

Το σύστημα RQL P2P : Ένα δίκτυο ομο- βάθμιων υπολογιστικών μονάδων βασισμέ- νο σε RDF, το οποίο κάνει χρήση RQL ερω- τήσεων και μιας υπηρεσίας ειδοποιήσεων

Νίκος Δημαρέσης

Εκπόνηση διπλωματικής εργασίας για την προϋπόθεση λήψη
του διπλώματος από το

Τμήμα Ηλεκτρονικών Μηχανικών Και Μηχανικών Υπολογι-
στών

Πολυτεχνείο Κρήτης

Εργαστήριο Προγραμματισμού Και Τεχνολογίας Ευφυών Υπο-
λογιστικών Συστημάτων

Περίληψη

Τα συστήματα ομοβάθμιων υπολογιστικών μονάδων (P2P) έχουν αναπτυχθεί εντυπωσιακά τα τελευταία χρόνια. Σε ένα P2P σύστημα, κατανεμημένοι κόμβοι ίσων δυνατοτήτων ανταλλάσσουν πληροφορία απευθείας μεταξύ τους, όπως έγγραφα, μουσικά αρχεία κ.τ.λ. Οι πόροι πληροφορίας στα P2P δίκτυα είναι αποθηκευμένοι στους πολυάριθμους κόμβους (peers) και κάθε κόμβος που ενδιαφέρεται για πόρους με κάποιο συγκεκριμένο περιεχόμενο, στέλνει την αντίστοιχη ερώτηση (query). Η εκτέλεση ερωτήσεων απαιτεί μεταδεδομένα που περιγράφουν τους πόρους που διαχειρίζονται οι κόμβοι. Σε πολλές περιπτώσεις, η χρήση σταθερών και αμετάβλητων μεταδεδομένων δεν μπορούν να περιγράψουν επαρκώς τους πόρους και οι ερωτήσεις δεν φέρνουν τα αναμενόμενα αποτελέσματα. Τα βασισμένα σε σχήμα (schema-based) P2P δίκτυα μας επιτρέπουν τελικά να λύσουμε αυτό το πρόβλημα, δίνοντας το δικαίωμα σε κάθε κόμβο να χρησιμοποιήσει και να ορίσει δικά του μεταδεδομένα.

Οι υπηρεσίες ειδοποίησης συμβάντος (event notification services) έχουν προκύψει τελευταία ως ένας πολλά υποσχόμενος τρόπος για δημιουργία συστημάτων που είναι βασισμένα στο διαδίκτυο. Σε μια τέτοια υπηρεσία, οι χρήστες στέλνουν το προφίλ τους για συμβάντα που τους ενδιαφέρουν, ώστε να λαμβάνουν ειδοποιήσεις ασύγχρονα σε τέτοια σχετιζόμενα συμβάντα.

Σε αυτή τη διπλωματική υλοποιήσαμε ένα βασισμένο σε σχήμα P2P δίκτυο, το οποίο χρησιμοποιεί την γλώσσα RDF για την αναπαράσταση των μεταδεδομένων των πόρων και τη γλώσσα RQL για την εκτέλεση ερωτήσεων πάνω στις RDF περιγραφές. Αυτό το P2P δίκτυο συνδυάζει τις ιδιότητες ενός κοινού P2P δικτύου και ενός συστήματος ειδοποίησης συμβάντων.

Περιεχόμενα

1	Εισαγωγή.....	6
1.1	Γενική Επισκόπηση.....	7
1.2	Οργάνωση Της Διπλωματικής.....	8
2	Θέματα Και Τεχνολογίες.....	9
2.1	Μεταδεδομένα	9
2.2	RDF.....	9
2.3	Συστήματα P2P.....	12
2.3.1	Καθαρά P2P Συστήματα.....	12
2.3.2	Υβριδικά P2P Συστήματα.....	13
2.3.3	Συστήματα Υπερ-κόμβων.....	13
2.3.4	Το Σύστημα P2P-DIET.....	14
2.4	Εργαλεία Του Σημασιολογικού Διαδικτύου.....	16
2.4.1	Το ICS-FORTH RDFSuite.....	17
2.4.2	Jena 2.....	23
2.4.3	Διαφορές Ανάμεσα Στα Δύο Εργαλεία.....	25
2.4	JDOM.....	26
2.5	Περίληψη.....	26
3	Το Σύστημα RQL P2P.....	27
3.1	Υβριδικό P2P Σύστημα.....	27
3.2	Δημοσίευση Πόρων.....	27
3.3	Ερωτήσεις.....	29
3.4	Προφίλ Χρηστών Και Ειδοποιήσεις.....	30
3.5	Αποθήκευση Πόρων.....	30
3.6	Ραντεβού.....	31
3.7	Διασύνδεση Γραφικών Με Το Χρήστη.....	31
3.8	Σενάριο Εφαρμογής.....	33
3.9	Περίληψη.....	34
4	Το Μηνύματα Επικοινωνίας.....	35
4.1	Τα Μηνύματα.....	35
4.2	Μηνύματα Επικοινωνίας Κόμβου – Διακομιστή.....	35
4.2.1	Μήνυμα Νέας Σύνδεσης.....	36

4.2.2	Μήνυμα Ελέγχου Ονόματος Χρήστη.....	36
4.2.3	Μήνυμα Αναζήτησης.....	36
4.2.4	Μήνυμα Δήλωσης Προφίλ.....	37
4.2.5	Μήνυμα Δημοσίευσης Πόρων.....	37
4.2.6	Μήνυμα Ορισμού Χώρου Ονομάτων.....	38
4.2.7	Μήνυμα Σύνδεσης.....	38
4.2.8	Μήνυμα Αίτησης Κατεβάσματος.....	38
4.2.9	Μήνυμα Αποδεκτού Ανεβάσματος.....	38
4.2.10	Μήνυμα Μη Διαθέσιμου Πόρου.....	39
4.2.11	Μήνυμα Ξεκινήματος Κατεβάσματος.....	39
4.2.12	Μήνυμα Αίτησης Ραντεβού.....	39
4.3	Μηνύματα Επικοινωνίας Διακομιστή – Κόμβου.....	39
4.3.1	Μήνυμα Αποδεκτής Σύνδεσης.....	39
4.3.2	Μήνυμα Μη Καταχωρημένου Ονόματος.....	40
4.3.3	Μήνυμα Καταχωρημένου Ονόματος.....	40
4.3.4	Μήνυμα Αποτελεσμάτων Ερώτησης.....	40
4.3.5	Μήνυμα Τέλους Αποτελεσμάτων Ερώτησης.....	41
4.3.6	Μήνυμα Ειδοποίησης.....	41
4.3.7	Μήνυμα Αίτησης Ανεβάσματος.....	41
4.3.8	Μήνυμα Αποσυνδεδεμένου Χρήστη.....	42
4.3.9	Μήνυμα Αποδεκτού Κατεβάσματος.....	42
4.3.10	Μήνυμα Αποτυχημένου Κατεβάσματος.....	42
4.3.11	Μήνυμα Ξεκινήματος Κατεβάσματος.....	42
4.3.12	Μήνυμα Λάθους.....	43
4.3.13	Μήνυμα Απάντησης Ραντεβού.....	43
4.3.14	Μήνυμα Συνδεδεμένου Χρήστη.....	43
4.4	Περίληψη.....	44
5	Η Υλοποίηση Του Συστήματος RQL P2P.....	45
5.1	Ο Διακομιστής.....	45
5.1.1	Οι PostgreSQL Βάσεις Δεδομένων.....	46
5.1.2	Το Java API του RSSDB.....	46
5.1.3	Το Java API Του RQL Ερμηνευτή.....	46
5.1.4	Το Java API Του JDOM.....	47
5.1.5	Σχηματική Αναπαράσταση Του Λογισμικού.....	47
5.2	Το RDF DC Peer.....	48
5.3	Περίληψη.....	49
6	Συμπερασματικά Σχόλια.....	50
	Βιβλιογραφία.....	52

Ο Κατάλογος Με Τις Εικόνες

2.1	Μία τριάδα	10
2.2	Η αρχιτεκτονική του P2P-DIET.....	15
2.3	Παράδειγμα RDF σχημάτων και περιγραφών πόρων για μία πολιτιστική πύλη.....	20
3.1	Πίνακας αναζήτησης του RDF dc peer.....	32
3.2	Πίνακας Ειδοποιήσεων του RDF dc peer.....	33
5.1	Σχηματική αναπαράσταση του λογισμικού.....	47

Ο Κατάλογος Με Τους Πίνακες

3.1	Πληροφορία Σχήματος των Μεταδεδομένων.....	28
3.2	Πληροφορία Σχήματος των Μεταδεδομένων ενημερωμένη, μετά την προσθήκη αρχείου.....	29
4.1	Μηνύματα Επικοινωνίας Μεταξύ Κόμβων και Διακομιστή.....	37
4.2	Μηνύματα Επικοινωνίας Μεταξύ Διακομιστή και Κόμβων.....	40

Κεφάλαιο 1

Εισαγωγή

Τα συστήματα *Ομοβάθμιων Υπολογιστικών Μονάδων* (*Peer-to-peer* ή εν συντομία *P2P*) έχουν αναπτυχθεί εντυπωσιακά τα τελευταία χρόνια. Σε ένα P2P σύστημα, κατανεμημένοι κόμβοι ίσων δυνατοτήτων ανταλλάσσουν πληροφορία απευθείας μεταξύ τους. Αυτά τα συστήματα αναπαριστούν ένα απίστευτο πλούτο πληροφοριών επιτρέποντας στους χρήστες να ανταλλάσσουν έγγραφα (Freenet [5]) και μουσικά αρχεία (Kazaa [6]). Ένα σημείο κλειδί ενός P2P συστήματος είναι η εύρεση πόρων. Υπάρχουν πολλοί μηχανισμοί ψαξίματος σε ένα σύστημα P2P. Το σύστημα Gnutella [7] χρησιμοποιεί ένα μηχανισμό όπου οι κόμβοι δεν έχουν δείκτες σε άλλους και οι *ερωτήσεις* (*queries*) διαδίδονται από κόμβο σε κόμβο μέχρι να βρεθούν πόροι που ταιριάζουν. Αυτός ο μηχανισμός ψαξίματος λειτουργεί και εξαπλώνεται σε ολόκληρο το δίκτυο (ή σε ένα υποσύνολο αυτού) με σκοπό να βρεθεί ένα ταίριασμα της ερώτησης. Τα κεντροποιημένα συστήματα ψαξίματος χρησιμοποιούν ειδικευμένους κόμβους που διατηρούν ένα κατάλογο των πόρων που είναι διαθέσιμα στο P2P σύστημα. Για να βρεθεί ένας πόρος, ο χρήστης εκτελεί μία ερώτηση στον ειδικευμένο κόμβο για να αναγνωρίσει κόμβους που έχουν πόρους με το περιεχόμενο ενδιαφέροντος.

Ενώ στο περιβάλλον *διακομιστή/πελάτη* (*client/server*) του παγκόσμιου ιστού τα μεταδεδομένα είναι χρήσιμα και σημαντικά, για τα Peer-to-Peer (P2P) περιβάλλοντα η ύπαρξη των μεταδεδομένων είναι απαραίτητη. Οι πόροι πληροφορίας στα P2P δίκτυα είναι αποθηκευμένοι σε πολυάριθμους κόμβους (*peers*) που περιμένουν ερωτήσεις, εάν ξέρουμε τι θέλουμε να ανακτήσουμε και ποιος κόμβος είναι ικανός να παρέχει αυτή την πληροφορία. Η εκτέλεση ερωτήσεων απαιτεί μεταδεδομένα που περιγράφουν τους πόρους που διαχειρίζονται οι κόμβοι, κάτι που είναι εύκολο να παραχθεί για εύκολες περιπτώσεις, αλλά δεν μας καλύπτει στις γενικές περιπτώσεις.

Οι P2P εφαρμογές είναι ιδιαίτερα δημοφιλείς για περιπτώσεις όπως η ανταλλαγή μουσικών αρχείων. Σε αυτές τις εφαρμογές, χρησιμοποιούνται σταθερά μεταδεδομένα για να περιγράψουν τους πόρους και οι ερωτήσεις που εκτελούνται δεν είναι ιδιαίτερα πολύπλοκες. Φυσικά για την ανάκτηση αυτών των πόρων, αλλά και άλλων, όπως είναι οι εκπαιδευτικοί πόροι που χτίζονται πάνω σε πρότυπα μεταδεδομένων όπως τα Dublin Core [30] και IEEE-LOM [8]/IMS [9], οι ερωτήσεις αυτές δεν είναι καθόλου επαρκείς.

Τα βασισμένα σε σχήμα (*schema-based*) P2P δίκτυα μας [39, 40] επιτρέπουν τελικά να λύσουμε τις ελλείψεις των P2P δικτύων που υποστηρίζουν μονάχα περιορισμένα και αμετάβλητα μεταδεδομένα, με την αποθήκευση κατανεμημένων δεδομένων και μεταδεδομένων. Αυτό γίνεται εφικτό με την υποστήριξη κόμβων που χρησιμοποιούν

σχήματα (πιθανώς ετερογενή) που περιγράφουν το περιεχόμενό τους και με την υλοποίηση στρατηγικών δρομολόγησης πάνω σε αυτή τη γνώση. Το Edutella project [10, 35] αντιμετωπίζει αυτές τις ελλείψεις των τωρινών P2P εφαρμογών στηριζόμενο πάνω στην χρήση της RDF.

Οι υπηρεσίες ειδοποίησης συμβάντος έχουν προκύψει τελευταία ως ένας πολλά υποσχόμενος τρόπος για δημιουργία συστημάτων που είναι βασισμένα στο διαδίκτυο. Σε μια τέτοια υπηρεσία, οι χρήστες στέλνουν το προφίλ τους για συμβάντα που τους ενδιαφέρουν, ώστε να λαμβάνουν ειδοποιήσεις ασύγχρονα για τέτοια σχετιζόμενα συμβάντα. Παραδείγματα συστημάτων ειδοποίησης συμβάντων είναι ολοκληρωμένα περιβάλλοντα ανάπτυξης, διασυνδέσεις γραφικών με το χρήστη, συστήματα διαχείρισης δικτύου κ.τ.λ. Εάν παράδειγμα συστήματος με δυνατότητες προφίλ/ειδοποίησης είναι το P2P-DIET [35, 36, 37, 38] και το SIENA [31].

Σε αυτή τη διπλωματική υλοποιήσαμε ένα βασισμένο σε σχήμα P2P δίκτυο, το οποίο χρησιμοποιεί την γλώσσα RDF [1, 2] για την αναπαράσταση των μεταδεδομένων των πόρων και τη γλώσσα RQL [17, 18] για την εκτέλεση ερωτήσεων πάνω στις RDF περιγραφές. Αυτό το P2P δίκτυο συνδυάζει τις ιδιότητες ενός κοινού P2P δικτύου και ενός συστήματος ειδοποίησης συμβάντων.

1.1 Γενική Επισκόπηση

Σε αυτή τη διπλωματική σχεδιάσαμε και υλοποιήσαμε ένα δίκτυο ομοβάθμιων υπολογιστικών μονάδων, το οποίο υποστηρίζει *ερωτήσεις, προφίλ και ειδοποιήσεις*. Το σύστημα αυτό το ονομάσαμε *RQL P2P*, γιατί χρησιμοποιεί την RQL για εκτέλεση ερωτήσεων. Πρόκειται για ένα σύστημα το οποίο μπορεί να χρησιμοποιηθεί για την κατασκευή διάφορων P2P εφαρμογών και εμείς επιλέξαμε να υλοποιήσουμε μία εφαρμογή μοιράσματος αρχείων (*file-sharing*). Οι χρήστες της εφαρμογής αυτής μπορούν :

- 1) να δημοσιεύσουν τα αρχεία τους, έτσι ώστε να μπορούν να τα δουν άλλοι χρήστες και να τα αποθηκεύσουν στον υπολογιστή τους.
- 2) να εκτελέσουν ερωτήσεις πάνω στο σύστημα για να ψάξουν για αρχεία.
- 3) να στείλουν το προφίλ τους, το οποίο παράγει ειδοποιήσεις όχι μόνο για όσους πόρους είναι διαθέσιμοι στο δίκτυο, αλλά και για μελλοντικούς πόρους.
- 4) να λαμβάνουν ειδοποιήσεις πάνω σε αρχεία, τα οποία διαχειρίζονται άλλοι χρήστες, και είναι ενδιαφέροντα σύμφωνα με το προφίλ που έχουν δώσει.
- 5) να λαμβάνουν αποθηκευμένες ειδοποιήσεις για πόρους που προστέθηκαν σε χρονική στιγμή που ο χρήστης δεν ήταν συνδεδεμένος με το δίκτυο και ταιριάζουν με το προφίλ του.
- 6) να κανονίσουν ραντεβού με ένα αρχείο, για το οποίο έχει λάβει ειδοποίηση, αλλά δεν είναι συνδεδεμένος ο χρήστης που το παρέχει.
- 7) να αποθηκεύσουν αρχεία απευθείας από το χρήστη που τα παρέχει.

Το σύστημα RQL P2P είναι ένα υβριδικό δίκτυο ομοβάθμιων υπολογιστικών μονάδων, στο οποίο η πληροφορία που ανταλλάσσεται, ελέγχεται μέσω ενός κεντρικού διακομιστή, ενώ η ροή δεδομένων γίνεται κανονικά μεταξύ των κόμβων. Πρόκειται για ένα P2P δίκτυο βασισμένο σε σχήμα και για την ακρίβεια ένα P2P δίκτυο βασισμένο σε RDF, το οποίο επιτρέπει τις πολύπλοκες και επεκτάσιμες περιγραφές των πόρων, αντί για τις περιορισμένες και αμετάβλητες. Η RDF είναι, όπως είπαμε, μια γλώσσα για αναπαράσταση πληροφορίας των πόρων του διαδικτύου και αυτή η περι-

γραφική της ικανότητα είναι που χρησιμοποιήθηκε για τις πολύπλοκες περιγραφές των πόρων των κόμβων του συστήματος RQL P2P.

Επίσης η χρήση της RDF για την περιγραφή των πόρων παρέχει διευκολύνσεις στην εκτέλεση πολύπλοκων ερωτήσεων πάνω στα μεταδεδομένα, αντί για το απλό ψάξιμο με βάση λέξεις-κλειδιά. Για αυτό το λόγο χρησιμοποιείται ως γλώσσα εκτέλεσης ερωτήσεων η RQL, η οποία παρουσιάζει μεγάλη λειτουργικότητα για την εκτέλεση ερωτήσεων πάνω σε RDF σχήματα και περιγραφές πόρων.

Τα συστήματα ειδοποίησης συμβάντων είναι συστήματα που επιτρέπουν σε κάποιον να δηλώσει ένα προφίλ ενδιαφέροντος, ώστε να ειδοποιηθεί όταν συμβούν συγκεκριμένα γεγονότα που τον ενδιαφέρουν. Στην εφαρμογή μοιράσματος αρχείων που υλοποιήσαμε έχουμε στηριχτεί στο σύστημα ειδοποίησης συμβάντων του συστήματος P2P-DIET. Θεωρούμε ότι ένα προφίλ είναι μία ερώτηση σε RQL μεγάλης χρονικής διάρκειας (*long-standing query*) και συμβάν μπορεί να είναι η ενέργεια ενός χρήστη να δημοσιεύσει ένα αρχείο ή να δηλώσει ένα προφίλ. Το προφίλ συνεχίζει να παράγει αποτελέσματα και στο μέλλον και όσο νέοι πόροι προστίθενται στο σύστημα.

