλγόριθμος παρουσιάζεται στο παράδειγμα 4.20.

Τέλος ο χρήστης σε αυτό το στάδιο έχει τη δυνατότητα να επιστρέψει στο βήμα επιλογής subject και να επιλέξει εκ νέου κάποια άλλα subject.

```
add PredicateWithOut (var Query, var lastPredicateVariable , var
    lastObjectVariable ,var lastTripleVariable, var newPredicate){

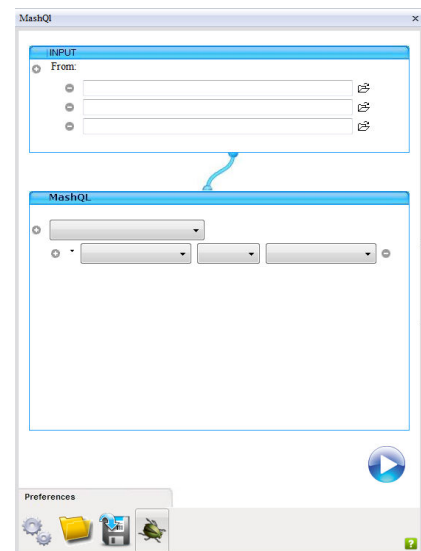
temp= addNewPredicateMaybe (Query, lastPredicateVariable,
    lastTripleVariable ,newPredicate)+”. FILTER(!Bound( “
    +lastObjectVariable +” )) .”};

return temp;
}
```

Παράδειγμα 4.20

4.3.1. Επιλογή Object και δημιουργία φίλτρων.

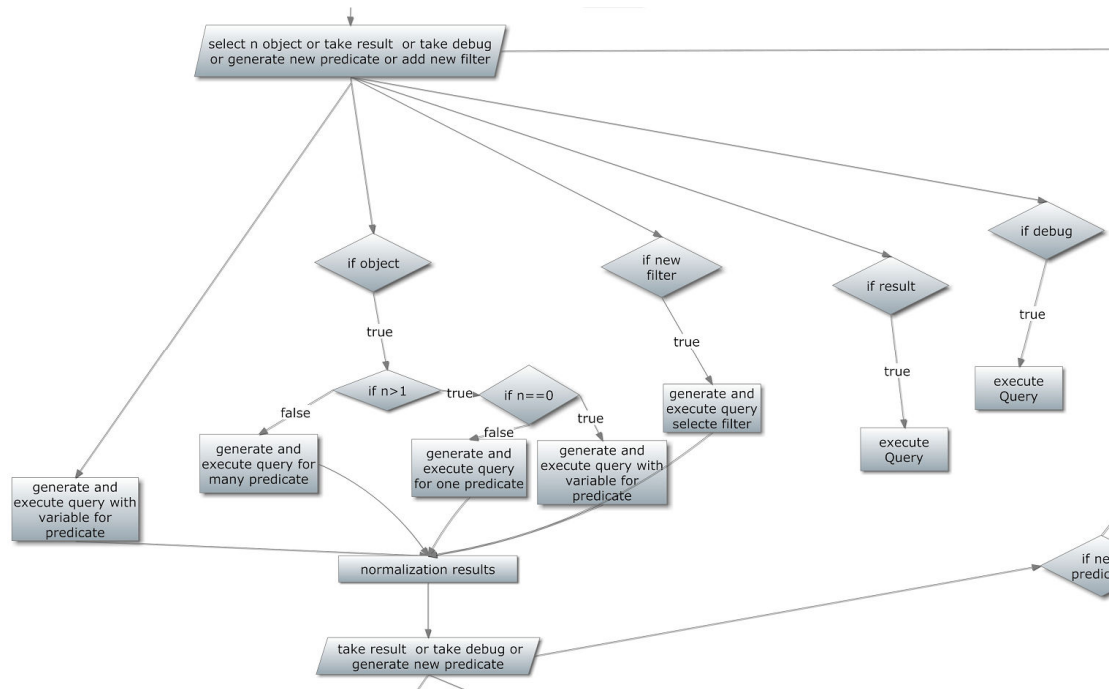
Στο παράδειγμα 4.22 παρουσιάζεται το μέρος από το διάγραμμα ροής που είναι υπεύθυνο για τη δημιουργία του ερωτήματος μετά την επιλογή Object. Στο παράδειγμα 4.22 μπορούμε να διακρίνουμε ότι ο αλγόριθμος για επιλογή η Object είναι ο ίδιος με τον αλγόριθμο επιλογής η subject. Στο διάγραμμα ροής επιπλέον διακρίνουμε τη δυνατότητα του χρήστη να μην επιλέξει Object και να εκτελέσει το ερώτημα παίρνοντας τα αποτελέσματα. Η εκτέλεση του ερωτήματος έχει τη δυνατότητα να υλοποιηθεί με δύο



παράδειγμα 4.21

τρόπους, ο πρώτος τρόπος είναι να επιλεγεί το κουμπί RUN όπως φαίνεται στο παράδειγμα 4.21 κάτω αριστερά και ο δεύτερος τρόπος είναι να επιλεγεί το κουμπί

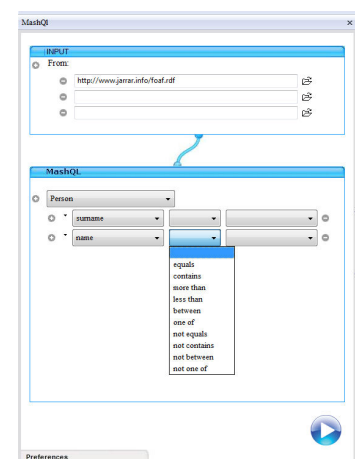
debug που βρίσκεται στη αναδυόμενη μπάρα όπως φαίνεται στο παράδειγμα 4.21. Με το πρώτο τρόπο θα επιστραφούν μόνο τα αποτελέσματα, αντίθετα με το δεύτερο τρόπο ο χρήστης έχει τη δυνατότητα να δει και το SPARQL QUERY που υλοποιήθηκε. Επιπλέον ο χρήστης έχει τη δυνατότητα να αγνοήσει αυτό το στάδιο και να προχωρήσει το ερώτημα προσθέτοντας καινούργια Predicate.



παράδειγμα 4.22

παράδειγμα 4.23

Τέλος στο παράδειγμα 4.22 διακρίνεται και ακόμη μια δυνατότητα του χρήστη, να προσθέτει φίλτρο. Στο παράδειγμα 4.23 παρουσιάζεται η μέθοδος επιλογής φίλτρου στο MashQL add-on. Ο χρήστης έχει τη δυνατότητα να επιλέξει το φίλτρο που χρειάζεται από την αναδυόμενη λίστα όπως παρουσιάζεται στο παράδειγμα 4.23. Μετέπειτα ο χρήστης πρέπει να εισάγει τη συμβολοσειρά ή τον αριθμό που επιθυμεί να



χρησιμοποιήσει στο φίλτρο του. Στο παράδειγμα 4.24 παρουσιάζεται ο αλγόριθμος που εισάγει φίλτρα στο ερώτημα.

```
///Set Filter
if(filter=="not equals"){
    if(IsNumeric(menuListObject))
        choice= 'FILTER ( '?' +lastObject+'!=' +menuListObject+' )';
    else
        choice= 'FILTER ( str(?' +lastObject+'!=' +menuListObject+' )';

    return choice;
}
///---
else if(filter=="equals"){
    if(IsNumeric(menuListObject))
        choice= 'FILTER ( '?' +lastObject+'=' +menuListObject+' )';
    else
        choice= 'FILTER ( str(?' +lastObject+'=' +menuListObject+' )';

    return choice;
}
///----
else if(filter==" not contains "){
    if(IsNumeric(menuListObject))
        choice= 'FILTER ( !regex( '?' +lastObject+' ,'" +menuListObject+' , "i") )';
    else
        choice= 'FILTER ( ! regex( str(?' +lastObject+' ) ,'" +menuListObject+' , "i") )';

    return choice;
}
else if(filter=="contains "){
    if(IsNumeric(menuListObject))
        choice= 'FILTER ( regex( '?' +lastObject+' ,'" +menuListObject+' , "i") )';
    else
        choice= 'FILTER ( regex( str(?' +lastObject+' ) ,'" +menuListObject+' , "i") )';

    return choice;
}
else if(filter==" more than"){
    if(IsNumeric(menuListObject))
        choice= 'FILTER ( '?' +lastObject+'>' +menuListObject+' )';
    else
        choice= 'FILTER ( str(?' +lastObject+'>' +menuListObject+' )';

    return choice;
```

```

}
else if(filter==" less than "){
    if(IsNumeric(menuListObject))
        choice= 'FILTER ( '?' +lastObject+'>'+" +menuListObject+" )';
    else
        choice= 'FILTER (str( '?' +lastObject+'>'+" +menuListObject+" ) )';

    return choice;
}

else if(filter=="between"){
    choice= 'FILTER (regex( '?' +lastObject+' , '^'+menuListObject+'.* '+'
                                                menuListObject2+'$', "i"))';
    return choice;
}

else if(filter=="not between"){
    choice= 'FILTER (!regex( '?' +lastObject+' , '^'+menuListObject+'.* '+'
                                                menuListObject2+'$', "i"))';
    return choice;
}

else if(filter=="not one of"){
    var objectTemp= menuListObject.split(";");
    var temp="";
    for( i in objectTemp)
        if(temp=="")

            temp = ' FILTER ( '?' +lastObject+' !='+ objectTemp[i]+' "' ;
        else
            temp+= '||' + '?' +lastObject+' !='+ objectTemp[i]+' "' ;
    return temp +").";
}
///---
else if(filter=="not one of"){
    var objectTemp= menuListObject.split(";");
    var temp="";
    for( i in objectTemp)
        if(temp=="")

            temp = ' FILTER ( '?' +lastObject+' =' + objectTemp[i]+' "' ;
        else
            temp+= '||' + '?' +lastObject+' =' + objectTemp[i]+' "' ;
    return temp +").";
}

```

Επιλέγοντας ο χρήστης το Object έχει τη δυνατότητα να συνεχίσει τη δημιουργία του ερωτήματος του με την εισαγωγή καινούργιου predicate ακολουθώντας την ίδια διαδικασία. Επιπλέον ο χρήστης έχει τη δυνατότητα να εξάγει τα αποτελέσματα του ερωτήματος που θα παρουσιαστούν σε μορφή ιστοσελίδας στο παράθυρο του φυλλομετρητή του όπως φαίνεται και στο παράδειγμα 4.25. Τέλος ο χρήστης έχει τη δυνατότητα να εξάγει τα αποτελέσματα του ερωτήματος και το υλοποιημένο ερώτημα που θα παρουσιαστούν σε μορφή ιστοσελίδας στο παράθυρο του φυλλομετρητή του όπως φαίνεται και στο παράδειγμα 4.26.