Τέλος την υλοποίηση μας μπορούμε να τη διαχωρίσουμε σε δύο τμήματα : α) στην υλοποίηση του κεντρικού διακομιστή β) στην υλοποίηση ενός κόμβου-παραδείγματος που μπορεί να συνδεθεί με το σύστημα. Για την ακρίβεια, στη δεύτερη περίπτωση υλοποιήσαμε ένα κόμβο που κάνει αναζήτηση πόρων που χρησιμοποιείται το Dublin Core σχήμα [30] για την περιγραφή τους.

1.2 Οργάνωση Της Διπλωματικής

Η διπλωματική είναι οργανωμένη ως εξής : Στο κεφάλαιο 2 παρουσιάζουμε τις τεχνολογίες και τα συστήματα με τα οποία ασχολούμαστε σε αυτήν την διπλωματική και χρησιμοποιήθηκαν στην υλοποίηση. Στο κεφάλαιο 3 παρουσιάζουμε τη δομή και την τοπολογία του δικτύου ομοβάθμιων υπολογιστικών μονάδων που υλοποιήσαμε, καθώς και τις δυνατότητες και λειτουργίες του. Στο κεφάλαιο 4 παρουσιάζουμε τα μηνύματα που χρησιμοποιούν οι κόμβοι και ο διακομιστής του συστήματος RQL P2P, για να επικοινωνήσουν μεταξύ τους. Στο κεφάλαιο 5 παρουσιάζουμε την υλοποίηση του συστήματος RQL P2P, δηλαδή αναφερόμαστε στα πακέτα και εργαλεία λογισμικού που χρησιμοποιήθηκαν για τις διάφορες λειτουργίες, καθώς και στην αρχιτεκτονική του, τα τμήματα δηλαδή από τα οποία αποτελείται. Τέλος στο κεφάλαιο 6 παρουσιάζουμε την περίληψη και τα συμπεράσματα αυτής της διπλωματικής.

Κεφάλαιο 2

Θέματα Και Τεχνολογίες

Σε αυτό το κεφάλαιο παρουσιάζουμε μερικές βασικές έννοιες, τεχνολογίες και συστήματα με τα οποία ασχολούμαστε σε αυτήν την διπλωματική ή συσχετίζονται με την εργασία μας.

2.1 Μεταδεδομένα

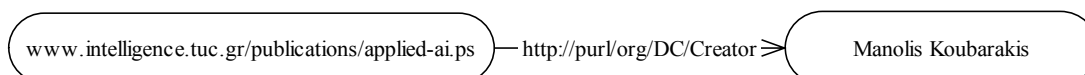
Σήμερα υπάρχει ένας πλούτος πληροφοριών για κάθε θέμα διαθέσιμο στο διαδίκτυο. Για τους περισσότερους ο ενθουσιασμός για το παγκόσμιο ιστό (web) βρίσκεται στην παροχή υπηρεσιών, στις οποίες μπορεί κάποιος να έχει πρόσβαση από το σπίτι ή το γραφείο. Τέτοιες υπηρεσίες είναι οι ειδήσεις, ο καιρός, οικονομικές υπηρεσίες κ.τ.λ. Επίσης, οι χρήστες μπορούν να αγοράσουν βιβλία, υπολογιστές και οποιοδήποτε άλλο προϊόν, όπως και να κλείσουν αεροπορικά εισιτήρια και δωμάτια σε ξενοδοχεία. Οι πιθανές χρήσεις του φαίνονται ατελείωτες και σε αυτό το σημείο γίνεται αναγκαία η ύπαρξη ενός συγκεκριμένου τμήματός του παγκόσμιου ιστού, το οποίο καταχωρεί και περιγράφει τις πληροφορίες του, δομημένων κατά τέτοιο τρόπο ώστε να επιτρέπει στις ιστοσελίδες να αναζητηθούν κατάλληλα και να υποβληθούν σε επεξεργασία από ένα υπολογιστή. Με άλλα λόγια, αυτό που είναι σήμερα αναγκαίο είναι η ύπαρξη των *μεταδεδομένων (metadata)*, δηλαδή *δεδομένων για δεδομένα*.

2.2 RDF

Το πλαίσιο εργασίας περιγραφής πόρων (*Resource Description Framework* ή εν συντομία *RDF*) [1, 2] είναι μια γλώσσα για αναπαράσταση πληροφορίας των πόρων του διαδικτύου. Προορίζεται για την αναπαράσταση των μεταδεδομένων των πόρων του παγκόσμιου ιστού, όπως ο τίτλος, ο δημιουργός, το θέμα και η ημερομηνία τροποποίησης μιας ιστοσελίδας, καθώς και τα πνευματικά δικαιώματα και οι πληροφορίες χορήγησης άδειας για ένα έγγραφο του παγκόσμιου ιστού. Αναπτύσσεται από την *Κοινοπραξία Παγκόσμιου Ιστού (World Wide Web Consortium ή W3C)* [3], ένα φόρουμ που ασχολείται με την ανάπτυξη τεχνολογιών (προδιαγραφές, οδηγίες, λογισμικό και εργαλεία) με σκοπό την πλήρη αξιοποίηση της δύναμης του διαδικτύου. Γενικεύοντας την έννοια ενός "πόρου του παγκόσμιου ιστού", η RDF μπορεί επίσης να χρησιμοποιηθεί για να αναπαραστήσει πράγματα που μπορούν να προσδιοριστούν από το

παγκόσμιο ιστό, ακόμα και όταν δεν μπορούν να ανακτηθούν άμεσα από αυτόν. Τέτοια παραδείγματα είναι οι πληροφορίες για προϊόντα από ηλεκτρονικούς τόπους αγορών (π.χ. πληροφορίες για τις προδιαγραφές, τις τιμές, και τη διαθεσιμότητα), ή την περιγραφή των προτιμήσεων ενός χρήστη για την παράδοση πληροφοριών.

Το βασικό στοιχείο του μοντέλου της RDF είναι η *τριάδα* (*triple*): ένας πόρος (το *θέμα* ή *subject*) συνδέεται με έναν άλλο πόρο (το *αντικείμενο* ή *object*) μέσω μιας ακμής, στην οποία αναγράφεται ένας τρίτος πόρος (το *κατηγορημα* ή *predicate*). Η RDF προσδιορίζει τους πόρους με τα *Ενιαία Αναγνωριστικών Πόρων* (*Uniform Resource Identifier* ή *URI* εν συντομία). Θα λέμε ότι το <θέμα> έχει το <κατηγορημα> ως ιδιότητα με τιμή το <αντικείμενο>. Οι τριάδες αναπαριστώνται με ένα γράφο, του οποίου οι κόμβοι και οι ακμές αναγράφονται με τα κατάλληλα URI. Για παράδειγμα, η τριάδα



Εικόνα 2.1: Μία τριάδα

στην εικόνα 2.1 θα μπορούσε να διαβαστεί ως “Ο Μανόλης Κουμπαράκης είναι ο δημιουργός του πόρου με URI www.intelligence.tuc.gr/publications/applied-ai.ps”. Στην RDF, οι ακμές του γράφου μπορεί να δείχνουν σε κόμβους που αναγράφεται τμήμα κειμένου αντί για το URI ενός πόρου. Αυτό το κείμενο ονομάζεται *κυριολεκτικό* ή *literal*.

Οι θεμελιώδεις έννοιες της RDF είναι:

i) *rdf:Resource*

Η RDF ουσιαστικά χρησιμοποιείται για την περιγραφή των πόρων (resources). Οι πόροι παίρνουν πάντα ως ονομασία κάποιο URI και οτιδήποτε μπορεί να αντιπροσωπευθεί με ένα URI. Έτσι η RDF μπορεί θεωρητικά να χρησιμοποιηθεί για να περιγράψει οτιδήποτε.

ii) *rdf:Property*

Οι *ιδιότητες* (*properties*) είναι πόροι που χρησιμοποιούνται ως κατηγορήματα των τριάδων. Η σημασιολογία μιας τριάδας εξαρτάται από την ιδιότητα που χρησιμοποιείται ως κατηγορημα. Το γεγονός ότι οι ιδιότητες είναι πόροι επιτρέπει την περιγραφή τους από την RDF.

iii) *rdf:Statement*

Μία *πρόταση* (*statement*) είναι ένας πόρος που περιγράφει μία τριάδα. Ένας τέτοιος πόρος πρέπει να έχει τουλάχιστον 3 ιδιότητες: *subject*, *object* και *predicate*, με τιμές τους αντίστοιχους πόρους (ή κυριολεκτικού για την περίπτωση του αντικείμενου).

Υπάρχουν και έννοιες που καθορίζουν σχήματα ώστε να έχουμε ένα λεξιλόγιο των πόρων που χρησιμοποιούμε στην RDF. Τα ονομάζουμε *RDF σχήματα* (*RDF Schemas*). Στα σχήματα, νέοι πόροι μπορούν να οριστούν ως εξειδίκευση παλαιότερων, συνεπάγοντας στη δημιουργία επιπλέον τριάδων. Τα σχήματα περιορίζουν επίσης το πεδίο στο οποίο οι πόροι που έχουν οριστεί, μπορούν να χρησιμοποιηθούν, οδηγώντας στην έννοια της εγκυρότητας σχημάτων. Οι έννοιες καθορισμού σχήματος της RDF είναι:

i) *rdfs:subPropertyOf*

Κάθε ιδιότητα δηλώνει μια σχέση μεταξύ των πόρων (το σύνολο ζευγαριών των πόρων που συνδέονται μεταξύ τους με μία ακμή στην οποία αναγράφεται η ιδιότητα).

Η έννοια *υπο-ιδιότητα* (*subProperty*) εφαρμόζεται στις ιδιότητες και ερμηνεύεται ως σχέση υποσυνόλου της σχέσης που αυτά δηλώνουν. Για παράδειγμα, εάν η ιδιότητα “μητέρα” είναι ένα υπο-ιδιότητα της ιδιότητας “γονέας”, κάθε τριάδα που έχει το “μητέρα” ως κατηγορημα πρέπει να θεωρηθεί ότι έχει και το “γονέας” ως κατηγορημα.

ii) *rdfs:Class*, *rdf:type* και *rdfs:subClassOf*

Οι κλάσεις (*classes*) είναι πόροι που υποδηλώνουν ένα σύνολο από πόρους. Μία περίπτωση πόρου (ένα *στιγμιότυπο* δηλαδή) χρησιμοποιεί την ιδιότητα *rdf:type* με τιμή μία κλάση, για να δηλώσει ότι ανήκει σε αυτήν την κατηγορία κλάσης. Από την άλλη μεριά, όλες οι ιδιότητες χρησιμοποιούν την ιδιότητα *rdf:type* με τιμή *rdf:property*. Οι κλάσεις είναι δομημένες με παρόμοιο τρόπο με τις ιδιότητες, δηλαδή με μία ιεραρχία υποσυνόλων που δηλώνεται με τη χρήση της ιδιότητας *rdfs:subClassOf* που σημαίνει *υποκλάση*.

iii) *rdfs:domain* και *rdfs:range*

Πρόκειται για ιδιότητες που εφαρμόζονται από ιδιότητες και παίρνουν ως τιμές κλάσεις. Χρησιμοποιούνται για να περιορίσουν το σύνολο των πόρων που μπορούν να έχουν μια συγκεκριμένη ιδιότητα, δηλαδή το *πεδίο* (*domain*) της ιδιότητας και το σύνολο των έγκυρων τιμών που μπορεί να έχει μία ιδιότητα, δηλαδή το *εύρος* (*range*) της ιδιότητας. Μία ιδιότητα μπορεί να έχει ως τιμές του πεδίου όσες χρειάζονται, αλλά όχι περισσότερες από μία για το εύρος. Για να είναι μία τριάδα έγκυρη, το αντικείμενο θα πρέπει να ταιριάζει με το εύρος (εφόσον υπάρχει) του κατηγορήματος (θα πρέπει δηλαδή το κατηγορημα να χρησιμοποιεί το *rdf:type* με την αντίστοιχη κλάση ή με μία από τις υποκλάσεις του), και το θέμα θα πρέπει να ταιριάζει με τουλάχιστον με ένα από τα παιδιά ορισμού (εφόσον υπάρχουν) του κατηγορήματος (εάν το κατηγορημα είναι υπο-ιδιότητα άλλων ιδιοτήτων, θα πρέπει να ελεγχθούν κατά τον ίδιο τρόπο).

Ένα έγγραφο της RDF είναι μία λίστα με περιγραφές. Κάθε περιγραφή εφαρμόζεται συνήθως σε ένα πόρο και περιέχει μία λίστα από ιδιότητες. Οι τιμές των ιδιοτήτων είναι είτε URIs είτε κυριολεκτικά είτε άλλες περιγραφές. Στην XML [4] τα μεταδεδομένα της RDF περιέχονται μέσα σε ένα *στοιχείο* (*element*) που έχει όνομα *rdf:RDF*. Αυτό το στοιχείο περιέχει μία σειρά από στοιχεία που έχουν όνομα *rdf:Description*. Αυτά τα στοιχεία μπορούν να έχουν ένα από τα δύο *χαρακτηριστικά* (*attributes*) *rdf:about* ή *rdf:ID* (αλλά όχι και τα δύο).

- Το *rdf:about* χρησιμοποιείται για να περιγράψει ένα πόρο. Η τιμή του μπορεί να είναι ένα απόλυτο ή σχετικό URI

```
<rdf:Description rdf:about="http://www.xyz.com/sw.html">
...
</rdf:Description>
```

- Το *rdf:ID* χρησιμοποιείται για να ορίσει ένα πόρο

```
<rdf:Description rdf:ID="something">
...
</rdf:Description>
```

- Μία περιγραφή χωρίς *rdf:about* και *rdf:ID* χρησιμοποιείται για να περιγράψει ένα ανώνυμο πόρο.

```
<rdf:Description>
```

```
...
</rdf:Description>
```

Ένα στοιχείο *rdf:Description* περιέχει μία σειρά από XML στοιχεία. Αυτά τα στοιχεία ερμηνεύονται ως ιδιότητες και το όνομά τους επεκταμένο αντιστοιχεί στο URI του κατηγορήματος. Εάν το στοιχείο είναι άδειο, θα πρέπει να έχει ως χαρακτηριστικό το *rdf:resource*, του οποίου η τιμή είναι το URI του αντικειμένου. Αλλιώς, μπορεί να περιέχει κείμενο ή ένα εσωτερικό *rdf:Description* στοιχείο.

```
<?xml version="1.0" encoding="UTF-8" ?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:dc="http://purl.org/dc/elements/1.1#">

  <rdf:Description rdf:about="http://www.xyz.com/sw.html">
    <dc:Creator rdf:resource="someone@abc.gr"/>
    <dc>Title> Software Engineering </dc>Title>
  </rdf:Description>
</rdf:RDF>
```

2.3 Συστήματα P2P

Στα δίκτυα ομοβάθμιων υπολογιστικών μονάδων (στα P2P συστήματα δηλαδή) οι κόμβοι του δικτύου έχουν ίσους ρόλους στην ανταλλαγή πληροφορίας και υπηρεσιών. Διάφορα συστήματα έχουν χαρακτηριστεί ως P2P συστήματα σε διαφορετικές περιοχές εφαρμογών. Τα γνωστότερα συστήματα είναι αυτά που χρησιμοποιούνται για την ανταλλαγή μουσικών αρχείων, όπως για παράδειγμα τα συστήματα Napster, Gnutella και Freenet. Επίσης υπάρχουν P2P συστήματα, όπως το ICQ [32] για ανταλλαγή προσωπικών μηνυμάτων, ή συστήματα συνεργασίας όπως το Groove [33]. Το μεγάλο πλεονέκτημα των P2P συστημάτων είναι ότι μπορούν να συγκεντρώσουν μεγάλη ποσότητα από πληροφορία και υπολογιστική δύναμη, με το να συγκεντρώνουν τους πόρους και τους κύκλους κεντρικής μονάδας επεξεργασίας (CPU cycles) των χρηστών.

Στη συνέχεια θα ταξινομήσουμε τα P2P συστήματα σε τρεις κατηγορίες : στα *καθαρά P2P συστήματα*, στα *υβριδικά P2P συστήματα* και τα συστήματα *υπερ-κόμβων*. Θα παρουσιάσουμε κάθε τοπολογία ξεχωριστά, αναφέροντας σε κάθε περίπτωση ένα γνωστό παράδειγμα P2P συστήματος που αντιστοιχεί εκεί. Επίσης θα τις συγκρίνουμε με βάση συγκεκριμένα χαρακτηριστικά, όπως την *προσαρμοστικότητα (scalability)*, την *συνοχή (coherence)*, την *αξιοπιστία (reliability)* και την *ευκολία διαχείρισης (manageability)*.

2.3.1 Καθαρά P2P Συστήματα

Στην τοπολογία των καθαρών P2P συστημάτων (*pure P2P systems*) δεν υπάρχει κεντρικός ελεγκτής (δεν υπάρχει κεντρικός διακομιστής δηλαδή) και όλοι οι κόμβοι συνδέονται μεταξύ τους με τον κλασσικό τρόπο που αντιστοιχεί σε ένα P2P σύστημα. Αυτή η τοπολογία έχει πολύ υψηλή απόδοση, καθώς τα δεδομένα μεταφέρονται με απευθείας P2P τρόπο. Επιπλέον, το σύστημα αυτό είναι τελείως προσαρμόσιμο, καθώς μπορούν να προστεθούν πολλοί επιπλέον κόμβοι χωρίς πρόβλημα στο δίκτυο. Η

τοπολογία είναι επίσης αρκετά αξιόπιστη, καθώς η βλάβη σε ένα από τους κόμβους δε σταματάει την ανταλλαγή δεδομένων μεταξύ των άλλων κόμβων.

Από την άλλη μεριά, η τοπολογία των καθαρών P2P συστημάτων παρουσιάζει μερικά εμφανή μειονεκτήματα. Κάθε κόμβος μπορεί να εισάγει βλαβερά δεδομένα στο σύστημα, τα οποία μπορούν να προκαλέσουν πρόβλημα στο σύστημα. Τα δεδομένα στο δίκτυο δεν είναι επίσης συνεκτικά. Από τη στιγμή που δεν υπάρχει διακομιστής που να ελέγχει, η ασφάλεια γίνεται μείζον ζήτημα. Τέλος η τοπολογία είναι ανεκτική σε σφάλματα, καθώς η βλάβη ή η αποσύνδεση ενός συγκεκριμένου κόμβου δεν έχει επιπτώσεις στο υπόλοιπο σύστημα.