Τέλος στο παράρτημα Β βρίσκετε αναλυτικότερα η μέθοδος χρήσης του MashQL add-on σε μορφή βοήθηματος. Εάν ενδιαφέρεστε να κατανοήσετε καλύτερα το add-on μπορείτε να συμβουλευτείτε το βοήθημα.

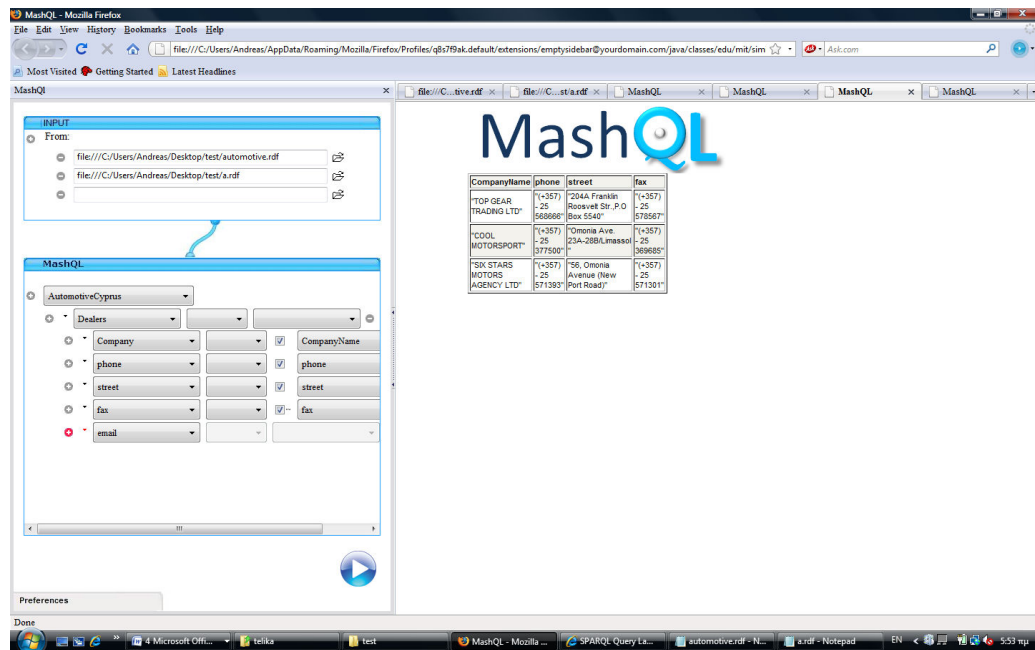
The screenshot shows the MashQL application running in a Mozilla Firefox browser. The interface is divided into several sections:

- INPUT:** A section for loading RDF files. It shows two file paths: `file:///C:/Users/Andreas/Desktop/test/automotive.rdf` and `file:///C:/Users/Andreas/Desktop/test/a.rdf`.
- MashQL:** A section for defining the query. It includes a dropdown menu for 'AutomotiveCyprus' and a table of properties (Company, phone, street, fax, email) with checkboxes for selection. The 'Company' property is selected, and the 'Dealers' dropdown is set to 'Company'.
- Results:** A section displaying the query results in a table format. The table has columns: CompanyName, phone, street, fax. The data includes rows for COOL MOTORSPORT, TOP GEAR TRADING LTD, and SIX STARS MOTORS AGENCY LTD.

The query results table is as follows:

CompanyName	phone	street	fax
COOL MOTORSPORT	(+357) 25 377500	Omonia Ave. 25A-208Lmassos	(+357) 369685
TOP GEAR TRADING LTD	(+357) 25 568666	204A Franklin Roosevelt Str. P.O. Box 5547	(+357) 578567
SIX STARS MOTORS AGENCY LTD	(+357) 25 871383	56, Omonia Avenue (New Port Road)	(+357) 871381

παράδειγμα 4.25



παράδειγμα 4.26

Κεφάλαιο 5. MashQL usability test.

5.1. Εισαγωγή.	53
5.2. Περίληψη	54
5.3. Μεθοδολογία	54
5.4. Αποτελέσματα	56

5.1. Εισαγωγή.

Η ιστοσελίδα <http://www.cyprusyellowpages.com/> περιλαμβάνει απαραίτητες πληροφορίες για εταιρίες πώλησης (dealers.rdf) και επιδιόρθωσης (services.rdf) αυτοκινήτων. Ζητείται από τους χρήστες να χρησιμοποιήσουν την ιστοσελίδα για να αναζητήσουν πληροφορίες για κάποιες από τις εταιρίες. Οι μέθοδοι για την ανάκτηση των πληροφοριών που αναζητούν, είναι καταρχάς η κλασική μέθοδος πλοήγησης μιας ιστοσελίδας και η αναζήτηση δεδομένων με την χρήση του MashQL add-on.

Τα στοιχεία που υπάρχουν στη ιστοσελίδα <http://www.cyprusyellowpages.com/> βρίσκονται σε δομημένη μορφή στο αρχείο dealers.rdf και στο αρχείο services.rdf. Χρησιμοποιώντας αυτά τα αρχεία θα μπορείτε να κάνετε την αναζήτηση μέσω του MashQL add-on. Το ζητούμενο είναι να αξιολογήσετε το MashQL add-on και να το συγκρίνετε με την κλασική μέθοδο πλοήγησης ως προς τη δική σας ευκολία. Η ιστοσελίδα ,όσο και τα αρχεία περιλαμβάνουν τα πιο κάτω δεδομένα: Company, street, city, phone, fax, Contact Person, email, Website και το logo της εταιρίας. Τα αρχεία dealers.rdf και services.rdf είναι αναρτημένα στο παράρτημα Α

5.2. Περίληψη

Στο MashQL Firefox add on, διενεργήθηκε το τεστ χρησιμότητας μεταξύ της 1^{ης} Δεκεμβρίου και 11^{ης} Δεκεμβρίου 2009. Σκοπός του τεστ ήταν η σύγκριση του MashQL firefox add on με την κλασσική μέθοδο πλοήγησης, όσον αφορά τη ταχύτητα, ευκολία και την αξιοπιστία ανεύρεσης δεδομένων.

Στο τεστ έλαβαν μέρος 15 άτομα. Το τεστ περιλάμβανε συνολικά 7 σενάρια. Το κάθε σενάριο διήρκεσε περίπου 5 λεπτά.

Γενικά όλοι οι συμμετέχοντες έμειναν ικανοποιημένοι από το MashQL add on και έδωσαν ενθαρρυντικά στοιχεία. Το 87% το βρήκαν εύκολο στην χρήση.

Το τεστ διάγνωσε μόνο μερικά μικροπροβλήματα:

- Στην αρχή του τεστ οι συμμετέχοντες καθυστερούσαν περισσότερο γιατί δεν γνώριζαν καλά το σύστημα (πρώτη χρήση).
- Η δημιουργία Union ήταν λίγο δυσνόητη στους χρήστες.
- Επίσης υπήρξε και μια δυσκολία κατανόησης του user interface από κάποιους χρήστες αλλά δεν καταγράφεται ως πρόβλημα (λόγω του ότι ήταν μεμονωμένο και λόγω του τυχαίου δείγματος) απλά η αναφορά του είναι αναγκαία.

5.3. Μεθοδολογία

Οι συμμετέχοντες που πήραν μέρος στο τεστ επιλέχθηκαν τυχαία από τον διαχειριστή της έρευνας. Η επιλογή έγινε ανεξαρτήτου φύλου, επαγγέλματος, σπουδής ή ηλικίας(που έτυχε να ήταν 20-30 ετών), και είχε μοναδική προϋπόθεση την ικανοποιητική γνώση Η/Υ και χρήση διαδικτύου. Το κάθε τεστ έγινε ατομικά ούτως ώστε να μην έχουν επαφή μεταξύ τους οι συμμετέχοντες. Ο ερευνητής αφού έδινε το

ερωτηματολόγιο στους συμμετέχοντες(appendix a), τους εξηγούσε τη διαδικασία που θα ακολουθούσε, το σύστημα που θα γινόταν χρήση του και ποιος είναι ο σκοπός του συστήματος. Αφού διάβαζαν το εισαγωγικό μέρος, τους ζητούσε να προχωρήσουν στην επόμενη σελίδα όπου υπήρχε ένα ερωτηματολόγιο σχετικό με προσωπικές πληροφορίες και χρήση τεχνολογίας.