Το σύστημα Gnutella είναι ένα καλό παράδειγμα καθαρού P2P δικτύου. Το Gnutella είναι ένα πρωτόκολλο μοιράσματος αρχείων, πάνω στο οποίο έχουν υλοποιηθεί εφαρμογές που επιτρέπουν στους χρήστες να αναζητούν και να αποθηκεύουν αρχεία από άλλους χρήστες. Ένας χρήστης του Gnutella πρέπει να γνωρίζει τη διεύθυνση διαδικτύου ενός κόμβου του Gnutella για να συνδεθεί στο δίκτυο. Όταν ένας χρήστης θέλει να κάνει αναζήτηση, στέλνει την ερώτηση σε όλους τους γείτονές του. Το μήνυμα θα ταξιδέψει μόνο σε ένα περιορισμένο αριθμό από κόμβους, καθώς υπάρχει χρονικό όριο στη διάδοση του μηνύματος στους διάφορους κόμβους.

2.3.2 Υβριδικά P2P Συστήματα

Η τοπολογία των υβριδικών P2P συστημάτων (*hybrid P2P systems*) έχει ένα μοναδικό διακομιστή ελέγχου. Υπό αυτή τη μορφή, όλη η ροή των δεδομένων συμβαίνει με το P2P τρόπο, οδηγώντας σε σχετικά υψηλή απόδοση.

Αυτή η τοπολογία προσπαθεί να συνδυάσει τα πλεονεκτήματα των κεντροκοιμημένων και των αποκεντρωμένων συστημάτων. Από τη στιγμή που υπάρχει ένας μόνο διακομιστής ελέγχου, η διαχείριση του συστήματος γίνεται ευκολότερη και τα δεδομένα είναι συνεκτικά. Αυτή η προσέγγιση λαμβάνει το όφελος της κεντρικής ασφάλειας και διαχείρισης, γιατί όλες οι μηχανές διευθετούνται από το κεντρικό διακομιστή.

Το υβριδικό P2P σύστημα έχει και τα δικά του μειονεκτήματα. Από τη στιγμή που υπάρχει ένας μόνο διακομιστής ελέγχου, μεγάλο πρόβλημα δημιουργείται από την πιθανή παρουσία ενός σημείου σφάλματος. Για παράδειγμα, εάν ο διακομιστής ανταλλαγή ελέγχου καταρρεύσει, κανένας νέος κόμβος δεν μπορεί να προστεθεί και καμία νέα δεδομένων μεταξύ κόμβων δεν μπορεί να ξεκινήσει. Η απόδοση μειώνεται επίσης σε σύγκριση με τα καθαρά P2P συστήματα, καθώς ο έλεγχος της πληροφορίας παίρνει μεγαλύτερο μονοπάτι, προκαλώντας επιβραδύνσεις όσο περισσότεροι πελάτες έχουν πρόσβαση στον ίδιο διακομιστή.

Ένα γνωστό υβριδικό σύστημα είναι το Napster, στο οποίο οι πελάτες συνδέονται με το διακομιστή, ο οποίος κρατάει ένα κατάλογο με τους πόρους όλων των κόμβων. Οι διακόμιστες του Napster είναι οργανωμένοι σε μία ασύνδετη αρχιτεκτονική, κάτι που σημαίνει ότι ένας χρήστης μπορεί να δει μόνο τα αρχεία των χρηστών που είναι συνδεδεμένοι στον ίδιο διακομιστή. Οι χρήστες του Napster δεν μπορούν να ψάξουν για αρχεία καθολικά, επειδή είναι περιορισμένοι να αναζητούν σε ένα μόνο διακομιστή που έχει καταχωρήσει ένα ποσοστό των διαθέσιμων αρχείων του δικτύου. Ο μόνος τρόπος για να επιτύχει καθολική αναζήτηση ένας χρήστης είναι να συνδεθεί με όλους τους διακόμιστες και να αναζητήσει.

2.3.3 Συστήματα Υπερ-κόμβων

Τα συστήματα υπερ-κόμβων (*Super-Peer Systems*) είναι μία βέλτιστη αρχιτεκτονική μιας κεντριοποιημένης τοπολογίας ενσωματωμένης σε αποκεντρωμένα συστήματα. Ένας υπερ-κόμβος είναι ένας κόμβος του δικτύου που δρα τόσο ως διακομιστής για ένα υποσύνολο πελατών, όσο και ως ίσος κόμβος σε ένα δίκτυο υπερ-κόμβων.

Από τη στιγμή που τα δεδομένα συνεχίζουν να μεταφέρονται με το P2P τρόπο, η απόδοση είναι υψηλή. Αρκετοί υπερ-κόμβοι υπάρχουν, κάθε ένας από τους οποίους διαχειρίζεται το δικό του σύνολο από πελάτες, οδηγώντας σε διευκόλυνση της διαχείρισης και της ασφάλειας. Το γενικό σύστημα είναι πιο ασφαλές καθώς οι υπερ-κόμβοι ελέγχουν τη ροή των δεδομένων που παράγονται από κάθε ένα από τους πελάτες, ενώ εντοπίζονται αυτοί που έχουν αποτύχει, περιορίζοντας τα σημεία βλάβης στο δίκτυο. Ο γενικός φόρτος εργασίας χωρίζεται μεταξύ των διάφορων κόμβων, κάνοντας το σύστημα πιο προσαρμοστικό.

Ένα παράδειγμα ενός συστήματος υπερ-κόμβων είναι το σύστημα *P2P-DIET* [35, 36, 37, 38], από το οποίο αντλήσαμε αρκετές πληροφορίες και ιδέες για την υλοποίηση αυτής της διπλωματικής και το οποίο θα παρουσιάσουμε στην επόμενη ενότητα.

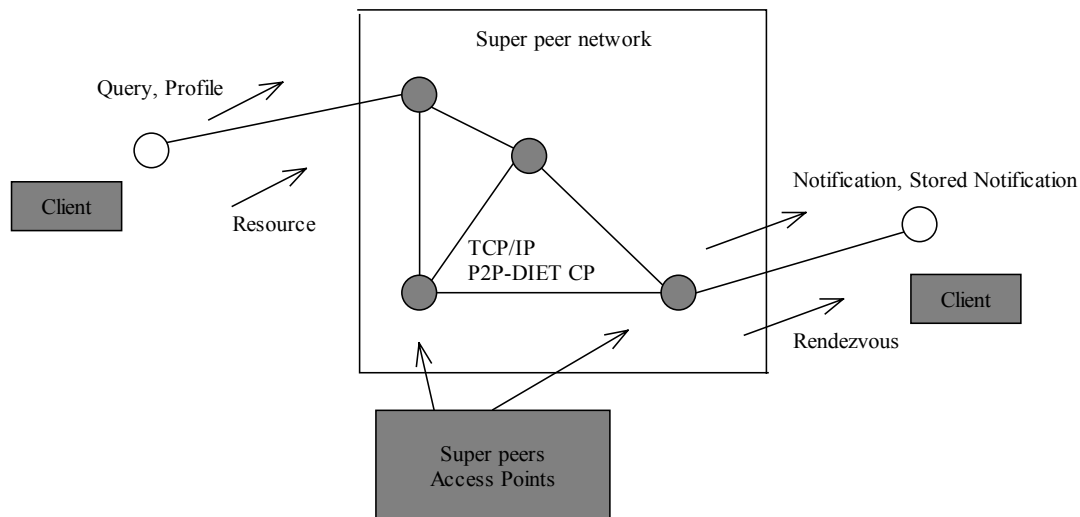
2.3.4 Το Σύστημα P2P-DIET

Το σύστημα P2P-DIET είναι ένα βασισμένο σε *πράκτορες (agents)* P2P σύστημα, το οποίο υποστηρίζει ερωτήσεις, προφίλ και ειδοποιήσεις. Η υπηρεσία από μόνη της μπορεί να χρησιμοποιηθεί για την κατασκευή διάφορων P2P εφαρμογών όπως μοίρασμα αρχείων, ηλεκτρονικό εμπόριο, διαχείριση δικτύου και διαχείριση λογισμικού. Το σύστημα περιέχει ένα μηχανισμό ανοχής λαθών (*fault-tolerance*), ο οποίος εγγυάται συνδετικότητα, όταν οι κόμβοι του δικτύου αποτυγχάνουν ή βγουν απροειδοποίητα από το δίκτυο. Υποστηρίζεται η δυνατότητα της ανεξάρτητης-από-τοποθεσία διευθυνσιοδότησης, η οποία επιτρέπει τη χρήση δυναμικών διευθύνσεων διαδικτύου χρήστη. Με αυτή τη δυνατότητα, οι κόμβοι μπορούν να αποσυνδέονται και να επανασυνδέονται με διαφορετικές διευθύνσεις σε διαφορετικά μέρη του δικτύου.

Μία εφαρμογή μοιράσματος αρχείων έχει υλοποιηθεί πάνω σε αυτή την υπηρεσία. Οι χρήστες αυτής της εφαρμογής μπορούν :

- 1) να δημοσιεύουν τα αρχεία τους, έτσι ώστε άλλοι χρήστες να μπορούν να τα δουν και να τα αποθηκεύσουν.
- 2) να εκτελέσουν ερώτηση στο σύστημα για να αναζητήσουν αρχεία σε ολόκληρο το δίκτυο.
- 3) να δηλώσουν το προφίλ τους, το οποίο θα συνεχίσει να παράγει ειδοποιήσεις σε μελλοντικούς πόρους.
- 4) να λαμβάνουν ειδοποιήσεις σε αρχεία ενδιαφέροντος, τα οποία ανήκουν σε άλλους χρήστες.
- 5) να λαμβάνουν αποθηκευμένες ειδοποιήσεις, οι οποίες παράχθηκαν σε χρονική στιγμή που ο χρήστης δεν ήταν συνδεδεμένος για να τις λάβει.
- 6) να κανονίσουν ραντεβού με ένα αρχείο, εάν ο χρήστης που το παρέχει δεν είναι συνδεδεμένος.
- 7) να χρησιμοποιήσουν δυναμικές διευθύνσεις διαδικτύου και διαφορετικούς κόμβους-σημεία πρόσβασης στο δίκτυο.
- 8) να κατεβάσουν αρχεία απευθείας από το χρήστη που παρέχει τον πόρο.
- 9) να εντοπίσουν άλλους χρήστες και πιθανώς να συνομιλήσουν μαζί τους.

Η αρχιτεκτονική για αυτό το P2P σύστημα είναι αυτή του δικτύου υπερ-κόμβων, η οποία επιτρέπει στους υπερ-κόμβους να επικοινωνούν μέσα από πολλαπλά μονοπάτια



Εικόνα 2.2 : Η αρχιτεκτονική του P2P-DIET

και να σχηματίσουν ένα γενικό μη κατευθυνόμενο γράφο, όπως φαίνεται στην εικόνα 2.2. Υπάρχουν δύο ειδών κόμβοι στο σύστημα P2P-DIET. Οι υπερ-κόμβοι και οι κόμβοι-πελάτες. Οι υπερ-κόμβοι συνδέονται με τέτοιο τρόπο, ώστε σχηματίζουν ένα γενικό μη κατευθυνόμενο γράφο. Όλοι οι υπερ-κόμβοι είναι ίσοι και έχουν τις ίδιες ευθύνες και έτσι το εσωτερικό δίκτυο που αποτελείται μόνο από υπερ-κόμβους είναι ένα καθαρό P2P δίκτυο. Κάθε υπερ-κόμβος εξυπηρετεί ένα ποσοστό από τους πελάτες και κρατάει σε κατάλογο τους πόρους των άλλων πελατών. Οι πελάτες-κόμβοι είναι ίσοι μεταξύ τους στο κατέβασμα. Όταν ένας πελάτης θέλει να κατεβάσει ένα πόρο, τον κατεβάζει με ένα τρόπο όπως σε ένα καθαρό P2P, από το πελάτη που παρέχει τον πόρο. Ένας κόμβος-πελάτης συνδέεται με το δίκτυο μέσα από ένα υπερ-κόμβο, ο οποίος είναι το σημείο πρόσβασης αυτού του πελάτη. Ο κόμβος-πελάτη δεν είναι απαραίτητο να είναι συνδεδεμένος στο ίδιο σημείο πρόσβασης συνέχεια. Έτσι με αυτή τη τοπολογία μπορούν να αντιμετωπιστούν αποδοτικά κρίσιμες καταστάσεις, όπως η αποτυχία ενός κόμβου.

Οι υπολογισμοί βασισμένοι σε πράκτορες προσφέρουν πολλά επιθυμητά χαρακτηριστικά για την υλοποίηση ενός P2P συστήματος που υποστηρίζει ερωτήσεις, προφίλ και ειδοποιήσεις. Οι πράκτορες λογισμικού είναι δυναμικές, ευπροσάρμοστες, υπολογιστικές οντότητες που λειτουργούν συνεχόμενα και αυτόνομα σε ένα συγκεκριμένο περιβάλλον, στο οποίο συχνά κατοικούν και άλλες διεργασίες πρακτόρων. Είναι ικανοί να κατοικήσουν και να μετακομίσουν σε περίπλοκα, απρόβλεπτα, δυναμικά και ετερογενή περιβάλλοντα. Οι πράκτορες λογισμικού εκτελούν δραστηριότητες με ένα ευέλικτο τρόπο και είναι ευαίσθητοι στις αλλαγές του περιβάλλοντος, χωρίς να χρειάζεται σταθερή ανθρώπινη βοήθεια. Αρκετή έρευνα έχει επικεντρωθεί πάνω στα συστήματα βασισμένα σε πράκτορες και πολλά εργαλεία ανάπτυξης πρακτόρων έχουν κατασκευαστεί τα τελευταία χρόνια. Η πλατφόρμα πρακτόρων που ονομάζεται DIET Core [41] χρησιμοποιήθηκε για την υλοποίηση του P2P-DIET.

Η βασική έννοια του DIET Core είναι ο *κόσμος (world)*. Για κάθε *εικονική μηχανή της Java (Java Virtual Machine ή εν συντομία JVM)* αντιστοιχεί και ένας κόσμος. Ένας κόσμος είναι μία αποθήκη περιβαλλόντων (*environments*) και μπορεί να περιέχει ένα ή περισσότερα περιβάλλοντα. Τα περιβάλλοντα στο DIET Core παρέχουν μία τοποθεσία για τους πράκτορες για να κατοικήσουν και μπορούν να φιλοξενήσουν έναν ή περισσότερους πράκτορες. Η πλατφόρμα του DIET μπορεί να περιέχει περισσότερους από ένα κόσμους. Αυτό συμβαίνει για παράδειγμα στην περίπτωση που το

DIET Core τρέχει σε πολλούς υπολογιστές. Αυτό μπορεί να θεωρηθεί ως το *σύμπαν* (*universe*) που περιέχει πολλούς πλανήτες (κόσμους). Κάθε περιβάλλον έχει γειτονικούς συνδέσμους προς άλλα περιβάλλοντα, τα οποία δεν ανήκουν απαραίτητα στον ίδιο κόσμο. Κάθε περιβάλλον μπορεί να συνδεθεί με άλλα περιβάλλοντα που δεν είναι απαραίτητα στον ίδιο κόσμο, μέσα από *γειτονικούς συνδέσμους* (*neighbor links*). Αυτοί οι σύνδεσμοι επιτρέπουν στους πράκτορες να μετακομίσουν σε διαφορετικά περιβάλλοντα και να εξερευνήσουν το διάστημα του DIET, χωρίς να έχουν προηγούμενη γνώση της τοποθεσίας των περιβαλλόντων. Αυτοί οι σύνδεσμοι ορίζονται όταν ο κόσμος στήνεται αλλά μπορούν να αλλάξουν δυναμικά, εάν το διάστημα του DIET αλλάξει.

Οι πράκτορες του DIET Core είναι πολύ *ελαφριοί*. Δεν χρειάζονται πολύ μνήμη για να τρέξουν ενώ το DIET εξασφαλίζει ότι οι πράκτορες εγκαταλείπουν το νήμα τους εάν δεν το χρειάζονται. Το DIET Core δίνει ένα νήμα σε ένα πράκτορα, όταν αυτός το χρειάζεται, όπως όταν για παράδειγμα πρέπει αυτός να χειριστεί ένα εισερχόμενο μήνυμα. Ο ελαφρύς χαρακτήρας των πρακτόρων επιτρέπει σε εκατοντάδες χιλιάδες διαφορετικούς πράκτορες να τρέξουν στο ίδιο JVM σε ένα συνηθισμένο υπολογιστή.

Στο διάστημα του P2P-DIET υπάρχουν δύο διαφορετικών ειδών περιβάλλοντα. Το περιβάλλον του πελάτη-κόμβου στους κόμβους των πελατών και το περιβάλλον υπερ-κόμβου στους υπερ-κόμβους. Κάθε περιβάλλον φιλοξενεί διαφορετικών ειδών πράκτορες, με διαφορετικές ευθύνες και σκοπούς. Ο σκοπός ενός πράκτορα είναι μερικές φορές ένα μέρος ενός μεγαλύτερου σκοπού και εργάζεται μαζί με άλλους πράκτορες για να επιτύχουν τον σκοπό. Η τοπική επικοινωνία επιτυγχάνεται με απευθείας σύνδεση μεταξύ των δύο πρακτόρων. Η απομακρυσμένη επικοινωνία βασίζεται στους κινητούς πράκτορες, που είναι το είδος του *αγγελιοφόρου* πράκτορα (*messenger*). Αυτοί οι πράκτορες μετακομίζουν σε διαφορετικά περιβάλλοντα για να παραδώσουν μηνύματα και συνεχίζουν να ταξιδεύουν γύρω από το δίκτυο για πάντα.

Το P2P-DIET αντιμετωπίζει και όλα τα προβλήματα δρομολόγησης. Έχει ως στόχο την εγκατάσταση των κατάλληλων μονοπατιών ώστε η χρήση του δικτύου να είναι αποδοτική με τη λιγότερη δυνατή επιβάρυνση εργασιών. Η κατασκευή δέντρων ελάχιστου βάρους και μικρότερων μονοπατιών στο κατανεμημένο περιβάλλον, ικανοποιούν τις απαιτήσεις του δικτύου αναφορικά με τα μηνύματα δρομολόγησης. Το P2P-DIET παρουσιάζει διεργασίες για την προσθήκη ή διαγραφή διακομιστών ή πελατών από το δίκτυο, με ένα τρόπο που εγγυάται ευστάθεια και συνδετικότητα. Επίσης παρουσιάζει στρατηγική χειρισμού των sockets, η οποία επιτρέπει στους υπερ-κόμβους να χρησιμοποιούν τον ελάχιστο αριθμό από πόρους. Τέλος όπως αναφέραμε και στην αρχή, το P2P-DIET παρουσιάζει μηχανισμό ανοχής λαθών και επιτρέπει την χρήση δυναμικών διευθύνσεων διαδικτύου.