Ακολούθως τους παρουσίαζε τα σενάρια όπως γράφτηκαν και τους ζητούσε να τα εκτελέσουν όπως αυτοί νόμιζαν (επίσης τους τόνισε ότι δεν υπήρχε σωστό ή λάθος και ότι δεν εξετάζονταν αυτοί αλλά η ευχρηστία του συστήματος, οπότε να αισθάνονται άνετα). Καθώς τα εκτελούσαν γίνονταν μετρήσεις χρόνων και παρατηρήσεις των σχολίων τους. Επίσης τους ζητήθηκε εάν μπορούσαν να εκφράζουν και με λόγια αυτά που σκέφτονταν, βασικά να σκέφτονται μεγαλόφωνα κατά τη διάρκεια της όλης διαδικασίας.

Μετά την ολοκλήρωση του κάθε σεναρίου τους ζητούσε να συμπληρώσουν ένα ερωτηματολόγιο (after scenario questionnaire). Με λίγα λόγια υπήρχαν 7 σενάρια άρα και 7 after scenarios. Το συγκεκριμένο ερωτηματολόγιο που χρησιμοποιήθηκε έχει δημοσιευθεί στο πιο κάτω άρθρο: Lewis, J. R. (1995) *IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. International Journal of Human-Computer Interaction*, 7:1, 57-78.

Το after scenario[16] βαθμολογούσε 3 βασικά σημεία(από 1/agree – 7/disagree):

1. Γενικά, είμαι ευχαριστημένος με την ευκολία της συμπλήρωσης της εργασίας σε αυτό το σενάριο.
2. Γενικά, είμαι ευχαριστημένος με το ποσοστό της ώρας που χρειάστηκε για να γίνει η εργασία σε αυτό το σενάριο.
3. Γενικά, είμαι ευχαριστημένος με τις πληροφορίες υποστήριξης όταν συμπληρώθηκε η εργασία.

Στο τέλος του τεστ ζητήθηκε από τους συμμετέχοντες να συμπληρώσουν ακόμη ένα ερωτηματολόγιο όσον αφορά την όλη διαδικασία(System Usability Scale questionnaire) το οποίο έχει δημοσιευθεί στο πιο κάτω άρθρο: SUS - A quick and dirty usability scale.

Το SUS[15] αξιολογούσε 10 σημεία(από 1/strongly disagree- 5/strongly agree) μερικά από αυτά είναι:

1. Νομίζω πως θα ήθελα να χρησιμοποιώ αυτό το σύστημα πιο συχνά.
2. Βρίσκω το σύστημα αχρείαστα πολύπλοκο.
3. Νομίζω πως το σύστημα ήταν εύκολο στη χρήση.
4. Νομίζω πως θα χρειαζόμουν την υποστήριξη ενός τεχνικού για να μπορέσω να χρησιμοποιήσω αυτό το σύστημα.
5. Βρήκα πως οι διάφορες λειτουργίες στο σύστημα ήταν καλά ολοκληρωμένες.
6. Νομίζω πως υπήρχε πολλή ασυνέπεια στο σύστημα.
7. Θα μπορούσα να φανταστώ ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν το σύστημα εύκολα.
8. Βρήκα το σύστημα πολύ δυσκίνητο για χρήση.
9. Ένοιωσα πολύ βέβαιος κάνοντας χρήση του συστήματος.
10. Χρειάστηκε να κατανοήσω πολλά πράγματα πριν να μπορέσω να αρχίσω να χρησιμοποιώ το σύστημα.

Όλα τα ερωτηματολόγια και τα σενάρια βρίσκονται αναρτημένα στο Apprentix A, στην μορφή που δόθηκαν στους συμμετέχοντες.

Όλοι οι συμμετέχοντες ήταν τυχαίας επιλογής, με μοναδική προϋπόθεση την ικανοποιητική γνώση του ηλεκτρονικού υπολογιστή και του διαδικτύου. Δέκα συμμετέχοντες έκαναν το τεστ ανάμεσα της 1^{ης} Δεκεμβρίου και της 11^{ης} Δεκεμβρίου του 2009. Και οι δέκα ολοκλήρωσαν το τεστ.

5.4. Αποτελέσματα

Στο υποκεφάλαιο που ακολουθεί θα επεξηγηθούν τα αποτελέσματα των ερωτηματολογίων που δόθηκαν στους χρήστες. Ξεκινώντας την ανάλυση των

αποτελεσμάτων πρέπει να αναφερθεί πως όλοι οι χρήστες κατάφεραν να υλοποιήσουν και τα εφτά σενάρια.

Στο παράδειγμα 5.1 παρουσιάζεται ο χρόνος που χρειάστηκε ο κάθε χρήστης να υλοποιήσει το κάθε σενάριο με τη χρήση MashQL add-on. Στην τελευταία στήλη παρουσιάζεται ο μέσος όρος του χρόνου για όλους τους χρήστες ανά σενάριο. Στο παράδειγμα 5.2 παρουσιάζεται ο αντίστοιχος πίνακας για την υλοποίηση των ιδίων σεναρίων χρησιμοποιώντας τη κλασική μέθοδο πλοήγησης σε μια ιστοσελίδα. Σημαντική παρατήρηση είναι ότι οι άπειροι χρήστες του MashQL add-on αλλά εξίσου έμπειροι χρήστες του φυλλομετρητή χρειάζονται έως και τέσσερις φορές περισσότερο χρόνο να εντοπίσουν τα δεδομένα με τη χρήση της κλασσικής πλοήγησης σε σχέση με την πλοήγηση που προσφέρει το MashQL add-on.

	<i>T1</i>	<i>T2</i>	<i>T3</i>	<i>T4</i>	<i>T5</i>	<i>T6</i>	<i>T7</i>
User1	35	24	30	40	30	35	35
User2	65	50	43	37	43	76	47
User3	55	22	35	33	29	50	56
User4	40	33	24	26	25	38	30
User5	25	62	37	42	63	43	19
User6	83	57	37	46	37	41	46
User7	90	40	45	32	30	53	40
User8	48	27	40	49	25	37	49
User9	125	65	59	63	57	43	47
User10	40	33	43	39	47	90	43
	60,6	41,3	39,3	40,7	38,6	50,6	41,2

παράδειγμα 5.1

	<i>T1</i>	<i>T2</i>	<i>T3</i>	<i>T4</i>	<i>T5</i>	<i>T6</i>	<i>T7</i>
User1	130	135	180	160	230	300	20
User2	200	200	210	160	203	295	10
User3	180	170	170	200	195	410	10
User4	150	148	160	180	210	330	15
User5	93	310	197	190	180	200	27
User6	153	170	275	210	275	300	17
User7	200	190	150	175	190	270	5
User8	175	160	190	170	230	260	15
User9	140	200	263	193	205	310	20
User10	137	210	230	163	195	280	17
	155,8	189,3	202,5	180,1	211,3	295,5	15,6

παράδειγμα 5.2.

Στους πίνακες που ακολουθούν παρουσιάζονται τα αποτελέσματα από τα after scenario. Στη τελευταία στήλη κάθε πίνακα παρουσιάζεται το ποσοστό επιτυχίας. Για τον υπολογισμό της επιτυχίας για το συγκεκριμένο ερωτηματολόγιο πρέπει να υπολογιστεί το βάρος της κάθε απάντησης που είναι βάση της κλιμάκωσης των επιλογών, συγκεκριμένα από το 1 έως 7 και τον αριθμό των ερωτήσεων δηλαδή τρεις. Το βάρος είναι ίσο με $100/6*3=5.5$. Στη συνέχεια πρέπει να αντιστοιχίσει το 1 με τη τιμή 6, το 2 με τη τιμή 5 και ούτω καθεξής. Για παράδειγμα εάν επιλέγηκαν από το χρήστη για το πρώτο ερώτημα η τιμή 2, για το δεύτερο η τιμή 3 και για το τρίτο η τιμή 1 τότε το άθροισμα θα είναι $5+4+6=15$. Ο υπολογισμός του ποσοστού επιτυχίας δίνεται από το

γινόμενο του αθροίσματος με το βάρος($15 \cdot 5.5 = 82.5\%$). Για τον υπολογισμό του γενικού ποσοστού χρησιμοποιείται ο μέσος όρος των ποσοστών των δέκα ατόμων.

Το παράδειγμα 5.3 παρουσιάζει τα ποσοστά επιτυχίας για το πρώτο σενάριο. Το πρώτο σενάριο προτρέπει τους χρήστες να εκτελέσουν μια γενική αναζήτηση στη ιστοσελίδα αναζητώντας τα ονόματα των εταιριών πώλησης αυτοκινήτων. Σύμφωνα με τα αποτελέσματα των παραδειγμάτων 5.1, 5.2 και 5.3 διαπιστώνεται πως η χρήση του MashQL add-on όχι μόνο αφήνει ικανοποιημένους τους χρήστες αλλά βελτιώνει και το χρόνο που χρειάζονται για να αναζητήσουν δεδομένα σε μια ιστοσελίδα.

	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	%
User1	2	2	1	88
User2	2	2	1	88
User3	1	1	1	100
User4	1	1	2	93,5
User5	1	1	2	93,5
User6	3	1	1	88
User7	1	2	2	88
User8	1	1	1	100
User9	1	2	2	88
User10	1	1	1	100
			Total	92,7
			%	

παράδειγμα 5.3.