2.4 Εργαλεία Του Σημασιολογικού Διαδικτύου

Το *Σημασιολογικό Διαδίκτυο* (*Semantic Web*) είναι μία επέκταση του τωρινού παγκόσμιου ιστού, στο οποίο η πληροφορία δίνεται με καθορισμένη σημασιολογία, επιτρέποντας στους υπολογιστές και στους χρήστες να συνεργαστούν καλύτερα. Στηρίζεται στην ιδέα ότι τα δεδομένα στο παγκόσμιο ιστό πρέπει να ορίζονται και να συνδέονται με τέτοιο τρόπο, ώστε να μπορούν να χρησιμοποιηθούν περισσότερο αποδοτικά για ανακάλυψη, αυτοματισμό, ολοκλήρωση και επαναχρησιμοποίηση μέσα από διάφορες εφαρμογές. Για να φτάσει ο παγκόσμιος ιστός την πλήρη δυνατότητά του, θα πρέπει να εξελιχθεί στο σημασιολογικό διαδίκτυο, παρέχοντας μία καθολικά προσπελάσιμη πλατφόρμα η οποία επιτρέπει στα δεδομένα να μοιράζονται και να επεξεργάζονται από αυτοματοποιημένα εργαλεία και χρήστες.

Στο επόμενο στάδιο εξέλιξης του σημασιολογικού διαδικτύου, τεράστιες ποσότητες από πληροφορίες πόρων (π.χ. δεδομένα, έγγραφα, προγράμματα) θα είναι διαθέσιμα μαζί με διάφορα είδη από περιγραφική πληροφορία, π.χ. μεταδεδομένα. Καλύτερη γνώση της σημασίας, χρήσης και δυνατότητας πρόσβασης των πόρων του παγκόσμιου ιστού θα διευκολύνει αρκετά την αυτόματη διεργασία του διαθέσιμου περιεχομένου και υπηρεσιών του. Η RDF επιτρέπει τη δημιουργία και ανταλλαγή των μεταδεδομένων των πόρων όπως συμβαίνει με τα δεδομένα του παγκόσμιου ιστού. Για την ερμηνεία των μεταδεδομένων στις κοινωνίες των χρηστών, η RDF επιτρέπει τον ορισμό των κατάλληλων σχημάτων (RDFS).

Σε αυτήν την ενότητα, παρουσιάζουμε δύο εργαλεία του σημασιολογικού διαδικτύου, τα οποία διαχειρίζονται τις περιγραφές και τα σχήματα της RDF και αποδοτικά αποθηκεύουν, προσπελαίνουν και εκτελούν ερωτήσεις στις περιγραφές των πόρων του παγκόσμιου ιστού. Αυτά είναι i) το ICS-FORTH RDFSuite [11] και το ii) Jena 2 [12]

2.4.1 Το ICS-FORTH RDFSuite

Το RDFSuite είναι ένα υψηλού επιπέδου και εξελίσσιμο εργαλείο λογισμικού επιτρέποντας την υλοποίηση της πλήρους δυνατότητας του σημασιολογικού διαδικτύου. Τα κύρια χαρακτηριστικά του είναι τα εξής:

a) Το RDF Schema Specific Database (RSSDB)

Η βάση δεδομένων συγκεκριμένων RDF σχημάτων (*RDF Schema Specific Data Base* ή εν συντομία *RSSDB* [13]) είναι ένας διαρκής αποθηκευτικός χώρος για φόρτωμα RDF περιγραφών, σε μία αντικειμενοστραφή-σχεσιακή βάση δεδομένων (ORDBMS), εκμεταλλεύοντας τη διαθέσιμη γνώση από τα RDF σχήματα. Διατηρεί την ευελιξία της RDF στον καθορισμό σχημάτων και στον εμπλουτισμό με νέες περιγραφές κάθε χρονική στιγμή, ενώ μπορεί να αποθηκεύσει περιγραφές των πόρων που δημιουργήθηκαν σύμφωνα με ένα ή περισσότερα συσχετιζόμενα RDF σχήματα. Ο κύριος στόχος της αναπαράστασης συγκεκριμένων σχημάτων στο RSSDB είναι ο διαχωρισμός του RDF σχήματος από την πληροφορία των δεδομένων, καθώς και ο διαχωρισμός μεταξύ των μοναδιαίων και δυαδικών σχέσεων που διατηρούν τα στιγμιότυπα των κλάσεων και των ιδιοτήτων. Το RSSDB έχει υλοποιηθεί πάνω σε μία ORDBMS όπως η PostgreSQL [14] (έκδοση v7.3 ή και πιο πρόσφατη). Περιέχει μία μονάδα φόρτωσης (Loading module) και μία μονάδα ενημέρωσης (Update module), και τα δύο υλοποιημένα σε Java [15] χρησιμοποιώντας ένα αριθμό από θεμελιώδεις μεθόδους (APIs) για εισαγωγή, διαγραφή, και τροποποίηση των τριάδων της RDF. Η πρόσβαση στην ORDBMS στηρίζεται στην διασύνδεση JDBC (v2.0). Για την εγκατάσταση και τρέξιμο του RSSDB, χρειάζεται Java (TM) 1.4 ή πιο πρόσφατης έκδοσης και ως σύστημα διαχείρισης βάσεως δεδομένων (DBMS) η PostgreSQL (v7.3 ή πιο πρόσφατης έκδοσης).

Ο χώρος ονόματος (*namespace*) που καθορίζει το RSSDB σε ένα σχήμα που αποθηκεύεται (πρόκειται για URI που αντιστοιχεί σε αυτό το σχήμα), εξαρτάται από το αρχείο εισόδου που δίνεται στο RSSDB. Εάν η είσοδος στο RSSDB είναι ένας Ενιαίος Εντοπιστής Πόρων (URL), δηλαδή ένα απομακρυσμένο χώρο ονόματος π.χ. <http://somewhere/schema.rdf>

ο χώρος ονόματος που θα καθοριστεί θα είναι το ίδιο το URL. Για παράδειγμα, ο χώρος ονόματος που καθορίζεται από το RSSDB στο προηγούμενο χώρο ονόματος θα είναι <http://somewhere/schema.rdf#>. Εάν η είσοδος στο RSSDB είναι ένα τοπικό αρ-

χείο (π.χ. ένα τοπικός χώρος ονόματος) π.χ. /path/somewhere/schema.rdf (ένα αρχείο Unix) ή C:\somewhere\schema.rdf (αρχείο των windows) ο χώρος ονόματος που καθορίζεται στο σχήμα θα είναι το πλήρες όνομα αρχείου με το πρόθεμα <file:> στην αρχή και την κατάληξη # στο τέλος. Για τις προηγούμενες περιπτώσεις θα καθορίζονται τα αντίστοιχα namespaces file:/path/ somewhere/schema.rdf # και file:C:\somewhere/schema.rdf #.

Το RSSDB δίνει στον χρήστη την επιλογή να καθορίσει οποιοδήποτε χώρο ονόματος αυτός επιθυμεί. Για να φορτωθεί ένα αρχείο στο RSSDB, πρέπει πρώτα να φορτωθούν τα σχήματα που χρησιμοποιούνται σε αυτό το αρχείο. Αυτά τα σχήματα δηλώνονται στο μέρος που δηλώνονται οι χώροι ονομάτων του αρχείου, ως τιμές των χαρακτηριστικών του στοιχείου rdf:RDF, π.χ.

```
<?xml version="1.0" ?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:sch="http://somewhere/schema.rdf#">
<sch:RealWorldObject rdf:ID="Artist" />
<sch:RealWorldObject rdf:ID="Artifact" />
```

Τα URIs των χώρων ονομάτων που δηλώνονται θα πρέπει να αντιστοιχούν σε χώρους ονομάτων που καθορίζονται από το RSSDB. Για παράδειγμα, για να αποθηκευτεί το προηγούμενο αρχείο, θα πρέπει το σχήμα schema.rdf να έχει αποθηκευτεί και να του έχει καθοριστεί ο χώρος ονόματος http://somewhere/schema.rdf#.

Το RSSDB έχει κάποιους περιορισμούς. Δεν υποστηρίζει την αποθήκευση των *αποδεκτών* (Containers) της RDF. Έτσι, για να καθοριστεί μία ιδιότητα σε ένα πόρο πολλές φορές, θα πρέπει η ιδιότητα να καθορίζεται στον πόρο επανειλημμένως. Το RSSDB δεν υποστηρίζει επίσης την αποθήκευση του <rdf:value>, αλλά κάποιος μπορεί να ορίσει δικιά του ιδιότητα, κατασκευάζοντας μία νέα ιδιότητα που έχει την ίδια χρήση με το <rdf:value>, αλλά έχει διαφορετικό χώρο ονόματος. Τέλος δεν υποστηρίζει ιδιότητες της RDF με απροσδιόριστο ή πολλαπλό πεδίο (ή εύρος). Έτσι, για ιδιότητες που μπορούν να περιγράψουν κάθε πόρο, το πεδίο δεν θα πρέπει να μείνει απροσδιόριστο, αλλά θα πρέπει να είναι ίσο με την κλάση rdfs:Resource. Αντίστοιχα, εάν η τιμή μίας ιδιότητας μπορεί να είναι οποιοδήποτε πόρο, η κλάση rdfs:Resource θα πρέπει να τεθεί ως εύρος.

b) To Validating RDF Parser (VRP)

Ο *έγκυρος RDF συντακτικός αναλυτής* (ICS-FORTH Validating RDF Parser ή εν συντομία *VRP* [16]) είναι ένα εργαλείο για ανάλυση, έλεγχο και επεξεργασία RDF σχημάτων και των περιγραφών πόρων. Αποτελείται από ένα *συντακτικό αναλυτή* (Parser) που αναλύει συντακτικά τις προτάσεις ενός RDF/ XML αρχείου, σύμφωνα με τις προδιαγραφές της RDF. Επίσης περιέχει ένα επικυρωτή (*Validator*) που ελέγχει εάν οι προτάσεις που περιέχονται στα RDF σχήματα και στις περιγραφές των πόρων ικανοποιούν τους σημασιολογικούς περιορισμούς, όπως ορίζονται από τις προδιαγραφές του RDF σχήματος. Η διανομή του VRP απαιτεί Java (TM) 1.4.

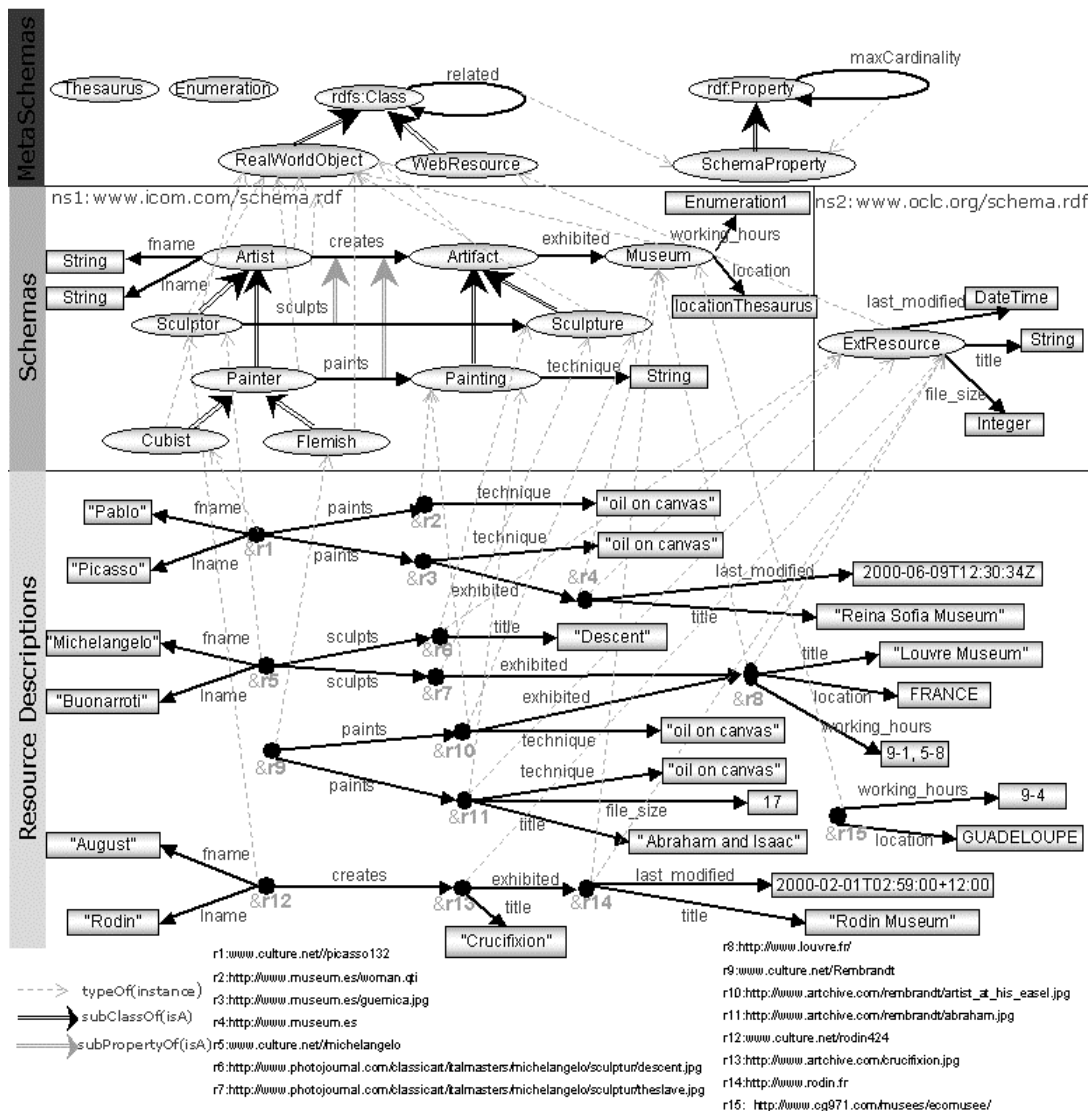
c) To RDF Query Language (RQL)

Η *γλώσσα ερωτήσεων της RDF* (RDF Query Language ή εν συντομία *RQL* [17, 18]) είναι μία γλώσσα για εκτέλεση ερωτήσεων πάνω σε RDF σχήματα και περιγραφές πόρων. Ο ερμηνευτής της RQL (Interpreter), έκδοση v2.0, έχει υλοποιηθεί σε C++ [19] πάνω από μία ORDBMS (PostgreSQL v7.3 ή νεότερη έκδοση) χρησιμοποιώ-

ντας την αρχιτεκτονική πελάτη/διακομιστή για Solaris και Linux πλατφόρμες. Αποτελείται από τέσσερις μονάδες (a) τον *συντακτικό αναλυτή (Parser)*, που αναλύει την σύνταξη των ερωτήσεων (b) τον *κατασκευαστή γράφου (Graph Constructor)*, που συλλαμβάνει τη σημασιολογία των ερωτήσεων και τις αλληλεξαρτήσεις των εκφράσεων που περιλαμβάνονται (c) τον *SQL μεταφραστή (SQL Translator)*, που μεταφράζει την RQL σε SQL ερωτήσεις και (d) τη *μηχανή υπολογισμού (Evaluation Engine)*, που προσπελαύνει την βάση δεδομένων μέσα από SQL ερωτήσεις. Για να εγκατασταθεί και να τρέξει η RQL, απαιτούνται C++ (gcc-2.95.1 ή νεότερη έκδοση) και PostgreSQL (v7.3 ή νεότερη έκδοση) βάση δεδομένων.

Σε αυτή την ενότητα θα παρουσιάσουμε μία γρήγορη περίληψη της λειτουργικότητας της RQL και θα διευκρινίσουμε τη χρήση της μέσα από ερωτήσεις. Αποτελείται από ένα σετ από βασικές ερωτήσεις και *επαναλήπτες (iterators)*, τα οποία μπορούν να χρησιμοποιηθούν για να κατασκευάσουν πιο πολύπλοκες ερωτήσεις. Ως παράδειγμα εφαρμογής των RQL ερωτήσεων θα χρησιμοποιήσουμε μία πολιτιστική πύλη (Portal, εικόνα 2.2, το οποίο χρησιμοποιείται, μαζί με τα παραδείγματα των ερωτήσεων που θα χρησιμοποιήσουμε στην συνέχεια, στο [18]), το οποίο περιέχει περιγραφές των πόρων, όπως δικτυακούς τόπους μουσείων ή ιστοσελίδες με εκτιθόμενες δημιουργίες, τα οποία περιγράφονται σύμφωνα με τα αντίστοιχα RDF σχήματα.

Στο πάνω μέρος της εικόνας, οι μετασχήμα κλάσεις που χρησιμοποιούνται είναι οι βασικές έννοιες της RDF, Class και Property, όπως και ορισμένες από τον χρήστη μετακλάσεις (π.χ *RealWorldObject*, *WebResource* και *SchemaProperty*). Περιέχει δύο μετασχήμα ιδιότητες, συγκεκριμένα η *related*, που συνδέει κλάσεις, και η *maxCardinality*, η οποία ορίζεται στις ιδιότητες και παίρνει ακέραια τιμή. Το μεσαίο μέρος αποτελείται από δύο σχήματα, απευθυνόμενα σε ειδικούς μουσείων και διαχειριστές Portal αντίστοιχα. Το πρώτο σχήμα ορίζει κλάσεις που είναι στιγμιότυπα της μετακλάσης *RealWorldObject* και το δεύτερο της μετακλάσης *WebResource*. Το κάτω μέρος παρουσιάζει περιγραφές πόρων, που δημιουργήθηκαν για διάφορους δικτυακούς τόπους μουσείων, καθώς και έργα τέχνης που είναι διαθέσιμα στο παγκόσμιο ιστό, σύμφωνα με αυτά τα σχήματα.



Εικόνα 2.2 : Παράδειγμα RDF σχημάτων και περιγραφών πόρων για μία πολιτιστική πύλη

Καταρχάς η RQL παρέχει ερωτήσεις πάνω στα σχήματα. Μάλιστα υποστηρίζει βασικές ερωτήσεις πάνω σε σχήματα, οι οποίες διασχίζουν τις ιεραρχίες των κλάσεων και των ιδιοτήτων που ορίζονται σε ένα σχήμα. Έτσι η RQL παρέχει συναρτήσεις όπως η `subClassOf` (καλύπτοντας τις μεταβατικές υποκλάσεις) και η `subClassOf ^` (μόνο για απευθείας υποκλάσεις). Για παράδειγμα μπορούμε να εκτελέσουμε τα queries

```
subClassOf (Artist)
subClassOf ^ (Artist)
```

για να βρούμε όλες τις υποκλάσεις (μεταβατικές ή απευθείας) της κλάσης `Artist`. Ομοίως, συναρτήσεις όπως η `superClassOf` και η `superClassOf ^` επιστρέφουν μεταβατικές και απευθείας υπερ-κλάσεις.

Συναρτήσεις υπάρχουν για τις ιδιότητες των σχημάτων (π.χ subPropertyOf και subPropertyOf[^]). Για παράδειγμα, μπορούμε να ρωτήσουμε για όλες τις μεταβατικές και απευθείας υπο-ιδιότητες της ιδιότητας creates :

```
subPropertyOf (creates)
subPropertyOf^ (creates)
```

Όμοια ορίζονται συναρτήσεις superPropertyOf και superPropertyOf[^] που επιστρέφουν υπεριδιότητες. Για μία συγκεκριμένη ιδιότητα μπορούμε να βρούμε που ορίζεται, εφαρμόζοντας τις συναρτήσεις domain και range :

```
domain (creates)
range (creates)
```

Επίσης η RQL παρέχει την συνάρτηση namespace, με σκοπό να ανακτήσουμε το χώρο ονόματος οποιουδήποτε τμήματος ενός σχήματος (κλάσης, ιδιότητας, μετακλάσης κ.τ.λ). Ανακτούμε δηλαδή το URI του σχήματος στο οποίο ορίζεται. Ένα τέτοιο παράδειγμα είναι η ερώτηση:

```
namespace (ns:Artist)
USING NAMESPACE ns =
&http://139.91.183.30:9090/RDF/VRP/Examples/demo/culture.rdf#
```

Η RQL παρέχει μία select-from-where δομή επιτρέποντας την εισαγωγή μεταβλητών. Δίνοντας μας μία βάση περιγραφής δεδομένων ή σχετιζόμενα σχήματα, αυτά μπορούν να θεωρηθούν ως μία συλλογή από κόμβους και ακμές, και για αυτό η RQL (με αυτή τη δομή) μπορεί να διασχίσει επαναληπτικά γράφους της RDF σε οποιοδήποτε βάθος. Μία πρώτη περίπτωση χρήσης τέτοιων ερωτήσεων είναι η αναζήτηση σχημάτων. Για παράδειγμα, μπορούμε για μία δοσμένη ιδιότητα ενός σχήματος να βρούμε όλες τις σχετιζόμενες κλάσεις του:

Ποιες κλάσεις μπορούν να εμφανιστούν ως πεδίο και εύρος της ιδιότητας creates;

```
SELECT $C1, $C2
FROM   {$C1}paints{$C2}
```

Η απάντηση σε αυτή την ερώτηση είναι :

\$C1	\$C2
Painter	Painting
Cubist	Painting
Flemish	Painting

Στο τμήμα που περιέχεται το <from> στην πρόταση, χρησιμοποιούμε μία ερώτηση σχήματος που αποτελείται από μία ιδιότητα με το όνομα paints (η επιγραφή μίας ακμής) και δύο μεταβλητές σε κλάσεις \$C1 και \$C2 (μεταβλητές πάνω σε επιγραφές κόμβων). Η ένδειξη {} χρησιμοποιείται στις RQL εκφράσεις για να εισάγουμε κατάλληλες μεταβλητές σχήματος ή δεδομένων. Το σύμβολο \$ είναι ένα πρόθεμα για ονόματα μεταβλητών που δηλώνουν μεταβλητές πάνω σε κλάσεις του σχήματος. Το τμήμα της πρότασης που περιέχεται το select, ορίζει μία προβολή των μεταβλητών που μας ενδιαφέρουν (\$C1,\$C2 στην συγκεκριμένη περίπτωση).

Με το παρακάτω παράδειγμα, ανακτούμε όλες τις σχετιζόμενες ιδιότητες του σχήματος για μία συγκεκριμένη κλάση.
Βρείτε όλες τις ιδιότητες που εφαρμόζονται από την κλάση Painter

```
SELECT @P FROM {$X} @P WHERE $X = Painter
```

Η απάντηση στην παραπάνω ερώτηση είναι:

P
first_name
Paints
last_name
Creates

Στο <from> τμήμα της ερώτησης χρησιμοποιούμε μία άλλη έκφραση, αποτελούμενη από μία μεταβλητή σε κλάση \$X (πάνω σε επιγραφές κόμβων) και μία μεταβλητή σε ιδιότητα @P (πάνω σε επιγραφές ακμών). Στην RQL, το πρόθεμα @ χρησιμοποιείται για ονόματα μεταβλητών για να δηλώσει μεταβλητές πάνω σε ιδιότητες του σχήματος. Στο where τμήμα η συνθήκη θα φιλτράρει τις τιμές του @P για να διατηρήσει μόνο τις ιδιότητες που έχουν ως πεδίο την κλάση Painter (η ιδιότητα paints) ή υπερκλάσεις του Painter (οι ιδιότητες last_name, first_name, creates).

Οι ερωτήσεις της RQL παρέχουν τα μέσα για προσπέλαση βάσεων με RDF περιγραφές με ελάχιστη γνώση των σχημάτων που πιθανώς αναφέρονται. Μπορούμε να προσπελάσουμε οποιοδήποτε στοιχείο της RDF (κλάση, ιδιότητα, οι τιμές ενός αποδέκτη με δεδομένα της RDF κ.α.) απλά γράφοντας το όνομά του. Για παράδειγμα, η ερώτηση

```
Artist
```

επιστρέφει το σύνολο των στιγμιότυπων αυτής της κλάσης και των υποκλάσεων του. Για κατάλληλο φιλτράρισμα των δεδομένων, η RQL χρησιμοποιεί τα Boolean κατηγορήματα =, <, > και like. Αυτά μπορούν να εφαρμοστούν σε τιμές κυριολεκτικών (π.χ strings, integers, reals, dates) ή σε URIs των πόρων και σε ονόματα κλάσεων, ιδιοτήτων και μετακλάσεων. Ένα τέτοιο παράδειγμα είναι η ερώτηση :

```
Painter < Artist
```

η οποία επιστρέφει true αφού ο πρώτος τελεστής είναι υποκλάση του δεύτερου. Άλλα παραδείγματα αυτής της κατηγορίας (ερώτηση σε RDF περιγραφές) είναι τα ακόλουθα :

Βρείτε το μέγεθος αρχείου του πόρου με URL
<http://www.artchive.com/rembrandt/abraham.jpg>

```
SELECT Y  
FROM {X}file_size{Y}  
WHERE X = &http://www.artchive.com/rembrandt/abraham.jpg
```

Το σύμβολο & χρησιμοποιείται στο τμήμα με το <where> του RQL query για να περιγράψει ισότητα μεταξύ μεταβλητών και πόρων.

Βρείτε τους τίτλους των πόρων των οποίων το URL ταιριάζει με
*“*www.artchive.com*”*

```
SELECT Y
FROM {X}title{Y}
WHERE X like "*www.artchive.com*"
```

Βρείτε τους δημιουργούς και τα έργα τους

```
SELECT X, Y
FROM {X}creates.title{Y}
```

Το σύμβολο “.” που αναφέρθηκε στην ερώτηση χρησιμοποιείται για τη σύνδεση των δύο συστατικών και υποδηλώνει συνθήκη ένωσης μεταξύ του εύρους της ιδιότητας *creates* και του πεδίου της ιδιότητας *title*. Επομένως η προηγούμενη ερώτηση είναι ισοδύναμη με την ακόλουθη ερώτηση :

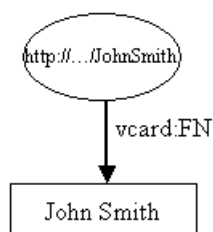
```
SELECT X, Y
FROM {X}creates{D}, {G}title{Y}
WHERE D=G
```

2.4.2 Jena 2

Το Jena [21] είναι ένα εργαλείο σε Java για εγγραφή εφαρμογών του Semantic Web, το οποίο υποστηρίζεται από το Hewlett-Packard Labs Semantic Web Research group [20]. Τα κύρια χαρακτηριστικά του είναι τα εξής:

a) Ένα RDF API

Το Jena είναι μία διασύνδεση προγραμματισμού εφαρμογών σε Java [22] (Java application programming interface ή εν συντομία API) το οποίο μπορεί να χρησιμοποιηθεί για να δημιουργήσει και να υλοποιήσει γράφους της RDF όπως ο παρακάτω:



Το Jena έχει κλάσεις για αναπαράσταση γράφων, πόρων, ιδιοτήτων και κυριολεκτικών. Οι διασυνδέσεις (interfaces) που αναπαριστούν πόρους, ιδιότητες και κυριολεκτικά καλούνται *Resource*, *Property* και *Literal* αντίστοιχα. Στο Jena, ένας γράφος καλείται μοντέλο (*model*) και αναπαριστάται με τη *Model* διασύνδεση. Παρέχει τα εξής:

- i) μέθοδοι που αφορούν προτάσεις για την υλοποίηση ενός μοντέλου της RDF ως ένα σύνολο με τριάδες της RDF
- ii) μέθοδοι που αφορούν πόρους για την υλοποίηση ενός μοντέλου της RDF ως ένα σύνολο από πόρους με ιδιότητες
- iii) μέθοδοι για την υλοποίηση αποδεκτών περιλαμβάνοντας την προσθήκη νέων μελών, την εισαγωγή νέων μελών στο μέσο ενός αποδέκτη και τη διαγραφή υπαρχόντων μελών

- iv) συντακτικούς αναλυτές για ανάγνωση και εγγραφή αρχείων της RDF σε RDF/XML, N3 και N-TRIPLES
- v) υποστήριξη εγγραφής κυριολεκτικών

b) ARP - Ο RDF/XML συντακτικός αναλυτής της Jena

Είναι ένας συντακτικός αναλυτής της RDF στηριζόμενος στην Java. Το ARP στοχεύει να είναι πλήρως συμβατό με τις τελευταίες προδιαγραφές της RDF και συνήθως επικαλείται με τη χρήση των εντολών ανάγνωσης του Jena, αλλά μπορεί να χρησιμοποιηθεί και αυτόνομα.

c) Το υποσύστημα λογισμού

Το υποσύστημα *λογισμού (reasoning)* της Jena έχει σχεδιαστεί να επιτρέπει σε μια σειρά από reasoners να συνδεθούν με την Jena. Αυτοί οι reasoners περιέχουν αξιώματα, κανόνες και προαιρετικά πληροφορίες οντολογίας, οι οποίοι εάν εφαρμοστούν σε μία βάση με RDF περιγραφές, συνεπάγονται επιπλέον ισχυρισμούς της RDF. Η κύρια χρήση αυτού του μηχανισμού είναι η υποστήριξη στην χρήση γλωσσών όπως η RDFS και η OWL [23], οι οποίες επιτρέπουν να συμπεραίνονται επιπλέον γεγονότα από τα στιγμιότυπα και τις περιγραφές των κλάσεων. Ωστόσο αυτός ο μηχανισμός είναι αρκετά γενικός, με σκοπό να μπορούν να χρησιμοποιηθούν αρκετοί τρόποι επεξεργασίας της RDF.

d) Το υποσύστημα οντολογιών

Το Jena2 API οντολογιών (Ontology API) [24] έχει ως σκοπό να υποστηρίξει τους προγραμματιστές που εργάζονται με δεδομένα οντολογίας που στηρίζονται στην RDF. Ειδικότερα, αυτό σημαίνει υποστήριξη για OWL, DAML+OIL [25] και RDFS. Ένα σύνολο από αφηρημένες κλάσεις της Java επεκτείνουν τις γενικές Resource και Property κλάσεις της RDF για να αναπαραστήσουν απευθείας τις εκφράσεις των κλάσεων και ιδιοτήτων όπως παρουσιάζονται σε αυτές τις οντολογίες χρησιμοποιώντας τις πιο πάνω γλώσσες, καθώς και τις σχέσεις μεταξύ των κλάσεων και των ιδιοτήτων. Το API οντολογιών δουλεύει στενά με το reasoning υποσύστημα, συμπεραίνοντας επιπλέον πληροφορία από μία συγκεκριμένα πηγή οντολογίας.

e) Η RDQL γλώσσα ερωτήσεων

Η RDQL [26] είναι μία γλώσσα εκτέλεσης ερωτήσεων για δεδομένα της RDF. Η RDQL είναι μια υλοποίηση της SquishQL RDF γλώσσας ερωτήσεων. Ως γνωστών, η RDF παρέχει ένα γράφο με κατευθυνόμενες ακμές, με τους κόμβους να είναι πόρους ή κυριολεκτικά και τις ακμές να είναι ιδιότητες. Η RDQL παρέχει ένα τρόπο για καθορισμό ενός σχεδίου γράφου, ο οποίος συγκρίνεται πάνω στον προηγούμενο γράφο για να παράγει ένα σύνολο από ταιριάσματα. Επιστρέφει μία λίστα από ζευγάρια ονόματος και τιμής, δηλαδή μεταβλητών και τιμών τους.

Ένα παράδειγμα RDQL ερώτησης είναι το ακόλουθο :

```
SELECT ?x, ?name
WHERE (?x <http://www.w3.org/2001/vcard-rdf/3.0#FN> ?name)
```

Σε αυτή την ερώτηση ψάχνουμε πόρους που χρησιμοποιούν την ιδιότητα `http://www.w3.org/2001/vcard-rdf/3.0#FN` καθώς και τις τιμές που παίρνει. Στις RDQL queries τα URI 's βρίσκονται ανάμεσα σε εισαγωγικά, οι μεταβλητές παρουσιάζονται με ένα "?" μπροστά τους και οι σταθερές είναι ένα αλφαριθμητικό μέσα σε εισαγωγικά ή αριθμοί χωρίς εισαγωγικά. Η τριάδα (`?x <http://www.w3.org/2001/vcard-rdf/3.0#FN> ?name`) συγκρίνεται με κάθε τριάδα του γράφου και τα αποτελέσματα συλλέγονται όλα μαζί. Τα αποτελέσματα που επιστρέφονται είναι της μορφής :

X	name
<code><http://somewhere/JohnSmith/></code>	"John Smith"
<code><http://somewhere/RebeccaSmith/></code>	"Becky Smith"
<code><http://somewhere/SarahJones/></code>	"Sarah Jones"
<code><http://somewhere/MattJones/></code>	"Matt Jones"

Η υποστήριξη του Jena για την RDQL γίνεται με την χρήση κατάλληλων κλάσεων της Java (μέρος του API). Παρέχει δηλαδή κλάσεις για την γραμματική ανάλυση (μέσω του κατάλληλου συντακτικού αναλυτή) και εκτέλεση της ερώτησης, καθώς και για την συλλογή των αποτελεσμάτων.

f) Το υποσύστημα διατήρησης

Το υποσύστημα *διατήρησης (persistence)* της Jena παρέχει μία υλοποίηση της διασύνδεσης του Jena μοντέλου αλλά με την ικανότητα για αποθήκευση και ανάκτηση των προτάσεων της RDF, χρησιμοποιώντας μία βάση δεδομένων. Υποστηρίζει MySQL [27], Oracle [28] and PostgreSQL για μόνιμη αποθήκευση και τρέχει κάτω από Linux και WindowsXP. Το υποσύστημα υποστηρίζει τη δυνατότητα *Fastpath*, σύμφωνα με την οποία οι RDQL ερωτήσεις δυναμικά μετατρέπονται σε SQL ερωτήσεις για να εκτελεστούν στη βάση δεδομένων. Επιτρέπει στους χρήστες με διάφορες επιλογές διαμόρφωσης να μειώσουν τον αποθηκευτικό χώρο που χρησιμοποιείται.

2.4.3 Διαφορές Ανάμεσα Στα Δύο Εργαλεία

Συνοψίζοντας τις διαφορές μεταξύ των δύο εργαλείων, έχουμε τα ακόλουθα:

- Το RDFSuite χρησιμοποιεί ως γλώσσα για ερωτήσεις πάνω στην RDF την RQL, ενώ το Jena χρησιμοποιεί την RDQL. Είναι δύο γλώσσες με πολλές διαφορές στην σύνταξη και στις δυνατότητες και θα αναλυθούν στην συνέχεια.
- Το Jena μπορεί να τρέξει τόσο στα windows όσο και στο linux. Εξαρτάται από το αν χρησιμοποιούμε ένα σύστημα βάσης δεδομένων στο background ή όχι. Η τελευταία έκδοση του Jena έχει δοκιμαστεί με τις κατανομές των βάσεων δεδομένων MySQL 4.0.12, PostgreSQL 7.3.3 και Oracle 9.2.0.1.0 πάνω στα Windows XP και στο RedHat Linux 7.2 (linux kernel 2.4.x). Το RDFSuite τρέχει μόνο στο linux. Έχει δοκιμαστεί στο Solaris 8 (SunOS 5.8) όπως και σε RedHat Linux 7.3, σε Mandrake Linux 8.1 και σε Debian Linux 3.0 (linux kernel 2.4.x).
- Το Jena μπορεί να τρέξει χωρίς τη χρήση μίας βάσης δεδομένων στο βάθος, απλά χρησιμοποιώντας Java κλάσεις, παρουσιάζοντας ένα ισχυρότερο API, το οποίο παρέχει τη δημιουργία προτάσεων, πόρων και ιδιοτήτων.
- Στο RDFSuite, και ειδικότερα στο RSSDB, ο χρήστης μπορεί να αποθηκεύσει ένα αρχείο, χρησιμοποιώντας οποιοδήποτε RDF σχήμα μέσα σε αυτό, αρκεί προηγουμένως να έχουν αποθηκευτεί τα αρχεία που καθορίζουν αυτά τα σχήματα. Αντίθετα, στο

Jena, δεν μπορεί να αποθηκευτεί ένα αρχείο που χρησιμοποιεί κάποιο σχήμα του πλαισίου εργασίας περιγραφής πόρων. Το μόνο που υποστηρίζει είναι η δημιουργία ενός σχήματος, μέσω των δυνατοτήτων που προσφέρει το API του.

e) Το RDFSuite δεν υποστηρίζει την χρήση των αποδεκτών του πλαισίου εργασίας περιγραφής πόρων

f) Το Jena API υποστηρίζει την χρήση της OWL και της DAML+OIL επιπλέον, καθώς και τη δυνατότητα να προσθέτουμε και να δημιουργούμε νέες οντολογίες.

Οι διαφορές ανάμεσα σε RQL και RDQL είναι :

a) Η RQL είναι μία γλώσσα για εκτέλεση ερωτήσεων πάνω σε δεδομένα του πλαισίου εργασίας περιγραφής πόρων και σχήματα, ενώ η RDQL μόνο σε δεδομένα του πλαισίου εργασίας περιγραφής πόρων. Οι RDQL ερωτήσεις δεν δίνουν την δυνατότητα να αναφερθούν π.χ. σε πόρους που ανήκουν σε συγκεκριμένες κλάσεις, σε ιδιότητες που έχουν συγκεκριμένο πεδίο ή εύρος, σε ιεραρχίες υποκλάσεων κ.τ.λ. Έχουν συγκεκριμένη δομή και χρησιμοποιούν τριάδες που περιέχουν πόρους και κυριολεκτικά.

b) Η RQL υποστηρίζει περισσότερα λογικά (*boolean*) κατηγορήματα από την RDQL, και ειδικότερα του <like>, ενός κατηγορήματος απαραίτητου για την υλοποίηση που θα παρουσιάσουμε.

c) Το RDFSuite επιστρέφει τα αποτελέσματα των ερωτήσεων σε συγκεκριμένο αρχείο με μορφή RDF/XML σύνταξης. Αντίθετα η RDQL επιστρέφει απευθείας τα αποτελέσματα στις αντίστοιχες μεταβλητές.

2.5 JDOM

Το JDOM [29] είναι μια αρκετά απλά μία αναπαράσταση ενός XML εγγράφου σε Java. Το JDOM παρέχει ένα τρόπο για αναπαράσταση του εγγράφου έτσι ώστε να επιτυγχάνεται η εύκολη και αποδοτική ανάγνωση, υλοποίηση και εγγραφή. Έχει ένα απευθείας API, το οποίο είναι γρήγορο και ελαφρύ και είναι βελτιστοποιημένο για τον προγραμματιστή της Java.