Στο παράδειγμα 5.4 παρουσιάζει τα ποσοστά επιτυχίας για το δεύτερο σενάριο. Το δεύτερο σενάριο προτρέπει τους χρήστες να εκτελέσουν μια γενική αναζήτηση στη ιστοσελίδα αναζητώντας όνομα, τηλέφωνο επικοινωνίας και το fax της εταιρίας. Σύμφωνα με τα αποτελέσματα των παραδειγμάτων 5.1,5.2 και 5.4 διαπιστώνεται πως η χρήση του MashQL add-on και σε αυτό το σενάριο αφήνει μόνο ικανοποιημένους τους χρήστες αλλά βελτιώνει και το χρόνο που χρειάζονται για να αναζητήσουν ένα σύνολο δεδομένων σε μια ιστοσελίδα.

	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	%
User1	2	1	1	93,5
User2	2	1	1	93,5
User3	2	1	1	93,5
User4	1	1	1	100
User5	1	1	1	100
User6	2	1	1	93,5
User7	3	2	1	82,5
User8	1	1	1	100
User9	1	1	3	88
User10	1	1	1	100
			total	94,45

παράδειγμα 5.4.

Το παράδειγμα 5.5 παρουσιάζει τα ποσοστά επιτυχίας για το τρίτο σενάριο. Το τρίτο σενάριο προτρέπει τους χρήστες να εκτελέσουν μια γενική αναζήτηση στη ιστοσελίδα αναζητώντας Company ,street ,city ,phone , fax , Contact Person, email και Website της εταιρείας. Η διαφορά αυτού του σεναρίου με τα υπόλοιπα είναι πως οι εταιρείες δεν περιέχουν όλα τα δεδομένα που αναζητά ο χρήστης. Το γεγονός αυτό δυσκολεύει τους χρήστες στη κλασική αναζήτηση. Σύμφωνα με τα αποτελέσματα των παραδειγμάτων 5.1,5.2 και 5.5 διαπιστώνεται πως η χρήση του MashQL add-on όχι μόνο δεν τους δυσκολεύει και αφήνει ικανοποιημένους τους χρήστες αλλά βελτιώνει και το χρόνο που χρειάζονται για να αναζητήσουν ένα σύνολο δεδομένων σε μια ιστοσελίδα.

	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	
User1	3	1	2	82,5
User2	1	1	2	93,5
User3	2	1	2	88
User4	1	1	1	100
User5	1	1	2	93,5
User6	2	1	2	88
User7	2	2	1	88
User8	2	2	1	88
User9	1	2	2	88
User10	1	1	1	100
				90,95

παράδειγμα 5.5.

Στο τέταρτο σενάριο δυσκολεύουν τα πράγματα για το χρήστη. Αυτό το σενάριο προτρέπει τους χρήστες να εκτελέσουν μια ειδικευμένη αναζήτηση στη ιστοσελίδα και να εντοπίσουν τις εταιρείες που βρίσκονται στην επαρχία Λεμεσού. Η δυσκολία που αντιμετωπίζουν οι χρήστες είναι ότι οι εταιρείες από τη συγκεκριμένη επαρχία είναι διάσπαρτες σε όλη τη σελίδα. Τα αποτελέσματα των παραδειγμάτων 5.1,5.2 και 5.6 που παρουσιάζουν τα ποσοστά επιτυχίας για το τέταρτο σενάριο, αποδεικνύουν για ακόμη μια φορά πως η χρήση του MashQL add-on όχι μόνο δεν δυσκολεύει το χρήστη αλλά τον βοηθά σε μια γρήγορη και εύκολη αναζήτηση.

	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	
User1	2	1	2	88
User2	2	1	3	82,5
User3	2	1	1	93,5
User4	1	1	1	100
User5	1	1	2	93,5
User6	1	1	1	100
User7	2	2	2	82,5
User8	2	3	3	71,5
User9	1	1	2	93,5
User10	1	1	1	100
			total	90,5

παράδειγμα 5.6.

Όπως και στο τέταρτο σενάριο και στο πέμπτο σενάριο οι χρήστες πρέπει να εκτελέσουν μια εξειδικευμένη αναζήτηση στη ιστοσελίδα και να εντοπίσουν τις εταιρείες που δεν διαθέτουν ιστοσελίδα . Η δυσκολία που αντιμετωπίζουν οι χρήστες είναι ότι οι εταιρείες που δεν διαθέτουν ιστοσελίδα είναι διάσπαρτες σε όλη τη ιστοσελίδα. Τα αποτελέσματα των παραδειγμάτων 5.1,5.2 και 5.7 που παρουσιάζουν τα ποσοστά επιτυχίας για το πέμπτο σενάριο, αποδεικνύουν για ακόμη μια φορά πως η χρήση του MashQL add-on όχι μόνο δεν δυσκολεύει το χρήστη αλλά τον βοηθά σε μια γρήγορη και εύκολη αναζήτηση.

	$Q1$	$Q2$	$Q3$	
User1	2	1	2	88
User2	1	1	1	100
User3	2	1	2	88
User4	1	1	1	100
User5	1	2	3	82,5
User6	2	1	2	88
User7	1	2	1	93,5
User8	1	1	3	88
User9	1	1	2	93,5
User10	1	1	1	100
			total	92,15

παράδειγμα 5.7.

Στο έκτο σενάριο δυσκολεύουν ακόμη περισσότερο τα πράγματα για το χρήστη. Εκτός από το ότι προτρέπει τους χρήστες να εντοπίσουν τις εταιρείες που δεν διαθέτουν ιστοσελίδα, επιπλέον τους ζητείται η αναζήτηση να είναι από δυο διαφορετικές ιστοσελίδες. Τα αποτελέσματα των παραδειγμάτων 5.1,5.2 και 5.8 που παρουσιάζουν τα ποσοστά επιτυχίας για το έκτο σενάριο, αποδεικνύουν για ακόμη μια φορά πως η χρήση του MashQL add-on όχι μόνο δεν δυσκολεύει το χρήστη αλλά τον βοηθά σε μια γρήγορη και εύκολη αναζήτηση.

	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>%</i>
User1	2	1	2	88
User2	2	2	1	88
User3	1	1	2	93,5
User4	1	1	2	93,5
User5	1	2	2	88
User6	2	1	2	88
User7	3	2	1	82,5
User8	1	1	3	88
User9	1	1	2	93,5
User10	1	1	1	100
			total	90,3

παράδειγμα 5.8.

Στους πίνακες που ακολουθούν παρουσιάζονται τα αποτελέσματα από τα SUS. Στην τελευταία στήλη κάθε πίνακα παρουσιάζεται το ποσοστό επιτυχίας. Για τον υπολογισμό της επιτυχίας για το συγκεκριμένο ερωτηματολόγιο πρέπει να υπολογιστεί το βάρος της κάθε απάντησης που είναι βάση της κλιμάκωσης των επιλογών, συγκεκριμένα από το 1 έως 5 και τον αριθμό των ερωτήσεων δηλαδή δέκα. Το βάρος είναι ίσο με $100/10*4=2.5$. Στη συνέχεια πρέπει να αντιστοιχίσει το 1 με τη τιμή 5, το 2 με τη τιμή 4 και ούτω καθεξής. Για τον υπολογισμό του γενικού ποσοστού χρησιμοποιείται ο μέσος όρος των ποσοστών των δέκα ατόμων.

	<i>Q1</i>	<i>Q2</i>	<i>Q3</i>	<i>Q4</i>	<i>Q5</i>	<i>Q6</i>	<i>Q7</i>	<i>Q8</i>	<i>Q9</i>	<i>Q10</i>	%
User1	4	4	4	4	4	4	4	4	4	4	100
User2	1	3	2	3	3	4	3	4	2	3	70
User3	3	4	4	4	3	4	2	4	3	3	85
User4	3	3	4	4	3	4	4	4	3	4	90
User5	4	4	3	4	3	4	4	4	4	4	95
User6	4	4	4	4	4	4	4	4	3	4	97,5
User7	2	2	2	4	3	4	3	4	2	3	72,5
User8	3	4	3	4	4	4	4	4	3	4	92,5
User9	3	3	3	4	3	4	4	4	2	4	85
User10	3	4	4	4	3	4	4	4	3	4	92,5
total											88

παράδειγμα 5.9.

Παρατηρώντας τα αποτελέσματα του ερωτηματολογίου μπορούμε να διαπιστώσουμε πως οι χρήστες θεωρούν το add-on εύκολο στη χρήση και κυρίως αποτελεσματικό. Το ερωτηματολόγιο αποτελείται τόσο από ερωτήσεις που αφορούν την απόδοση, ερωτήσεις που αφορούν την αξιοπιστία των αποτελεσμάτων και τέλος

ερωτήσεις που αφορούν την ευκολία του χρήστη να το χρησιμοποιήσει. Παρόλο που τα ερωτηματολόγια συμπληρώθηκαν από άπειρους χρήστες το γενικό ποσοστό των αποτελεσμάτων είναι 88%. Αρκετά ικανοποιητικό και αισιόδοξο για τη χρησιμότητα του add-on.

Κεφάλαιο 6: Συμπεράσματα.

6.1. Γενικά συμπεράσματα.	67
6.2. Μελλοντική Εργασία	68

6.1. Γενικά συμπεράσματα.

Όπως προαναφέραμε, σήμερα το διαδίκτυο αποτελεί τη μεγαλύτερη πηγή πληροφοριών. Μέσω του διαδικτύου ο κάθε άνθρωπος έχει τη δυνατότητα να εντοπίσει πληροφορίες για τα πάντα. Το πρόβλημα που υφίσταται για την αναζήτηση δεδομένων στο διαδίκτυο είναι ότι τα περισσότερα δεδομένα βρίσκονται σε μη δομημένη μορφή.