2.6 Περίληψη

Σε αυτό το κεφάλαιο παρουσιάσαμε κάποιες βασικές έννοιες, τις οποίες θα χρησιμοποιήσουμε σε αυτή τη διπλωματική, όπως τα μεταδεδομένα και η RDF. Επίσης κάναμε μία εισαγωγή στο σημασιολογικό διαδίκτυο και παρουσιάσαμε εργαλεία του σημασιολογικού διαδικτύου, τα οποία αποθηκεύουν, προσπελούν και κάνουν ερωτήσεις σε RDF περιγραφές, επειδή στο επόμενο κεφάλαιο θα παρουσιάσουμε την υλοποίηση ενός δικτύου ομοβάθμιων υπολογιστικών μονάδων, το οποίο στηρίζεται πάνω στην RDF.

Κεφάλαιο 3

Το Σύστημα RQL P2P

Σε αυτό το κεφάλαιο θα παρουσιάσουμε το δίκτυο ομοβάθμιων υπολογιστικών μονάδων που υλοποιήσαμε, το οποίο κάνει χρήση της γλώσσας RQL για να κάνει τις ερωτήσεις και θα το ονομάσουμε “RQL P2P”. Θα αναφερθούμε στην αρχιτεκτονική, στις δυνατότητες και στις υπηρεσίες που προσφέρει το σύστημα RQL P2P.

3.1 Υβριδικό Δίκτυο Ομοβάθμιων Υπολογιστικών Μονάδων

Σε αυτή τη διπλωματική υλοποιήσαμε ένα υβριδικό δίκτυο ομοβάθμιων υπολογιστικών μονάδων, όπως ακριβώς είναι κατασκευασμένο το Napster [31]. Πρόκειται για ένα δίκτυο, στο οποίο η πληροφορία που ανταλλάσσεται, ελέγχεται μέσω ενός κεντρικού διακομιστή, ενώ η ροή δεδομένων γίνεται κανονικά μεταξύ των κόμβων. Κάθε κόμβος που συμμετέχει στο σύστημα, συντονίζεται από τον κεντρικό διακομιστή, ενώ για ανταλλαγή δεδομένων, εγκαθίστανται απευθείας συνδέσεις μεταξύ των κόμβων (απεικονίζονται με πιο έντονες μαύρες γραμμές). Ο βασικός λόγος που χρησιμοποιήσαμε αυτή την τοπολογία δικτύου είναι η απλότητα στην κατασκευή και η ευκολία διαχείρισής της. Κρίναμε πως από τη στιγμή που θα χρησιμοποιήσουμε μια PostgreSQL βάση δεδομένων για την αποθήκευση μεταδεδομένων των πόρων (θα εξηγήσουμε στην ενότητα 3.2), είναι καλύτερα να χρησιμοποιηθεί αυτή η βάση δεδομένων στο υπόβαθρο ενός διακομιστή.

3.2 Δημοσίευση Πόρων

Το σύστημα RQL P2P υποστηρίζει την αποθήκευση RDF περιγραφών πόρων στη βάση δεδομένων του διακομιστή. Κάθε κόμβος δηλαδή μπορεί να στέλνει στο διακομιστή αρχεία RDF/XML με μεταδεδομένα των πόρων που παρέχει και είναι διαθέσιμα για αποθήκευση για οποιοδήποτε άλλο κόμβο στο δίκτυο. Φυσικά RDF/XML αρχεία που ορίζουν συγκεκριμένα RDF σχήματα μπορούν να σταλούν επίσης. Αυτό χρησιμεύει σε περιπτώσεις όπου ένας κόμβος θέλει να χρησιμοποιήσει RDF σχήματα, που πιθανώς δεν έχουν αποθηκευτεί στη βάση δεδομένων του διακομιστή, έτσι ώστε να μπορεί στη συνέχεια να στέλνει όχι μόνο αυτός ο κόμβος, αλλά και οποιοδήποτε άλλος, μεταδεδομένα με χρήση αυτών των σχημάτων. Πριν την αποθήκευση ενός σχήματος, θα πρέπει ο κόμβος να δηλώσει το χώρο ονόματος που αντιστοιχεί σε αυ-

τό. Το σύστημα διαχείρισης βάσεως δεδομένων του διακομιστή που χρησιμοποιήσαμε είναι PostgreSQL.

Με αυτό τον τρόπο κατορθώνουμε να υλοποιήσουμε ένα P2P δίκτυο βασισμένο στην RDF. Πρόκειται για συγκεκριμένο παράδειγμα των “P2P δικτύων βασισμένα σε σχήμα” (αναφέρθηκαν στην ενότητα 2.3) που επιτρέπουν τις πολύπλοκες και επεκτάσιμες περιγραφές των πόρων αντί για τις περιορισμένες και αμετάβλητες που συνηθίζονται στα περισσότερα P2P δίκτυα σήμερα. Επίσης παρέχουν διευκολύνσεις στην εκτέλεση πολύπλοκων ερωτήσεων πάνω στα μεταδεδομένα, αντί για το απλό ψάξιμο με βάση λέξεις-κλειδιά. Έτσι το P2P δίκτυο βασισμένο στην RDF αποτελείται από κόμβους που χρησιμοποιούν συγκεκριμένα σχήματα για να περιγράψουν το περιεχόμενό τους, αλλά τα μεταδεδομένα τους στηρίζονται σε ετερογενή σχήματα.

Μία διεργασία που γίνεται κάθε φορά που ο διακομιστής αποθηκεύει μεταδεδομένα των πόρων ενός κόμβου, είναι η ενημέρωση των πινάκων πληροφορίας που αφορά τα μεταδεδομένα που βρίσκονται στη βάση δεδομένων του. Ο λόγος που γίνεται σχετίζεται με την εκτέλεση των ερωτήσεων (θα εξηγηθεί στην ενότητα 3.3) Πρόκειται για έναν πίνακα που περιέχει πληροφορία σχήματος, με τη πρώτη γραμμή να περιέχει τα σχήματα που χρησιμοποιούνται και την άλλη τις ιδιότητες. Για παράδειγμα, ας υποθέσουμε πως ένας διακομιστής έχει αποθηκεύσει στη βάση δεδομένων του μεταδεδομένα, των οποίων η πληροφορία σχήματος έχει ως εξής :

Σχήματα	dc, lom
Ιδιότητες	dc:subject, dc:language, lom:context

Πίνακας 3.1 Πληροφορία Σχήματος των Μεταδεδομένων

Δηλαδή τα μεταδεδομένα του χρησιμοποιούν το Dublin Core σχήμα (το έχουμε γράψει εν συντομία dc) και το Learning Object Metadata σχήμα (το έχουμε γράψει εν συντομία lom) και τις ιδιότητες dc:subject, dc:language, lom:context. Να σημειώσουμε πως τα σχήματα αναγνωρίζονται μοναδικά, έχουν δηλαδή συγκεκριμένους χώρους ονομάτων που τα αναγνωρίζουν. Ας υποθέσουμε τώρα πως ένας κόμβος στέλνει στον διακομιστή τα εξής μεταδεδομένα του πλαισίου εργασίας περιγραφής πόρων :

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

  xmlns:dc="file:/home/john/diplom/rql/Rssdb/examples2/Dublin_Core.rdf#"
  "

  xmlns:lom="http://ltsc.ieee.org/2002/09/lom-base#"
  xmlns:vcard=" http://www.w3.org/2001/vcard-rdf/3.0#">

  <rdf:Description rdf:about="http://www.xyz.com/sw.html">
    <dc:title>
      Software Engineering
    </dc:title>
  <dc:creator >
    <lom:Entity>
      <vcard:fn> Helmut Balzert </vcard:fn>
    </lom:Entity>
  </rdf:Description>

</rdf:RDF>
```

Το πρώτο βήμα μετά την αποθήκευση του αρχείου στη βάση δεδομένων είναι η ανάλυσή του, ώστε να ξεχωρίσουν οι χώροι ονομάτων και τα στοιχεία που χρησιμοποιούνται. Τότε ενημερώνουμε τον πίνακα και ειδικότερα τη γραμμή που αφορά τα σχήματα, ώστε να προσθέσει σχήματα καινούργια που χρησιμοποιούνται στο αρχείο που αποθηκεύτηκε και δεν περιέχονται σε αυτόν. Το αρχείο χρησιμοποιεί τα σχήματα dc, lom και vcard, οπότε ενημερώνεται ο πίνακας του διακομιστή ότι περιέχει το vcard σχήμα. Από τα στοιχεία που βρήκαμε ότι χρησιμοποιούνται στο αρχείο, προκύπτει ότι χρησιμοποιούνται οι ιδιότητες dc:title, dc:creator, lom:Entity και vcard:fn. Κάνοντας σύγκριση με τις ιδιότητες που περιέχει ο πίνακας, προσθέτουμε και τις τέσσερις ιδιότητες σε αυτόν. Έτσι ο πίνακας γίνεται :

Σχήματα	dc, lom, vcard
Ιδιότητες	dc:subject, dc:language, lom:context, dc:title, dc:creator, lom:Entity, vcard:fn

Πίνακας 3.2 Ενημερωμένος, μετά την προσθήκη του αρχείου

Να σημειωθεί ότι τα αρχεία του πλαισίου εργασίας περιγραφής πόρων που περιέχουν μεταδεδομένα και στέλνονται στο διακομιστή από ένα κόμβο, δεν αποθηκεύονται μονάχα στη βάση δεδομένων του, αλλά και στον ίδιο το διακομιστή. Όταν ένας κόμβος αποσυνδέεται από το δίκτυο, αυτόματα διαγράφονται από τη βάση δεδομένων τα μεταδεδομένα που έχει στείλει, καθώς οι πόροι που περιγράφονται δεν είναι διαθέσιμοι για αποθήκευση από άλλους κόμβους. Για αυτό το λόγο, ο διακομιστής έχει κρατήσει τα μεταδεδομένα του (στην αρχική RDF/XML μορφή τους), έτσι ώστε όταν επανασυνδεθεί ο κόμβος, αυτόματα να τα αποθηκεύσει ξανά στη βάση δεδομένων και οι αντίστοιχοι πόροι να είναι διαθέσιμοι για αποθήκευση.

3.3 Ερωτήσεις

Όπως κάθε P2P δίκτυο, έτσι και το σύστημα RQL P2P υποστηρίζει την εκτέλεση ερωτήσεων. Ένας κόμβος μπορεί να στείλει στο διακομιστή ερώτηση γραμμένη στη γλώσσα RQL (περιγράφεται στην υποενότητα 2.3.1). Εκεί ο διακομιστής εκτελεί την RQL ερώτησή στην PostgreSQL βάση δεδομένων του και επιστρέφει τα τυχόν αποτελέσματα στον κόμβο. Να σημειωθεί ότι μια διεργασία που γίνεται από πλευράς διακομιστή, πριν εκτελεστεί η ερώτηση, είναι η ανάλυσή της και πιο συγκεκριμένα η εύρεση των σχημάτων και των ιδιοτήτων που χρησιμοποιούνται σε αυτή. Στη συνέχεια, γίνεται σύγκριση αυτών των πεδίων με τα αντίστοιχα πεδία του πίνακα του διακομιστή που περιγράψαμε στην προηγούμενη ενότητα. Εάν κάποιο σχήμα ή ιδιότητα της ερώτησης βρεθεί ότι δεν χρησιμοποιείται από τα μεταδεδομένα της βάσης δεδομένων, ο διακομιστής δεν εκτελεί καθόλου την ερώτηση και επιστρέφει κενά αποτελέσματα στον κόμβο. Αυτός ο σημασιολογικός έλεγχος δείχνει εάν ο διακομιστής μπορεί να απαντήσει μια συγκεκριμένη ερώτηση, αλλά δεν εγγυάται ότι θα επιστρέψει μη-κενά αποτελέσματα. Ας υποθέσουμε πως ένας διακομιστής έχει αποθηκεύσει στη βάση δεδομένων του μεταδεδομένα, των οποίων η πληροφορία σχήματος απεικονίζεται στον πίνακα της προηγούμενης ενότητας 3.2. και ένας κόμβος στέλνει την εξής ερώτηση:

```
SELECT X,Y FROM {X}ns:title{Y} WHERE Y like "*peer-to-peer*"
using namespace ns=&http://purl.org/dc/elements/1.1#
```

Αναλύοντας την ερώτηση, βρίσκουμε ότι χρησιμοποιεί το σχήμα dc (το Dublin Core σχήμα δηλαδή που αναγνωρίζεται από το χώρο ονόματος (<http://purl.org/dc/elements/1.1#>) και την ιδιότητα dc:title. Συγκρίνοντας με τον πίνακα 3.2, παρατηρούμε ότι το σχήμα dc και η ιδιότητα dc:title περιλαμβάνονται και στα δύο πεδία του, οπότε η ερώτηση μπορεί να απαντηθεί, εκτελείται κανονικά πάνω στη βάση δεδομένων και τα αποτελέσματα (εφόσον υπάρχουν) επιστρέφονται στο κόμβο.

3.4 Προφίλ Χρηστών Και Ειδοποιήσεις

Κάθε χρήστης μπορεί να προσθέσει ένα προφίλ ώστε να λαμβάνει *ειδοποιήσεις* (*notifications*), δηλαδή μεταδεδομένα πόρων που ανήκουν σε άλλους κόμβους του δικτύου, σύμφωνα με το προφίλ που έχει δώσει. Το προφίλ είναι μία ερώτηση σε RQL (όπως ακριβώς παρουσιάστηκε στην προηγούμενη ενότητα), με την διαφορά ότι δεν θα εκτελεστεί μονάχα μια φορά, αλλά θα εκτελεστεί και θα επιστρέψει αποτελέσματα και στο μέλλον. Ο κόμβος στέλνει το προφίλ του (την RQL ερώτηση δηλαδή) στον διακομιστή και αυτός την εκτελεί πάνω στη βάση δεδομένων (εφόσον μπορεί να απαντηθεί, κάνοντας πρώτα το σημασιολογικό έλεγχο) και επιστρέφει τις ειδοποιήσεις, τα αποτελέσματα της ερώτησης δηλαδή. Η διαφορά είναι ότι ο διακομιστής αποθηκεύει το προφίλ, ώστε να βρεθεί αν ταιριάζει με μελλοντικά μεταδεδομένα που θα προστεθούν. Τα ζευγάρια όνομα χρήστη – προφίλ αποθηκεύονται σε διαδοχικές θέσεις σε ένα πίνακα, όπου κάθε χρήστης μπορεί να έχει ένα μόνο προφίλ. Εάν δηλαδή ένας χρήστης δηλώσει ένα προφίλ, ενώ ήδη έχει ένα στον πίνακα του διακομιστή, το καινούργιο προφίλ θα αντικαταστήσει το παλιότερο.

Κάθε φορά που κάποιος κόμβος στέλνει μεταδεδομένα στο διακομιστή, αφού αποθηκευτούν στην κύρια βάση δεδομένων του, αποθηκεύεται προσωρινά σε μία άλλη βάση δεδομένων, πάνω στην οποία εκτελούνται οι ερωτήσεις-προφίλ των κόμβων. Για κάθε χρήστη, που το προφίλ του ταιριάζει με κάποιο από τα μεταδεδομένα που προστέθηκαν και είναι συνδεδεμένος στο δίκτυο, του στέλνονται αμέσως οι αντίστοιχες ειδοποιήσεις. Αν όμως δεν είναι συνδεδεμένος, τότε γίνεται χρήση των *αποθηκευμένων ειδοποιήσεων* (*stored notifications*). Δηλαδή ο διακομιστής αποθηκεύει τις ειδοποιήσεις για μεταδεδομένα καινούργια που ταιριάζουν με προφίλ χρηστών που δεν έχουν ενημερωθεί ακόμα, καθώς δεν συνδέθηκαν. Πιο συγκεκριμένα χρησιμοποιεί μία δομή δεδομένων με διάταξη όνομα χρήστη – αποθηκευμένες ειδοποιήσεις, όπου ένας χρήστης μπορεί να έχει περισσότερες από μία αποθηκευμένες ειδοποιήσεις και οι οποίες στέλνονται όταν αυτός συνδεθεί. Στη συνέχεια διαγράφονται από τον πίνακα και από τον διακομιστή.

3.5 Αποθήκευση Πόρων

Η κύρια λειτουργία ενός P2P δικτύου είναι η αποθήκευση πόρων. Σε αυτή την περίπτωση, οι κόμβοι ανταλλάσσουν δεδομένα μεταξύ τους με χρήση των sockets της Java, χωρίς να αναμειγνύεται ο διακομιστής. Υπάρχει και το ενδεχόμενο να γίνει ανταλλαγή δεδομένων μεταξύ διακομιστή και κόμβου, κάτι που θα εξηγηθεί στην επόμενη ενότητα.

3.6 Ραντεβού

Όπως αναφέραμε και προηγουμένως, το σύστημα RQL P2P δίνει τη δυνατότητα στο κόμβο να λαμβάνει ειδοποιήσεις πάνω σε πόρους, των οποίων τα μεταδεδομένα ται-

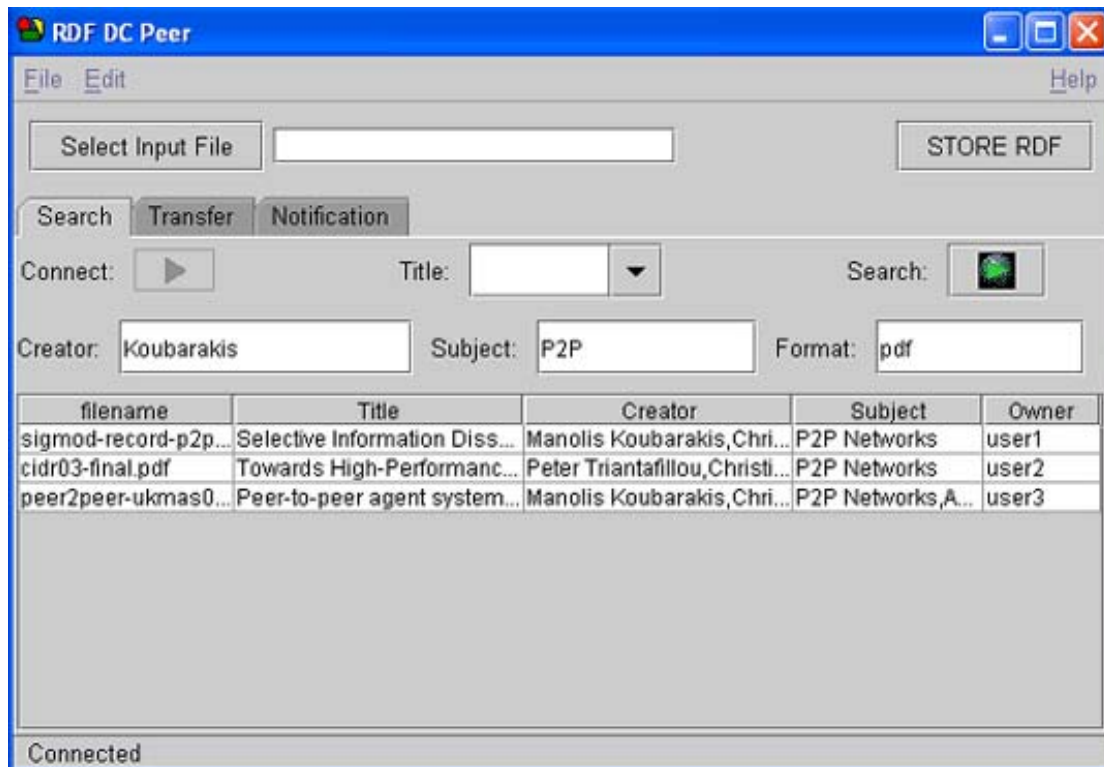
ριάζουν με το προφίλ του χρήστη, ακόμα και σε περιπτώσεις που αυτοί οι πόροι προστέθηκαν στο δίκτυο σε χρονική στιγμή που ο κόμβος δεν ήταν συνδεδεμένος στο δίκτυο (αποθηκευμένες ειδοποιήσεις). Το πρόβλημα βρίσκεται στο γεγονός ότι για να αποθηκεύσει ο κόμβος ένα πόρο, για τον οποίο έχει λάβει αποθηκευμένη ειδοποίηση, θα πρέπει ο κόμβος που τον παρέχει να είναι συνδεδεμένος με το δίκτυο. Σε αυτή τη περίπτωση θα πρέπει ο χρήστης να περιμένει να συνδεθεί αυτός ο κόμβος ή να είναι τυχερός κάποια στιγμή στο μέλλον, ώστε και οι δύο κόμβοι να είναι συγχρόνως συνδεδεμένοι με το δίκτυο. Για αυτό το λόγο, το σύστημα RQL P2P κάνει χρήση των *ραντεβού*.

Κάθε φορά που ο χρήστης θέλει να αποθηκεύσει ένα πόρο, για τον οποίο έχει λάβει ειδοποίηση, και ο κόμβος που τον παρέχει δεν είναι συνδεδεμένος εκείνη τη στιγμή με το δίκτυο, τότε ο διακομιστής ενημερώνει το χρήστη ότι δεν μπορεί να αποθηκεύσει τον πόρο. Όμως έχει τη δυνατότητα να ζητήσει ένα <ραντεβού> του διακομιστή με το συγκεκριμένο πόρο του απομακρυσμένου χρήστη. Σε αυτή την περίπτωση, μόλις συνδεθεί ο κόμβος που παρέχει το πόρο, ο διακομιστής αναλαμβάνει να τον αποθηκεύσει τοπικά. Στη συνέχεια, εφόσον ο κόμβος που ζήτησε το ραντεβού είναι συνδεδεμένος, ο διακομιστής του μεταφέρει αμέσως το συγκεκριμένο πόρο. Αλλιώς, εκτελεί την ίδια εργασία μόλις συνδεθεί ο συγκεκριμένος κόμβος στο δίκτυο.

3.7 Διασύνδεση Γραφικών Με Το Χρήστη

Στις προηγούμενες ενότητες αυτού του κεφαλαίου περιγράψαμε τις λειτουργίες και δυνατότητες που παρέχει στον χρήστη αυτό το σύστημα. Οι συγκεκριμένες λειτουργικότητες θα γίνουν αντιληπτές, μόνο εάν παρουσιαστεί ένα παράδειγμα χρήσης του συστήματος, δείχνοντας τη διασύνδεση γραφικών με το χρήστη. Όπως έχουμε πει, επιλέξαμε να χρησιμοποιήσουμε το σύστημα RQL P2P για την υλοποίηση μιας εφαρμογής μοιράσματος αρχείων. Στα πλαίσια αυτής της διπλωματικής υλοποιήσαμε ένα κόμβο με το όνομα *RDF dc peer* (περισσότερα για αυτόν θα αναφέρουμε στην ενότητα 5.2), για να ελέγξουμε τη λειτουργικότητα του συστήματος. Αυτός ο κόμβος χρησιμοποιεί μεταδεδομένα του γνωστού Dublin Core σχήματος.

Το RDF dc peer παρέχει ένα πίνακα για αναζήτηση, αφήνοντας τον χρήστη να επιλέξει όποια θέλει από τα πεδία αναζήτησης, τα οποία είναι και στοιχεία του προτύπου μεταδεδομένων του Dublin Core.

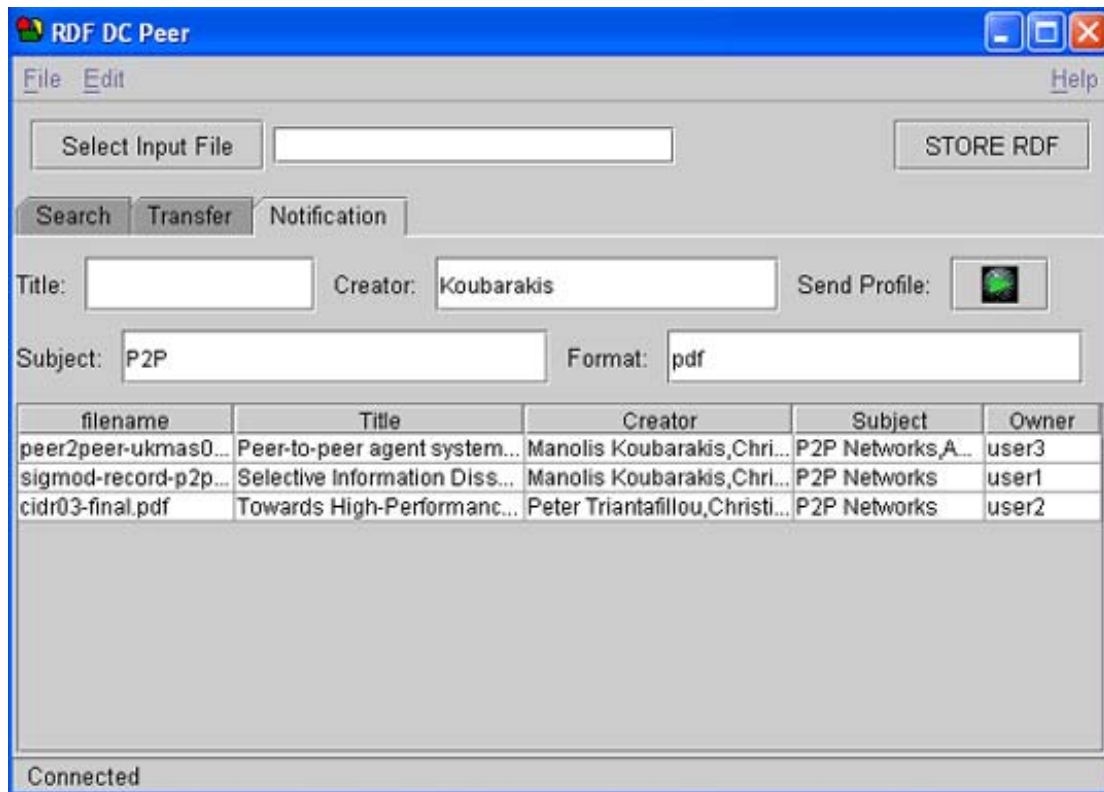


Εικόνα 3.1 : Ο πίνακας αναζήτησης του RDF dc peer

Στην εικόνα βλέπουμε ένα συγκεκριμένο παράδειγμα αναζήτησης. Δηλαδή ο χρήστης αναζητεί πόρους, που στους δημιουργούς της ανήκει ο Μανόλης Κουμπαρακής, το περιεχόμενό τους ασχολείται και με τα P2P και είναι μία εφαρμογή σε pdf. Δηλαδή, η συμπλήρωση των πεδίων αναζήτησης μεταφράζεται από το κόμβο στην RQL ερώτηση :