Το πρόβλημα κλιμακώνεται με το γεγονός ότι οι πληροφορίες επαναλαμβάνονται, με συνεπακόλουθο να ξοδεύονται πολλά χρήματα και να γίνεται το διαδίκτυο πολύπλοκο. Από την προσωπική μου μελέτη, διαπίστωσα πως η τεχνολογία των Mashups θα περιορίσει αρκετά αυτό το πρόβλημα.

Η νέα τεχνολογία της MashQL που προτάθηκε, αποτελεί μια καινούργια άποψη των Mashups. Η MashQL επιτρέπει την αναζήτηση σε μία πηγή δεδομένων ανεξάρτητα εάν είναι γνωστό το σχήμα ή η δομή της. Επίσης, οι χρήστες δεν χρειάζεται να διαθέτουν γνώσεις για RDF και SPARQL. Τα πιο πάνω χαρακτηριστικά την καθιστούν ιδιαίτερα ενδιαφέρουσα και ίσως μια μελλοντική μέθοδο για αναζήτηση δεδομένων στο διαδίκτυο.

Τα αποτελέσματα που πήραμε από τα usability tests δείχνουν έμπρακτα ότι ο κοινός χρήστης του διαδικτύου έχει τη δυνατότητα να τη χρησιμοποιήσει σωστά και εύκολα. Επιπλέον, από τα αποτελέσματα που πήραμε φάνηκε ότι, εκτός της ευκολίας στη χρήση που προσφέρει, μειώνει σημαντικά και τον χρόνο που χρειάζεται ο χρήστης για να εντοπίσει δεδομένα σε μια σελίδα σε σχέση με τον παραδοσιακό τρόπο αναζήτησης.

Μια ακόμα σημαντική λειτουργία της MashQL, είναι η δυνατότητα που δίνει στο χρήστη να αναζητά πληροφορίες από πολλαπλές πηγές χρησιμοποιώντας μόνο τη MashQL χωρίς να χρειάζεται να πλοηγείται σε αυτές.

Τέλος, λόγω του ότι η MashQL είναι υλοποιημένη σαν add-on στον Firefox, δίνει την ευελιξία στον χρήστη να έχει άμεση πρόσβαση στο εργαλείο βοηθώντας τον να εκτελεί γρηγορότερα τις αναζητήσεις του.

6.2. Μελλοντική Εργασία

Λόγω του ότι τα RDFa και microformat αρχεία είναι αρκετά διαδεδομένα στο διαδίκτυο και υποστηρίζονται από μεγάλο αριθμό ιστοσελίδων θα καθιστούσαν τη MashQL μια πολύ αξιόλογη μέθοδο αναζήτησης.

Σαν μελλοντική εργασία, σημαντικό θα ήταν η υποστήριξη από το MashQL add-on των πιο πάνω τύπων αρχείων. Αυτό θα καθιστούσε το MashQL add-on σαν ένα σημαντικό εργαλείο για όλους τους χρήστες του διαδικτύου.

Επιπλέον, για την καλύτερη διάδοση της MashQL στο ευρύτερο κοινό θα βοηθούσε ιδιαίτερα η υλοποίηση της και σε άλλους φυλλομετρητές.

Βιβλιογραφία.

1. **W3C.** SPARQL Query Language for RDF. [Ηλεκτρονικό] <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/>.
2. —. Resource Description Framework. [Ηλεκτρονικό] <http://www.w3.org/RDF/>.
3. —. RDF/XML Syntax Specification (Revised). [Ηλεκτρονικό] <http://www.w3.org/TR/REC-rdf-syntax/#basicconcepts>.
4. —. W3C Semantic Web Activity. [Ηλεκτρονικό] <http://www.w3.org/2001/sw/>.
5. *MashQL: A Query-by-Diagram Language -Towards Semantic Data Mashups.* **Dikaiakos, Mustafa Jarrar and Marios D.** s.l. : ACM, 2008. 9781605582559.
6. *A Data Mashup Language for the Data Web.* **Dikaiakos, Mustafa Jarrar and Marios D.** ISSN 1613-0073, s.l. : ACM, 2009.
7. *Services Mashups: The New Generation of Web Applications.* **D. Benslimane, S. Dustdar, A. Seth.** s.l. : IEEE.
8. **Mozilla.** Mozilla Developer Center. [Ηλεκτρονικό] <https://developer.mozilla.org/En>.
9. **McCarthy, Philip.** Search RDF data with SPARQL. *SPARQL and the Jena Toolkit open up the semantic Web.* [Ηλεκτρονικό] 2005. <http://www.ibm.com/developerworks/library/j-sparql/>.
10. —. Introduction to Jena. [Ηλεκτρονικό] 2004. <http://www.ibm.com/developerworks/xml/library/j-jena/>.
11. **McBride, Brian.** An Introduction to RDF and the Jena RDF API. [Ηλεκτρονικό] 2007. http://jena.sourceforge.net/tutorial/RDF_API/index.html.
12. **Dodds, Leigh.** Introducing SparQL: Querying the Semantic Web. [Ηλεκτρονικό] 2005. <http://www.xml.com/lpt/a/1628>.
13. **Davis, Ian.** An Introduction to RDF. [Ηλεκτρονικό] <http://research.talis.com/2005/rdf-intro/>.
14. **W3C.** SparQL protocol for RDF. [Ηλεκτρονικό] 2008. <http://www.w3.org/TR/rdf-sparql-protocol>.
15. *SUS - A quick and dirty usability scale.* **Brooke, John.** United Kingdom : s.n.
16. *After-Scenario Questionnaire.* **Lewis, J. R.** s.l. : International Journal of Human-Computer Interaction, 1995.

Παράρτημα Α.

Σενάριο και ερωτηματολόγιο.

Πλοήγηση σε ιστοσελίδα με τη βοήθεια του MashQL add-on.

Εισαγωγή:

Η ιστοσελίδα <http://www.cyprusyellowpages.com/> περιλαμβάνει απαραίτητες πληροφορίες για εταιρίες πώλησης(dealers.rdf) και επιδιόρθωσης(services.rdf) αυτοκινήτων. Εσύ σαν χρήστης της ιστοσελίδας αναζητάς πληροφορίες για κάποιες από τις εταιρίες . Οι μέθοδοι για την ανάκτηση των πληροφοριών που αναζητάς, είναι καταρχάς η κλασική μέθοδος πλοήγησης μιας ιστοσελίδας και η αναζήτηση δεδομένων με την χρήση του MashQL add-on.

Τα στοιχεία που υπάρχουν στην ιστοσελίδα <http://www.cyprusyellowpages.com/> βρίσκονται σε δομημένη μορφή στο αρχείο dealers.rdf και στο αρχείο services.rdf. Χρησιμοποιώντας αυτά τα αρχεία θα μπορείτε να κάνετε την αναζήτηση μέσω του MashQL add-on. Το ζητούμενο είναι να αξιολογήσετε το MashQL add-on και να το συγκρίνετε κλασική μέθοδο πλοήγηση ως προς τη δική σας ευκολία.

Η ιστοσελίδα όσο και τα αρχεία περιλαμβάνουν τα πιο κάτω δεδομένα Company ,street ,city ,phone , fax , Contact Person, email, Website και το logo της εταιρίας.

1. Name: _____
2. Type: ☐ Teacher ☐ Student ☐ General Public
3. Grade Level: ☐ High School ☐ Bachelor ☐ Master
4. Age Group: ☐ 18-24 ☐ 25-34 ☐ 35-44 ☐ 45-55 ☐ over 55
5. Gender: ☐ Female ☐ Male
6. Profession: _____
7. Years Using the Web: _____
8. How often do you use the internet?
☐ Daily ☐ Weekly ☐ Monthly ☐ Occasionally ☐ Never
9. Which Internet browser or browsers do you use?
☐ Internet Explorer ☐ Mozilla Firefox ☐ Opera ☐ Other _____
10. What do you usually do on the internet?
☐ surfing ☐ selling/buying ☐ email
☐ site development ☐ chat ☐ interactive games
☐ information gathering ☐ MP3 downloading ☐ Other _____
11. What do you like about each of the Web sites?

12. What do you dislike about each of the Web sites?

13. What are the last three Web accessibility topics or questions you researched on the Internet?

14. What type of computer do you use? ☐ PC ☐ Mac
15. How many hours a week do you use a computer?

At home: _____

At work: _____

16. What do you typically use your computer for?

☐ Games and pleasure

☐ Graphics Accounting/Finance

☐ Data storage (i.e., databases)

☐ Email or chat Internet

☐ Word Processing

☐ Other _____

Σενάριο:

*εκτελεστεί τα πιο κάτω βήματα και με τους δυο τρόπους και καταγράψετε τα σχόλια σας

1. Εργάζεστε σε μια διαφημιστική εταιρία και σας ανατέθηκε να καταγράψετε όλες τις εταιρίες πώλησης αυτοκινήτων που βρίσκονται στη ιστοσελίδα <http://www.cyprusyellowpages.com/> σε ένα αρχείο. *

Disagree(1-7)

A. Overall, I am satisfied with the ease of
completing the tasks in this scenario .

B. Overall, I am satisfied with the amount
of time it took to complete the tasks in this scenario.

C. Overall, I am satisfied with the support
information (online-line help, messages, documentation) when
completing the tasks

Time

2. Επιπλέον πρέπει να επικοινωνήσετε με τις εταιρίες για να τους προτείνετε συνεργασία. Θα χρειαστείτε να καταγράψετε σε ένα αρχείο σε τρεις στήλες τις εταιρίες που χαρακτηρίζονται από τα στοιχεία όνομα εταιρίας, τηλέφωνο επικοινωνίας και fax. Να δώσετε στη στήλη με το όνομα της εταιρίας την ονομασία Company, στη στήλη με το τηλέφωνο την ονομασία phone και στη στήλη με τον αριθμό του fax την ονομασία fax. *

Disagree(1-7)	
A. Overall, I am satisfied with the ease of completing the tasks in this scenario .	
B. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario.	
C. Overall, I am satisfied with the support information (online-line help, messages, documentation) when completing the tasks	
Time	

3. Στη προσπάθεια σας για να βρείτε περισσότερους υποψήφιους πελάτες χρειάζεται να πάρετε όλα τα στοιχεία που καταγράφονται στη ιστοσελίδα και να τα καταγράψετε σε ένα αρχείο. Τα δεδομένα που χρειάζεσαι είναι τα πιο κάτω: Company ,street ,city ,phone , fax , Contact Person, email και Website. Επειδή μερικές εταιρίες δεν καταγράφουν όλα τα δεδομένα τα πεδία street ,city ,phone , fax , Contact Person, email και Website δεν είναι απαραίτητα.*

Disagree(1-7)	
A. Overall, I am satisfied with the ease of completing the tasks in this scenario .	
B. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario.	
C. Overall, I am satisfied with the support information (online-line help, messages, documentation) when	

completing the tasks Time

4. Ο διευθυντής της διαφημιστικής σας ζητά να δημιουργήσετε εκ νέου μια ίδια λίστα με του ερωτήματος 3 που να περιέχει μόνο τις εταιρίες της επαρχίας Λεμεσού για να αποσταλεί στο υπεύθυνο για την επαρχία.*

	Disagree(1-7)
A. Overall, I am satisfied with the ease of completing the tasks in this scenario .	
B. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario.	
C. Overall, I am satisfied with the support information (online-line help, messages, documentation) when completing the tasks	
Time	

5. Επιπλέον θα θέλατε να ξεχωρίσετε τα άτομα που δεν έχουν ιστοσελίδα με απώτερο σκοπό να τους προτείνετε το πακέτο διαφήμισης μέσω ιστοσελίδας*.

	Disagree(1-7)
A. Overall, I am satisfied with the ease of completing the tasks in this scenario .	

B. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario. C. Overall, I am satisfied with the support information (online-line help, messages, documentation) when completing the tasks Time

6. Καθώς μελετάτε την ιστοσελίδα παρατηρείτε πως αρκετοί μηχανικοί δεν διαθέτουν ιστοσελίδα έτσι παίρνετε την πρωτοβουλία να δημιουργήσετε μια ενιαία λίστα που θα περιλαμβάνει τα στοιχεία και από τις δυο κατηγορίες για όσες εταιρίες δεν διαθέτουν ιστοσελίδα.*

	Disagree(1-7)
A. Overall, I am satisfied with the ease of completing the tasks in this scenario . B. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario. C. Overall, I am satisfied with the support information (online-line help, messages, documentation) when completing the tasks Time	

7. Τελειώνοντας απ' τη δουλειά πρέπει να επισκεφτείς το μηχανικό σου στη Λεμεσό, άλλα ξέχασες την οδό του και δεν έχεις μαζί σου τον αριθμό

τηλεφώνου του. Έτσι αποφασίζεις να το αναζητήσεις μέσα στην ιστοσελίδα ,
 άλλα γρήγορα διαπιστώνεις πως δεν γνωρίζεις τίποτα για την εταιρία εκτός από
 το λογότυπο της.*



	Disagree(1-7)
A. Overall, I am satisfied with the ease of completing the tasks in this scenario .	
B. Overall, I am satisfied with the amount of time it took to complete the tasks in this scenario.	
C. Overall, I am satisfied with the support information (online-line help, messages, documentation) when completing the tasks	
Time	

	Strongly disagree					Strongly agree
1. I think that I would like to use this system frequently						
	1	2	3	4	5	
2. I found the system unnecessarily complex						
	1	2	3	4	5	
3. I thought the system was easy to use						
	1	2	3	4	5	
4. I think that I would need the support of a technical person to be able to use this system						
	1	2	3	4	5	
5. I found the various functions in this system were well integrated						
	1	2	3	4	5	
6. I thought there was too much inconsistency in this system						
	1	2	3	4	5	
7. I would imagine that most people would learn to use this system very quickly						
	1	2	3	4	5	
8. I found the system very cumbersome to use						
	1	2	3	4	5	
9. I felt very confident using the system						
	1	2	3	4	5	
10. I needed to learn a lot of things before I could get going with this system						
	1	2	3	4	5	

RDF αρχείο για πωλητές αυτοκινήτων.

```
<rdf:RDF xmlns="http://xmlns.com/foaf/0.1/"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:ical="http://www.w3.org/2002/12/cal/ical#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:swc="http://data.semanticweb.org/ns/swc/ontology#"
  xmlns:swrc="http://swrc.ontoware.org/ontology#"
  xmlns:swrc_ext="http://www.cs.vu.nl/~mcaklein/onto/swrc_ext/2005/05#">

  <swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
    <ical:Dealers rdf:parseType="Resource">
      <ical:Company>A. PASTELLAS LTD</ical:Company>
      <ical:street>144 Ayiou Ilarionos St, CY 1720, Nicosia, Cyprus </ical:street>
      <ical:city>Nicosia</ical:city>
      <ical:phone>(+357) - 22431162</ical:phone>
      <ical:fax>(+357) - 22431998</ical:fax>
      <ical:ContactPerson>George Pastellas</ical:ContactPerson>
      <ical:email rdf:resource="info@pastellas.com"/>
      <ical:Website rdf:resource="http://www.pastellas.com"/>
      <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/pastellas.jpg"/>
    </ical:Dealers>
  </swc:AutomotiveCyprus>
```

```

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:Dealers rdf:parseType="Resource">
    <ical:Company>A.G. MAKRIS MOTORS</ical:Company>
    <ical:street>Gaggi and Akanthous Corner (Near Agias Phylas Round About),
Limassol - Cyprus </ical:street>
    <ical:city>Limassol</ical:city>
    <ical:phone>(+357) - 25 735319</ical:phone>
    <ical:fax>(+357) - 25735318</ical:fax>
    <ical:email rdf:resource="andrelia@cytanet.com.cy"/>
    <ical:Website rdf:resource="http://www.makrismotors.cyauto.com"/>
    <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/makrismotors.GIF"/>
  </ical:Dealers>
</swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:Dealers rdf:parseType="Resource">
    <ical:Company>CHARALAMBOS PILAKOUTAS</ical:Company>
    <ical:street>Meteora Str, Strovolos Industr. Area , P.O.Box21168 </ical:street>
    <ical:city>Nicosia</ical:city>
    <ical:phone>(+357) - 22487711</ical:phone>
    <ical:fax>(+357) - 22487722</ical:fax>
    <ical:email rdf:resource="BMWgroup@charpilakoutas.com.cy "/>
    <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/charpilakoutas.jpg"/>
  </ical:Dealers>

```

```

</swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:Dealers rdf:parseType="Resource">
    <ical:Company>COOL MOTORSPORT</ical:Company>
    <ical:street>Omonia Ave. 23A-28B/Limassol </ical:street>
    <ical:city>Limassol</ical:city>
    <ical:phone>(+357) - 25 377500</ical:phone>
    <ical:fax>(+357) - 25 369685</ical:fax>
    <ical:ContactPerson>Michalis</ical:ContactPerson>
    <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/dactari.jpg"/>
  </ical:Dealers>
</swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:Dealers rdf:parseType="Resource">
    <ical:Company>ELITE MOTORS</ical:Company>
    <ical:city>Limassol</ical:city>
    <ical:phone>(+357) - 25-561777</ical:phone>
    <ical:fax>(+357) - 25-560641</ical:fax>
    <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/elitemotor.jpg"/>
  </ical:Dealers>
</swc:AutomotiveCyprus>

```

```

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:Dealers rdf:parseType="Resource">
    <ical:Company>FLAMINGO MOTORS AGENCY LTD</ical:Company>
    <ical:street>109, Omonia Avenue, 3045, Limassol - Cyprus</ical:street>
    <ical:city>Limassol</ical:city>
    <ical:phone>(+357) - 25 566456</ical:phone>
    <ical:fax>(+357) - 25 572334 </ical:fax>
    <ical:ContactPerson>Alex Ioannou (Managing
Director)</ical:ContactPerson>
    <ical:email rdf:resource="flamingo@cytanet.com.cy"/>
    <ical:Website rdf:resource="http://www.flamingo.cyauto.com"/>
    <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/flamingomot.gif"/>
  </ical:Dealers>
</swc:AutomotiveCyprus>

```

```

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:Dealers rdf:parseType="Resource">
    <ical:Company>GEORGE NATHANAEL SCOOTER LINE
LTD</ical:Company>
    <ical:street>28 Agias Phylaxeos, 3025 - Limassol, Cyprus </ical:street>
    <ical:city>Limassol</ical:city>
    <ical:phone>(+357) - 25358245</ical:phone>
    <ical:fax>(+357) - 25350369</ical:fax>
    <ical:ContactPerson>Nathanael George (Kokos) and Nathanael
Chrysis</ical:ContactPerson>
    <ical:email rdf:resource="harleyd@cytanet.com.cy"/>

```

```

    <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/kymco.jpg"/>

    </ical:Dealers>

    </swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">

    <ical:Dealers rdf:parseType="Resource">

        <ical:Company>GEORGIADES MOTORS</ical:Company>

        <ical:street>Mellina's Mercouri 45, Kato Polemidia, 4156 Limassol,
Cyprus</ical:street>

        <ical:city>Limassol</ical:city>

        <ical:phone>(+357) - 25 392662</ical:phone>

        <ical:fax>(+357) - 25 397118</ical:fax>

        <ical:ContactPerson>Andreas Georgiades</ical:ContactPerson>

        <ical:email rdf:resource="ageorgiadesmotors@yahoo.co.uk"/>

        <ical:Website rdf:resource="http://www.ageorgiades.cyauto.com"/>

        <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/georgiades.jpg"/>

    </ical:Dealers>

    </swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">

    <ical:Dealers rdf:parseType="Resource">

        <ical:Company>JOHN MOTOR LTD</ical:Company>

        <ical:street>Omonias 36, 3052 Lemesos</ical:street>

        <ical:city>Limassol</ical:city>

        <ical:phone>(+357) - 25574089</ical:phone>

```

```

    <ical:fax>(+357) - 25374383</ical:fax>

    <ical:email rdf:resource="elenan@cytanet.com.cy"/>

    <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/johnmotor.jpg"/>

    </ical:Dealers>

</swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">

    <ical:Dealers rdf:parseType="Resource">

        <ical:Company>K.A. THEODOROU and SON MOTORS
LTD</ical:Company>

        <ical:street>67, Paphou and 1 Miltonos Str. 3051, Limassol, Cyprus
</ical:street>

        <ical:city>Limassol</ical:city>

        <ical:phone>(+357) - 25395343</ical:phone>

        <ical:fax>(+357) - 25396535</ical:fax>

        <ical:ContactPerson>Kyriakos A. Theodorou </ical:ContactPerson>

        <ical:email rdf:resource="kyrtheo@avacom.net"/>

        <ical:Website rdf:resource="http://www.katheodorou.cyauto.com"/>

        <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/katheodorou.gif"/>

    </ical:Dealers>

</swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">

    <ical:Dealers rdf:parseType="Resource">

        <ical:Company>MOTORAMA ENTERPRISES LTD</ical:Company>

```

```

<ical:street>106, Pericleous, Nicosia - Cyprus </ical:street>

<ical:city>Nicosia</ical:city>

<ical:phone>(+357) - 22441660</ical:phone>

<ical:fax>(+357) - 22441661</ical:fax>

<ical:ContactPerson>Michalis Andreou</ical:ContactPerson>

<ical:email rdf:resource="motorama@cytanet.com.cy"/>

<ical:Website rdf:resource="http://www.motorama.com.cy"/>

<ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/motorama.gif"/>

</ical:Dealers>

</swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">

<ical:Dealers rdf:parseType="Resource">

<ical:Company>MOTORAMA ENTERPRISES LTD</ical:Company>

<ical:street>4 Glifados Str., Strovolos, Nicosia - Cyprus</ical:street>

<ical:city>Nicosia</ical:city>

<ical:phone>(+357) - 22 494332</ical:phone>

<ical:fax>(+357) - 22 313030</ical:fax>

<ical:ContactPerson>Andonis Kyriacou</ical:ContactPerson>

<ical:email rdf:resource="motorama@cytanet.com.cy"/>

<ical:Website rdf:resource="http://www.motorama.com.cy"/>

<ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/motorama.gif"/>

</ical:Dealers>

</swc:AutomotiveCyprus>

```

```

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:Dealers rdf:parseType="Resource">
    <ical:Company>OMIROS LIVADIOTIS</ical:Company>
    <ical:street>Egeou 4, 3065 Limassol, Cyprus</ical:street>
    <ical:city>Limassol</ical:city>
    <ical:phone>(+357) - 25 562163</ical:phone>
    <ical:fax>(+357) - 25 578625</ical:fax>
    <ical:ContactPerson>Omiros Livadiotis</ical:ContactPerson>
    <ical:email rdf:resource="omirosl@cytanet.com.cy"/>
    <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/omirosscooter.jpg"/>
  </ical:Dealers>
</swc:AutomotiveCyprus>

```

```

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:Dealers rdf:parseType="Resource">
    <ical:Company>SAM SAHOTA MOTORS LTD </ical:Company>
    <ical:street>22 Paphos Road, Limassol, Cyprus </ical:street>
    <ical:city>Limassol</ical:city>
    <ical:phone>+357-25 560055</ical:phone>
    <ical:fax>+357-25 560202</ical:fax>
    <ical:ContactPerson>Sam Sahota</ical:ContactPerson>
    <ical:email rdf:resource="info@samsmotors.com"/>
    <ical:Website rdf:resource="http://www.samsmotors.com"/>

```

```

        <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/samsmotors.gif"/>

    </ical:Dealers>

</swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">

    <ical:Dealers rdf:parseType="Resource">

        <ical:Company>SIX STARS MOTORS AGENCY LTD</ical:Company>

        <ical:street>56, Omonia Avenue (New Port Road)</ical:street>

        <ical:city>Limassol</ical:city>

        <ical:phone>(+357) - 25 571393</ical:phone>

        <ical:fax>(+357) - 25 571301</ical:fax>

        <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/renault2.gif"/>

    </ical:Dealers>

</swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">

    <ical:Dealers rdf:parseType="Resource">

        <ical:Company>TIMARK MOTORS</ical:Company>

        <ical:street>Osiou Louka-Kriton Corner, 1st Monovolikos,
Limassol</ical:street>

        <ical:city>Limassol</ical:city>

        <ical:phone>(+357) - 25391384</ical:phone>

        <ical:fax>(+357) - 25399811</ical:fax>

        <ical:ContactPerson>Markos Markou</ical:ContactPerson>

```

```

    <ical:email rdf:resource="timark@spidernet.com.cy"/>

    <ical:Website rdf:resource="http://www.timarkmotors.com"/>

    <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/timark.gif"/>

    </ical:Dealers>

    </swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">

    <ical:Dealers rdf:parseType="Resource">

        <ical:Company>TOP GEAR TRADING LTD</ical:Company>

        <ical:street>204A Franklin Roosevelt Str.,P.O Box 5540</ical:street>

        <ical:city>Limassol</ical:city>

        <ical:phone>(+357) - 25 568666</ical:phone>

        <ical:fax>(+357) - 25 578567</ical:fax>

        <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/nopic.gif"/>

        </ical:Dealers>

        </swc:AutomotiveCyprus>

</rdf:RDF>

```

RDF αρχείο για μηχανικούς αυτοκινήτων.

```
<rdf:RDF xmlns="http://xmlns.com/foaf/0.1/"
xmlns:dc="http://purl.org/dc/elements/1.1/"
xmlns:foaf="http://xmlns.com/foaf/0.1/"
xmlns:ical="http://www.w3.org/2002/12/cal/ical#"
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:swc="http://data.semanticweb.org/ns/swc/ontology#"
xmlns:swrc="http://swrc.ontoware.org/ontology#"
xmlns:swrc_ext="http://www.cs.vu.nl/~mcaklein/onto/swrc_ext/2005/05#">

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:CarServices rdf:parseType="Resource">
    <ical:Company>A. SKOUROS</ical:Company>
    <ical:street>Ipsonas Industrial Area, Limassol, Cyprus</ical:street>
    <ical:city>Limassol</ical:city>
    <ical:phone>(+357) - 25 711422</ical:phone>
    <ical:fax>(+357) - 25 711710</ical:fax>
    <ical:ContactPerson>Andreas Skouros</ical:ContactPerson>
    <ical:email rdf:resource="info@askouros.com"/>
    <ical:Website rdf:resource="http://www.askouros.com"/>
    <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/askouros.jpg"/>
  </ical:CarServices>
</swc:AutomotiveCyprus>
```

```

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:CarServices rdf:parseType="Resource">
    <ical:Company>ANDREAS STAVROU GARAGE</ical:Company>
    <ical:street>Behind Nemitsa Factory, Limassol</ical:street>
    <ical:city>Limassol</ical:city>
    <ical:phone>(+357) - 25353081</ical:phone>
  </ical:CarServices>
</swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:CarServices rdf:parseType="Resource">
    <ical:Company>ANDROS MATSAKOU</ical:Company>
    <ical:street>A. Vakalopoulou 27 C,Pafos Av.(Opp. Laniti)</ical:street>
    <ical:city>Limassol</ical:city>
    <ical:phone>(+357) - 25711097</ical:phone>
    <ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/anmatsakou.GIF"/>
  </ical:CarServices>
</swc:AutomotiveCyprus>

<swc:AutomotiveCyprus rdf:about="http://www.cyprusyellowpages.com/">
  <ical:CarServices rdf:parseType="Resource">
    <ical:Company>ANTONIS and STAVROS</ical:Company>
    <ical:street>6, Kiouri Str., Limassol</ical:street>
    <ical:city>Limassol</ical:city>

```

```
<ical:phone>(+357) - 25576923</ical:phone>

<ical:fax>(+357) - 25564963</ical:fax>

<ical:email rdf:resource="antonis-stavros@cytanet.com.cy"/>

<ical:Website rdf:resource="http://www.antonis-stavros.com"/>

<ical:logo
rdf:resource="http://www.cyprusyellowpages.com/cyyp/logos/antonisandstavros.jpg"/>

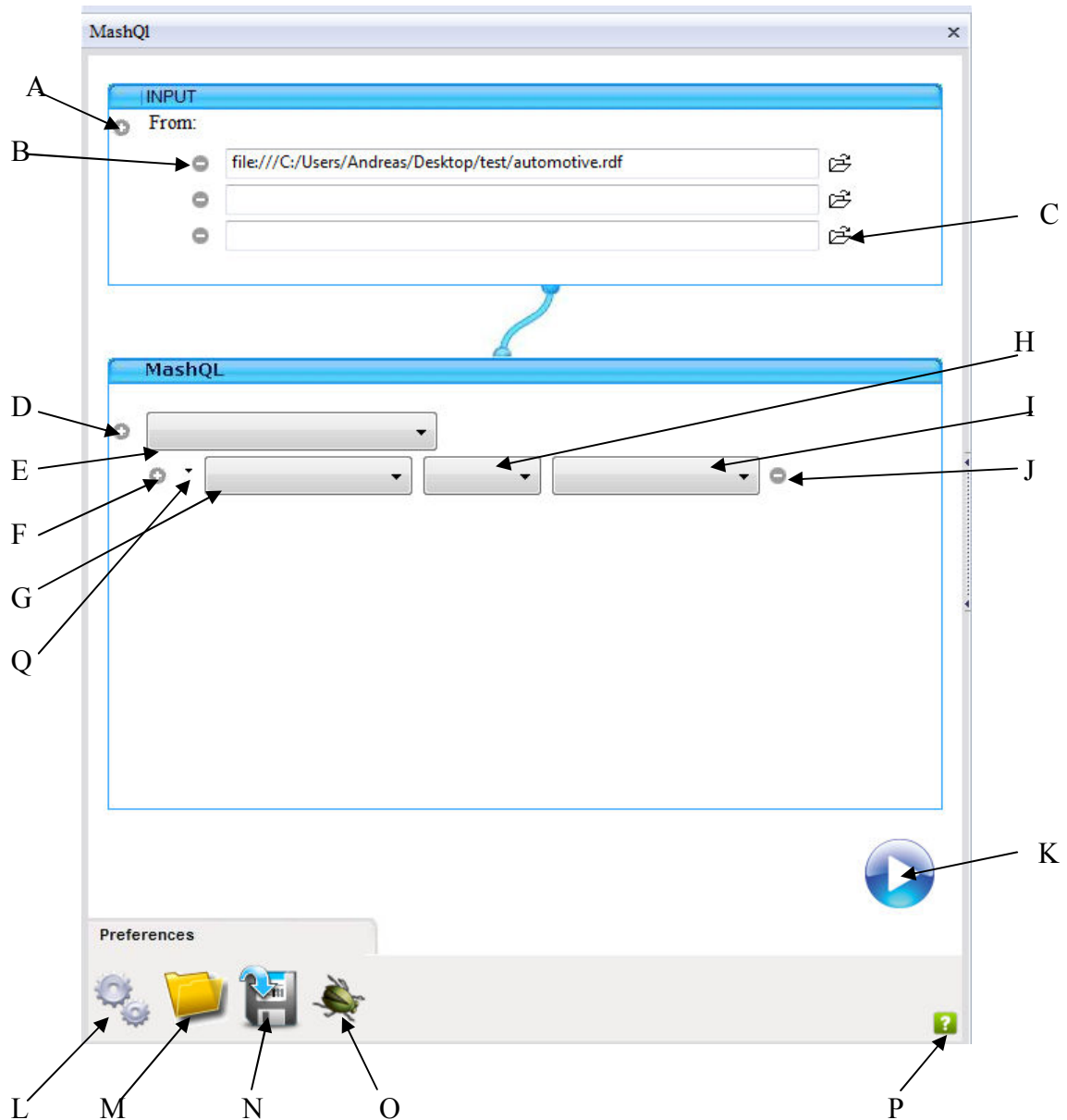
</ical:CarServices>

</swc:AutomotiveCyprus>

</rdf:RDF>
```

Παράρτημα Β.

Σχεδιάγραμμα Λειτουργιών



A= Εισαγωγή καινούργιου textbox για την εγγραφή του URL.

B= Διαγραφή του URL

C=Εντοπισμός RDF αρχείου στο προσωπικό σας υπολογιστή.

D=Εισαγωγή καινούργιου περιορισμού για το subject του πρώτου επιπέδου.

E=Compo box για επιλογή subject.

F= Εισαγωγή καινούργιου περιορισμού για το subject του δεύτερου επιπέδου.

G= Compo box για επιλογή predicate.

H= Compo box για επιλογή φίλτρου.

I= Compo box επιλογή object.

J= Διαγραφή του περιορισμού.

K= Εκτέλεση του ερωτήματος.

L= Εισαγωγή καινούργιου Prefix.

M= Φόρτωση αποθηκευμένου ερωτήματος.

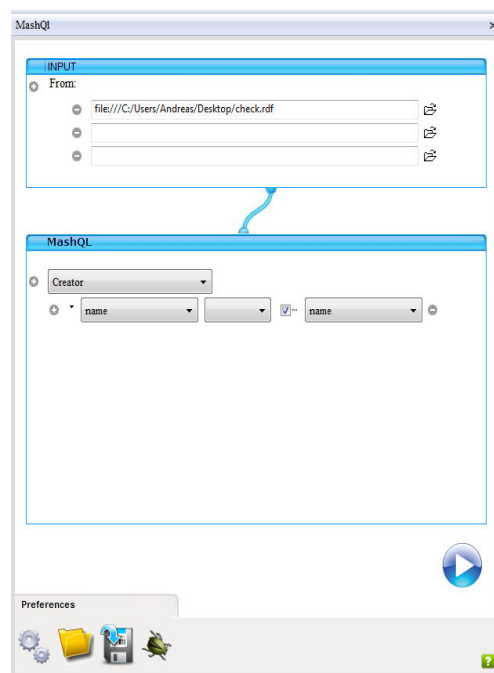
N= Αποθήκευση ερωτήματος.

O= Εκτέλεση του ερωτήματος και παρουσίαση του.

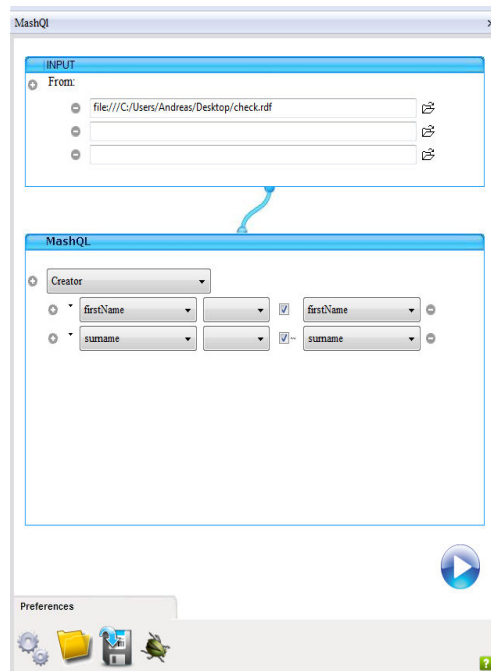
P= Βοήθεια.

Q= Επιλογή with, maybe και without.

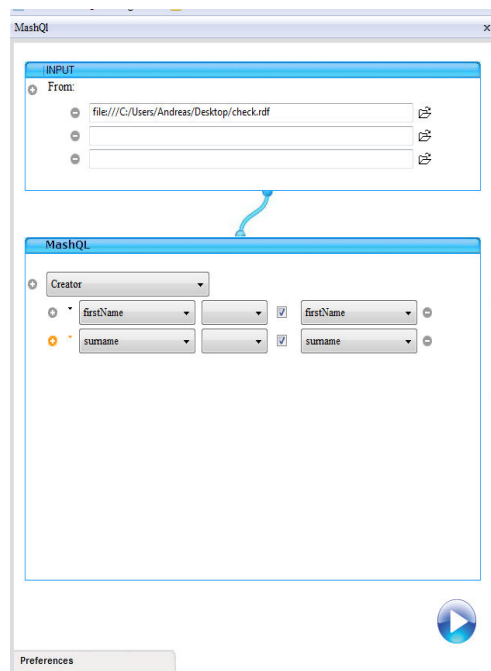
Εισαγωγή Ετικέτα



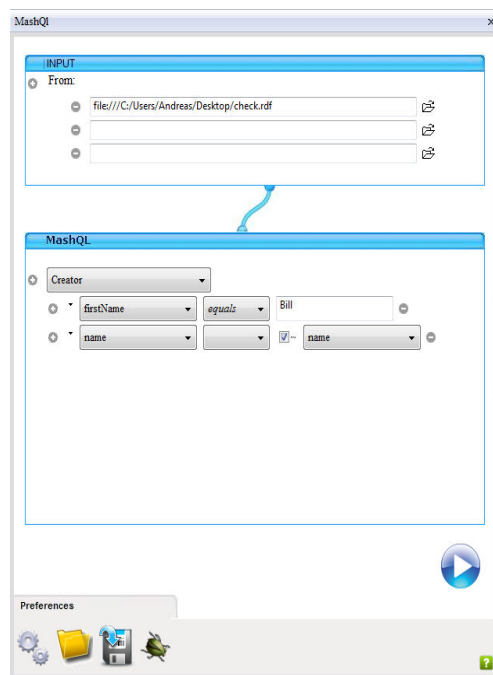
Δημιουργία ερωτήματος με τον Τελεστή And



Χρήση του χαρακτηριστικού Maybe



Χρήση Φίλτρου



Χρήση του χαρακτηριστικού Without