```
SELECT X,Y,K,S FROM {X}ns:title{Y},{X}ns:creator{K},{X}ns:subject{S},
{X}ns:format{F} WHERE K like "*Koubarakis*" AND S like "*P2P*" AND F
like "*pdf*" using namespace ns=&http://purl.org/dc/elements/1.1#
```

Αυτή η ερώτηση στέλνεται στον διακομιστή και τα αποτελέσματα που επιστρέφονται και ικανοποιούν τα πεδία αναζήτησης, φαίνονται στην εικόνα 5.2. Επίσης στην εικόνα φαίνονται οι διάφορες λειτουργικότητες του χρήστη, όπως το κουμπί που ενεργοποιεί την αναζήτηση, το κουμπί σύνδεσης στο δίκτυο, η επιλογή τοπικού αρχείου με μεταδεδομένα πόρων και η αποθήκευσή τους στο διακομιστή κ.τ.λ. Εκτός από τον πίνακα αναζήτησης, υπάρχει και ο πίνακας μεταφορών δεδομένων, που παρουσιάζει ποια αρχεία “κατεβάζει” και “ανεβάζει” ο χρήστης και ποια η εξέλιξη αυτής της μεταφοράς. Τέλος υπάρχει και ο πίνακας ειδοποιήσεων, στον οποίο ο χρήστης συμπληρώνει τα πεδία όπως και στο πίνακα αναζητήσεων, τα οποία μεταφράζονται σε RQL ερώτηση και στη συνέχεια στέλνονται οι ειδοποιήσεις-αποτελέσματα. Η διαφορά είναι πως και όταν επανασυνδεθεί ο χρήστης, δεν θα χρειάζεται να συμπληρώσει αυτά τα πεδία και να στείλει το προφίλ του, καθώς έχει ήδη καταχωρηθεί στο διακομιστή και οπότε οι ειδοποιήσεις θα συνεχίσουν να έρχονται.



Εικόνα 3.2 : Πίνακας Ειδοποιήσεων του RDF dc peer

3.8 Σενάριο Εφαρμογής

Το RQL P2P σύστημα είναι ένα υβριδικό P2P σύστημα, το οποίο χρησιμοποιήθηκε σε μία εφαρμογή μοιράσματος αρχείων. Είναι ένα σύστημα που θα μπορούσε να χρησιμοποιηθεί για υλοποίηση εφαρμογών του διαδικτύου, λειτουργώντας ως μια εφαρμογή ανταλλαγής μουσικών ή και οποιουδήποτε άλλου είδους αρχείων. Θα μπορούσε δηλαδή να λειτουργήσει αντίστοιχα με τις σημερινές γνωστές εφαρμογές ανταλλαγής αρχείων, με πιο παραπλήσιο παράδειγμα αυτό του Napster, το οποίο υπήρξε και αυτό υβριδικό P2P σύστημα.

Σε μία τέτοια περίπτωση, ένας χρήστης αποφασίζει να χτίσει μία εφαρμογή ανταλλαγής συγκεκριμένου είδους αρχείων μέσω του διαδικτύου. Τότε χρησιμοποιεί το λογισμικό που κατασκευάσαμε σε αυτή τη διπλωματική και υλοποιεί το διακομιστή. Δηλαδή λαμβάνει και τρέχει το διακομιστή στον δικό του υπολογιστή, σε συγκεκριμένη διεύθυνση διαδικτύου, και ορίζει αριθμό θύρας στον οποίο δέχεται νέες συνδέσεις πελατών-κόμβων. Παράλληλα, υλοποιεί και ένα κόμβο με συγκεκριμένη διασύνδεση γραφικών του χρήστη και με συγκεκριμένα κριτήρια αναζήτησης, τα οποία είναι μεταδεδομένα που περιγράφουν επαρκώς το είδος πόρων που ανταλλάσσονται σε αυτή την εφαρμογή. Δηλαδή στη βάση δεδομένων του διακομιστή θα αποθηκευτούν μονάχα ορισμένα RDF σχήματα, τα οποία ορίζουν τα μεταδεδομένα αυτά. Ο κόμβος αυτός θα ακολουθεί το πρωτόκολλο και τη μορφή των μηνυμάτων που ανταλλάσσονται με το διακομιστή, όπως τα έχουμε ορίσει και θα παρουσιάσουμε στο κεφάλαιο 4. Αυτό το λογισμικό υλοποίησης κόμβου θα τον διανέμει σε χρήστες μέσω κάποιου δικτυακού τόπου, στο οποίο θα παρουσιάζει αυτή την εφαρμογή P2P δικτύου. Από εκεί και μετά, κάθε φορά που κάποιος χρήστης τρέχει αυτό το λογισμικό από τον υπολο-

γιστή του και συνδέεται με το διακομιστή, αποτελεί και ένα ξεχωριστό κόμβο του P2P δικτύου.

Η περίπτωση αυτή που περιγράψαμε αντιστοιχεί σε ένα κοινό P2P δίκτυο, όπως γνωρίζουμε πως λειτουργούν σήμερα. Όμως το σύστημα RQL P2P είναι ένα δίκτυο βασισμένο σε σχήμα (για την ακρίβεια βασισμένο σε RDF), το οποίο παρέχει στους χρήστες την δυνατότητα να χρησιμοποιούν ετερογενή σχήματα και περιγραφές πόρων. Δηλαδή, ο χρήστης θα ήταν προτιμότερο να μη χτίσει μία εφαρμογή ανταλλαγής συγκεκριμένου είδους αρχείων. Αντίθετα θα παρέχει υποστήριξη για πολλά διαφορετικά RDF σχήματα, τα οποία αντιστοιχούν σε διαφορετικού είδους μεταδεδομένα. Μέσω του διαδικτυακού του τόπου θα παρέχει διαφορετικούς κόμβους, οι οποίοι θα χρησιμοποιούν διαφορετικά σχήματα και συνδυασμούς τους, ανάλογα με το είδος αρχείων που θέλουν να ανταλλάσσουν. Θα υπάρχουν κόμβοι που χρησιμοποιούν σχήματα που περιγράφουν μουσικά αρχεία, άλλοι εκπαιδευτικού περιεχομένου, κινηματογραφικού κ.α., ή και συνδυασμός αυτών.

Φυσικά θα δίνεται η δυνατότητα και σε χρήστες να δημιουργήσουν δικούς τους κόμβους, ορίζοντας το είδος των αρχείων που θέλουν να ανταλλάσσουν και φτιάχνοντας τη γραφική διασύνδεση χρήστη που αυτοί προτιμούν. Οι κόμβοι αυτοί, απλώς θα πρέπει να συνδέονται με τη διεύθυνση διαδικτύου του διακομιστή και να χρησιμοποιούν το πρωτόκολλο επικοινωνίας του RQL P2P. Ίσως θα μπορούσε να δοθεί η δυνατότητα, χρήστες που δημιουργούν δικούς τους κόμβους, να τους διανέμουν μέσω ενός διαδικτυακού τόπου. Έτσι θα μπορούσαν και οι χρήστες να φτιάξουν κόμβους που χρησιμοποιούν μεταδεδομένα από δικά τους RDF σχήματα, τα οποία τα έχουν ορίσει για να περιγράψουν πόρους, ενημερώνοντας και στέλνοντας στο διακομιστή το RDF/XML αρχείο που ορίζει αυτό το σχήμα.

3.9 Περίληψη

Σε αυτό το κεφάλαιο παρουσιάσαμε την αρχιτεκτονική του συστήματος RQL P2P. Αρχικά παρουσιάσαμε τη δομή και την τοπολογία του, και στη συνέχεια τις δυνατότητες και λειτουργίες του και ποιες είναι οι αντίστοιχες διαδικασίες και ενέργειες που συντελούνται κάθε φορά. Αναφερθήκαμε δηλαδή σε ένα υβριδικό δίκτυο ομοβάθμιων υπολογιστικών μονάδων, στο οποίο κάθε χρήστης δημοσιεύει πόρους σε RDF, εκτελεί ερωτήσεις σε RQL, στέλνει προφίλ, λαμβάνει ειδοποιήσεις, ζητάει ραντεβού και αποθηκεύει πόρους. Η γραφική διασύνδεση χρήστη του κόμβου που υλοποιήσαμε, καθώς και ένα πιθανό σενάριο χρήσης του συστήματος, κάνουν πιο κατανοητή την λειτουργικότητά του.

Κεφάλαιο 4

Τα Μηνύματα Επικοινωνίας

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα μηνύματα που χρησιμοποιούν οι κόμβοι και ο διακομιστής του συστήματος RQL P2P, για να επικοινωνήσουν μεταξύ τους. Πρόκειται για συμφωνημένη μορφή μετάδοσης δεδομένων και πληροφορίας μεταξύ διακομιστή και κόμβων. Θα παρουσιάσουμε ποια ακριβώς είναι η μορφή αυτή που υιοθετεί το δίκτυο μας και ποια είναι τα μηνύματα που ανταλλάσσουν διακομιστής και κόμβοι, ανάλογα με τη λειτουργία που θέλουν να εκτελέσουν.

4.1 Τα Μηνύματα

Η επικοινωνία μεταξύ διακομιστή και κόμβων γίνεται με τη χρήση των sockets της java, όπου εκεί ανταλλάσσουν *μηνύματα*. Κάθε μήνυμα είναι της μορφής

<μήκος><είδος><δεδομένα>

όπου τα πεδία μήκος και είδος είναι 2 bytes το καθένα. Το πεδίο μήκος καθορίζει τον αριθμό των bytes του πεδίου δεδομένα, ενώ το πεδίο είδος καθορίζει το είδος του μηνύματος. Κάθε είδος μηνύματος, το οποίο έχει και συγκεκριμένη λειτουργία, αντιστοιχεί σε συγκεκριμένο αριθμό. Το πεδίο δεδομένα είναι ένα αλφαριθμητικό ASCII που περιέχει την πληροφορία και ανάλογα με το είδος του μηνύματος, πρέπει να ακολουθεί συγκεκριμένη μορφή. Τα πεδία δηλώνονται με <>, επειδή διαχωρίζονται μεταξύ τους με ένα κενό χαρακτήρα. Τα μηνύματα επικοινωνίας θα τα διαχωρίσουμε σε δύο κατηγορίες

1. *Μηνύματα Επικοινωνίας Κόμβων – Διακομιστή*, που ορίζει τα μηνύματα που στέλνουν οι κόμβοι στο διακομιστή
2. *Μηνύματα Επικοινωνίας Διακομιστή - Κόμβων*, που ορίζει τα μηνύματα που στέλνει ο διακομιστής στους κόμβους

Στη συνέχεια θα αναφέρουμε τα όλων των ειδών τα μηνύματα, τη λειτουργία για την οποία χρησιμοποιούνται και τη μορφή που πρέπει να έχει το πεδίο δεδομένα κάθε φορά.

4.2 Μηνύματα Επικοινωνίας Κόμβων – Διακομιστή

Συγκεντρωτικά παρουσιάζονται τα μηνύματα που στέλνουν οι κόμβοι στο διακομιστή στον πίνακα 4.1.

4.2.1 Μήνυμα Νέας Σύνδεσης

μορφή : <όνομα χρήστη><κωδικός πρόσβασης><αριθμός θύρας><διεύθυνση ηλεκτρονικού ταχυδρομείου>

Το μήνυμα νέας σύνδεσης στέλνεται όταν ένας καινούργιος χρήστης θέλει να συνδεθεί για πρώτη φορά με το δίκτυο. Στην περίπτωση αυτή στέλνει τα στοιχεία του (όνομα χρήστη, κωδικός πρόσβασης, αριθμός θύρας στην οποία ανταλλάσσει δεδομένα, διεύθυνση ηλεκτρονικού ταχυδρομείου). Ο πρώτος έλεγχος που κάνει ο διακομιστής είναι εάν υπάρχει άλλος χρήστης με το ίδιο όνομα. Εάν δεν υπάρχει, τότε δημιουργείται καινούργια εγγραφή στη δομή δεδομένων του διακομιστή, η οποία περιέχει τα στοιχεία των χρηστών που έχουν συνδεθεί. Επίσης ο διακομιστής δημιουργεί καινούργιους φακέλους, έναν για την αποθήκευση των αρχείων με τα μεταδεδομένα του χρήστη που στέλνει στο διακομιστή, ένα για την αποθήκευση των αποτελεσμάτων των ερωτήσεων του και ένα για την προσωρινή αποθήκευση αρχείων που έχει ζητήσει ο χρήστης μέσω ραντεβού.

4.2.2 Μήνυμα Ελέγχου Ονόματος Χρήστη

μορφή : <όνομα χρήστη>

Το μήνυμα ελέγχου ονόματος χρήστη στέλνεται όταν ο κόμβος θέλει να ελέγξει εάν έχει δηλωθεί προηγουμένως στο διακομιστή το όνομα που θέλει να δηλώσει. Αυτός ο έλεγχος γίνεται στη δομή δεδομένων του διακομιστή, που περιέχει τα στοιχεία των χρηστών. Η χρήση αυτού του μηνύματος ουσιαστικά προορίζεται πριν τη χρήση του μηνύματος νέας σύνδεσης (υποενότητα 4.2.1), έτσι ώστε να είναι σίγουρος ο κόμβος ότι τα στοιχεία χρήστη που θα στείλει (όνομα χρήστη, κωδικός πρόσβασης κ.τ.λ.) θα αποθηκευτούν κανονικά και μια καινούργια εγγραφή θα προστεθεί στη δομή δεδομένων.

4.2.3 Μήνυμα Αναζήτησης

μορφή : <ερώτηση>

Το μήνυμα αναζήτησης στέλνεται όταν ο κόμβος αναζητεί πόρους που είναι διαθέσιμοι τη χρονική στιγμή εκείνη από άλλους κόμβους του δικτύου. Οι πόροι αυτοί θα πρέπει να ακολουθούν κάποιες συγκεκριμένες περιγραφές, οπότε αυτή η αναζήτηση μεταφράζεται από τον κόμβο σε RQL ερώτηση. Η RQL ερώτηση αυτή στέλνεται στο διακομιστή, ο οποίος ακολουθεί τη διαδικασία που περιγράφηκε στην ενότητα 3.3. Δηλαδή η ερώτηση ελέγχεται σημασιολογικά και εφόσον υποστηρίζεται, εκτελείται στην PostgreSQL βάση δεδομένων.

Μήνυμα	Περιεχόμενο
Νέας Σύνδεσης	<όνομα χρήστη><κωδικός πρόσβασης><αριθμός θύρας> <διεύθυνση ηλεκτρονικού ταχυδρομείου>
Ελέγχου Ονόματος Χρήστη	<όνομα χρήστη>
Αναζήτησης	<ερώτηση>
Δήλωσης Προφίλ	<προφίλ>
Δημοσίευσης Πόρων	<αρχείο RDF/XML>
Ορισμού Χώρου Ονόματος	<χώρος ονόματος>
Σύνδεσης	<όνομα χρήστη><κωδικός πρόσβασης>
Αίτησης Κατεβάσματος	<όνομα απομακρυσμένου χρήστη><όνομα πόρου>
Αποδεκτού Ανεβάσματος	<όνομα απομακρυσμένου χρήστη><όνομα πόρου>
Μη Διαθέσιμου Πόρου	<όνομα απομακρυσμένου χρήστη><όνομα πόρου>
Εκκινήματος Κατεβάσματος	<όνομα απομακρυσμένου χρήστη><όνομα πόρου>
Αίτησης Ραντεβού	<όνομα απομακρυσμένου χρήστη><όνομα πόρου>

Πίνακας 4.1 : Μηνύματα Επικοινωνίας Μεταξύ Κόμβων και Διακομιστή

4.2.4 Μήνυμα Δήλωσης Προφίλ

μορφή : <προφίλ>

Το μήνυμα δήλωσης προφίλ στέλνεται όταν ο κόμβος θέλει να δηλώσει ότι θέλει να RQL ερώτηση και όταν ο διακομιστής την παραλάβει, ενημερώνει τον πίνακα με ενημερώνεται για πόρους που χρησιμοποιούνται και δημοσιεύονται από άλλους κόμβους στο δίκτυο, ακόμα και όταν αυτό δεν είναι συνδεδεμένο. Αυτό το προφίλ είναι μία προφίλ των χρηστών. Η υπόλοιπη διαδικασία περιγράφεται στην ενότητα 3.4.

4.2.5 Μήνυμα Δημοσίευσης Πόρων

μορφή : <αρχείο RDF/XML>

Το μήνυμα δημοσίευσης πόρων στέλνεται όταν ο κόμβος θέλει να δηλώσει πόρους που παρέχει και είναι διαθέσιμα για αποθήκευση από οποιοδήποτε άλλο κόμβο στο δίκτυο, κάθε φορά που είναι συνδεδεμένο. Αυτό το μήνυμα παρέχει περιγραφές σε πλαίσιο εργασίας περιγραφής πόρων για οποιοδήποτε αριθμό πόρων ή ορίζει ένα σχήμα του πλαισίου εργασίας περιγραφής πόρων, ώστε να το χρησιμοποιήσει για περιγραφές πόρων του πλαισίου εργασίας περιγραφής πόρων. Όταν ο διακομιστής λάβει το μήνυμα, αναλύει συντακτικά και σημασιολογικά το αρχείο με το VRP (υποενότητα 2.3.1). Εάν είναι έγκυρο, αποθηκεύεται πρώτα στην προσωρινή PostgreSQL βάση δεδομένων, ώστε να εκτελεστούν οι ερωτήσεις-προφίλ των χρηστών και να προκύψουν καινούργιες ειδοποιήσεις, σε περιπτώσεις που βρεθεί ταίριασμα των πόρων που δημοσιεύτηκαν και των προφίλ κάποιων χρηστών. Στη συνέχεια αποθηκεύεται στην κύρια PostgreSQL βάση δεδομένων που περιέχει τις περιγραφές των πόρων των κόμβων που είναι συνδεδεμένοι. Επίσης ενημερώνεται ο πίνακας πληροφορίας σχήματος των μεταδεδομένων της βάσης δεδομένων, σε περίπτωση που στις περιγραφές χρησιμοποιούνται καινούργια σχήματα ή ιδιότητες. Εάν το αρχείο είναι περιγραφές του πλαισίου εργασίας περιγραφής πόρων, αρχικά αυτό αποθηκεύεται από το διακομιστή στο φάκελο που αντιστοιχεί σε αυτό το χρήστη και στη συνέχεια διαγράφεται από την προσωρινή βάση δεδομένων. Έτσι, όταν επανασυνδεθεί ο κόμβος, να μην χρειάζεται

να στείλει ξανά το μήνυμα δημοσίευσης πόρων για να γνωστοποιήσει τους πόρους που παρέχει στο δίκτυο, αλλά να εγκαθίστανται αυτό το αρχείο από το διακομιστή στην κύρια βάση δεδομένων.

4.2.6 Μήνυμα Ορισμού Χώρου Ονόματος

μορφή : <χώρος ονόματος>

Το μήνυμα ορισμού χώρου ονόματος στέλνεται πριν το μήνυμα δημοσίευσης πόρων (4.2.5). Ο λόγος που χρησιμοποιείται είναι για να ορίσει το χώρο ονόματος που θα αντιστοιχεί στο αρχείο που θα σταλεί αμέσως μετά και είναι χρήσιμο σε περιπτώσεις θα αποθηκευτεί σχήμα του πλαισίου εργασίας περιγραφής πόρων. Στις περιπτώσεις που θα σταλεί μήνυμα με περιγραφές του πλαισίου εργασίας περιγραφών πόρων, αρκεί να σταλεί αυτό το μήνυμα με κενό χώρο ονόματος.

4.2.7 Μήνυμα Σύνδεσης

μορφή : <όνομα χρήστη><κωδικός πρόσβασης>

Το μήνυμα σύνδεσης στέλνεται όταν ο κόμβος θέλει να συνδεθεί στο δίκτυο, χωρίς να είναι καινούργιος χρήστης. Για αυτό δηλώνει τα πεδία όνομα χρήστη και κωδικός πρόσβασης όπως ακριβώς τα είχε δηλώσει και είχε εγγραφεί έτσι όταν συνδέθηκε την πρώτη φορά. Ο διακομιστής όταν λάβει το μήνυμα, ελέγχει τη δομή δεδομένων για να ανακτήσει τα στοιχεία του χρηστή με αυτό το όνομα. Εάν βρεθεί ο χρήστης, ο κόμβος συνδέεται με το δίκτυο και ο διακομιστής αποθηκεύει στην κύρια PostgreSQL βάση δεδομένων όλα τα αρχεία που βρίσκονται στο φάκελο αυτού του χρήστη. Αναλαμβάνει δηλαδή ο διακομιστής να δημοσιεύσει τους πόρους που έχει δημοσιεύσει στο παρελθόν αυτός ο χρήστης.

4.2.8 Μήνυμα Αίτησης Κατεβάσματος

μορφή : <όνομα απομακρυσμένου χρήστη><όνομα πόρου>

Το μήνυμα αίτησης κατεβάσματος στέλνεται όταν κάποιος κόμβος θέλει να “κατεβάσει” (*download*) τον πόρο που παρέχει κάποιος άλλος κόμβος και έχει ως όνομα το δεύτερο πεδίο του μηνύματος. Θέλει δηλαδή να μεταφερθεί αντίγραφο αυτού του πόρου από τον άλλο υπολογιστή, στον οποίο βρίσκεται ο κόμβος με όνομα χρήστη το πρώτο πεδίο του μηνύματος.

4.2.9 Μήνυμα Αποδεκτού Ανεβάσματος

μορφή : <όνομα απομακρυσμένου χρήστη><όνομα πόρου>

Το μήνυμα αποδεκτού ανεβάσματος στέλνεται όταν κάποιος κόμβος επιτρέπει να “ανεβάσει” (*upload*) κάποιο πόρο που προσφέρει, με όνομα το δεύτερο πεδίο του μηνύματος, σε κάποιο άλλο κόμβο. Ακολουθεί μετά το μήνυμα αίτησης ανεβάσματος (4.3.7) που στέλνει ο διακομιστής και σημαίνει πως ο πόρος αυτός είναι διαθέσιμος για μεταφορά αντιγράφου του στον υπολογιστή, στον οποίο βρίσκεται ο κόμβος με όνομα χρήστη το πρώτο πεδίο του μηνύματος.

4.2.10 Μήνυμα Μη Διαθέσιμου Πόρου

μορφή : <όνομα απομακρυσμένου χρήστη><όνομα πόρου>

Το μήνυμα μη διαθέσιμου πόρου στέλνεται όταν κάποιος κόμβος δηλώνει ότι δεν θα να ανεβάσει κάποιο πόρο, με όνομα το δεύτερο πεδίο του μηνύματος, σε κάποιο άλλο κόμβο, με όνομα χρήστη το πρώτο πεδίο, που το ζήτησε για αποθήκευση, γιατί ο πόρος αυτός δεν υπάρχει ή δεν είναι διαθέσιμος. Ακολουθεί το μήνυμα αποτυχημένου κατεβάσματος (4.3.10) που στέλνει ο διακομιστής στο απομακρυσμένο κόμβο.

4.2.11 Μήνυμα Ξεκινήματος Κατεβάσματος

μορφή : <όνομα απομακρυσμένου χρήστη><όνομα πόρου>

Το μήνυμα ξεκινήματος κατεβάσματος στέλνεται από το κόμβο αμέσως μετά το μήνυμα αποδεκτού κατεβάσματος (4.3.9) που στέλνει ο διακομιστής. Δηλώνει πως είναι έτοιμο να αρχίσει να παραλαμβάνει δεδομένα από το κόμβο με όνομα χρήστη το πρώτο πεδίο του μηνύματος, κατεβάζοντας τον πόρο του με όνομα το δεύτερο πεδίο του μηνύματος. Περιμένει δηλαδή να ενημερωθεί το απομακρυσμένο κόμβο, ώστε να εγκατασταθεί σύνδεση μεταξύ τους με χρήση Java sockets και να γίνει η μεταφορά των δεδομένων.

4.2.12 Μήνυμα Αίτησης Ραντεβού

μορφή : <όνομα απομακρυσμένου χρήστη><όνομα πόρου>

Το μήνυμα αίτησης ραντεβού στέλνεται όταν κάποιος κόμβος ζητάει από το διακομιστή να κατεβάσει εκ μέρους του, τον πόρο με όνομα το δεύτερο πεδίο του μηνύματος, από το κόμβο του χρήστη με όνομα το πρώτο πεδίο του ονόματος.

4.3 Μηνύματα Επικοινωνίας Διακομιστή – Κόμβων

Συγκεντρωτικά παρουσιάζονται τα μηνύματα που στέλνει ο διακομιστής στους κόμβους στον πίνακα 4.2.

4.3.1 Μήνυμα Αποδεκτής Σύνδεσης

μορφή : <διεύθυνση ηλεκτρονικού ταχυδρομείου>

Το μήνυμα αποδεκτής σύνδεσης στέλνεται από το διακομιστή σε ένα κόμβο για να το ενημερώσει ότι συνδέθηκε επιτυχώς στο δίκτυο. Στέλνεται ως απάντηση του μηνύματος νέας σύνδεσης (4.2.1), αμέσως μετά την εξακρίβωση ότι το όνομα χρήστη που δήλωσε ο κόμβος δεν έχει ξαναδηλωθεί από αλλού, οπότε ο διακομιστής εισάγει καινούργια εγγραφή στη δομή δεδομένων για το χρήστη. Στέλνεται όμως και ως απάντηση του μηνύματος σύνδεσης (4.2.7), όταν ανακτηθούν από τη δομή δεδομένων τα στοιχεία του χρήστη με βάση το όνομα χρήστη που δήλωσε ο κόμβος.

Μήνυμα	Περιεχόμενο
Αποδεκτής Σύνδεσης	<διεύθυνση ηλεκτρονικού ταχυδρομείου>
Μη Καταχωρημένου Ονόματος	<>
Καταχωρημένου Ονόματος	<>
Αποτελεσμάτων Ερώτησης	<αποτέλεσμα>
Τέλους Αποτελεσμάτων Ερώτησης	<>
Ειδοποίησης	<αποτελέσματα>
Αίτησης Ανεβάσματος	<απομακρυσμένος χρήστης><όνομα πόρου>
Αποσυνδεδεμένου Χρήστη	<όνομα απομακρυσμένου χρήστη><όνομα πόρου>
Αποδεκτού Κατεβάσματος	<απομακρυσμένος χρήστης><διεύθυνση διαδικτύου χ στη><αριθμός θύρας><όνομα πόρου>
Αποτυχημένου Κατεβάσματος	<απομακρυσμένος χρήστης><όνομα πόρου>
Ξεκινήματος Ανεβάσματος	<απομακρυσμένος χρήστης><διεύθυνση διαδικτύου χ στη><αριθμός θύρας><όνομα πόρου>
Λάθους	<λάθος>
Απάντησης Ραντεβού	<απομακρυσμένος χρήστης><αριθμός θύρας> <όνομα πόρου>
Συνδεδεμένου Χρήστη	<απομακρυσμένος χρήστης><όνομα πόρου>

Πίνακας 4.2 : Μηνύματα Επικοινωνίας Μεταξύ Διακομιστή και Κόμβων

4.3.2 Μήνυμα Μη Καταχωρημένου Ονόματος

μορφή : <>

Το μήνυμα μη καταχωρημένου ονόματος είναι ένα κενό μήνυμα που στέλνεται από το διακομιστή σε ένα κόμβο, ως απάντηση του μηνύματος ελέγχου ονόματος χρήστη (4.2.2). Με αυτό το μήνυμα ο διακομιστής γνωστοποιεί στο κόμβο ότι το όνομα χρήστη που ρώτησε δεν είναι καταχωρημένο από κανένα άλλο κόμβο.

4.3.3 Μήνυμα Καταχωρημένου Ονόματος

μορφή : <>

Το μήνυμα καταχωρημένου ονόματος είναι ένα κενό μήνυμα που στέλνεται από το διακομιστή σε ένα κόμβο, ως απάντηση του μηνύματος ελέγχου ονόματος χρήστη (4.2.2), με τη διαφορά από το προηγούμενο μήνυμα ότι ο διακομιστής γνωστοποιεί ότι το όνομα χρήστη που ρώτησε, είναι ήδη καταχωρημένο από άλλο κόμβο.

4.3.4 Μήνυμα Αποτελεσμάτων Ερώτησης

μορφή : <αποτέλεσμα>

Το μήνυμα αποτελεσμάτων ερώτησης στέλνεται από τον διακομιστή ως απάντηση του μηνύματος αναζήτησης (4.2.3), στην περίπτωση που η ερώτηση που έστειλε ο κόμβος ταιριάζει με μεταδεδομένα πόρων που είναι αποθηκευμένα στην κύρια PostgreSQL βάση δεδομένων. Τότε για κάθε πόρο, που οι RDF περιγραφές του ταιριάζουν με την

ερώτηση του κόμβου, επιστρέφονται τα μεταδεδομένα του μέσω αυτού του μηνύματος, όπως ακριβώς τα ζητάει στην ερώτησή του ο κόμβος. Δηλαδή, αν π.χ. η ερώτηση ήταν

```
SELECT X,Y,Z FROM {X}ns:title{Y},{X}ns:creator{Z} WHERE Y like
"*peer-to-peer*" using namespace ns=&http://purl.org/dc/elements/1.1#
```

για κάθε πόρο που θα ταίριαζε με την ερώτηση, θα επιστρέφονταν σύμφωνα με αυτήν, το URI, ο τίτλος και ο συγγραφέας. Να σημειώσουμε πως κάθε πεδίο που επιστρέφεται ως αποτέλεσμα, στέλνεται από το διακομιστή στο κόμβο μέσα σε <>. Ένα παράδειγμα του πεδίου <δεδομένα> για το μήνυμα αποτελέσματος ερώτησης, για την RQL ερώτηση που αναφέραμε παραπάνω, είναι

```
<www.intelligence.tuc.gr/publications/cidr03-final.pdf> <Toward High-
Performance Peer-to-Peer Content and Resource Sharing Systems> <Mano-
lis Koubarakis>
```

4.3.5 Μήνυμα Τέλους Αποτελεσμάτων Ερώτησης

μορφή : <>

Το μήνυμα τέλους αποτελεσμάτων ερώτησης στέλνεται από το διακομιστή στον κόμβο για να του δηλώσει ότι στάλθηκαν όλα τα αποτελέσματα της ερώτησης (μπορεί να μην έχουν σταλεί και καθόλου επειδή δεν βρέθηκαν), την οποία έστειλε με το μήνυμα αναζήτησης (4.2.3).

4.3.6 Μήνυμα Ειδοποίησης

μορφή : <αποτελέσματα>

Το μήνυμα ειδοποίησης στέλνεται από τον διακομιστή στον κόμβο, για να του στείλει ειδοποίηση πάνω σε πόρο που ταιριάζει με το προφίλ που έχει δώσει. Το μήνυμα περιέχει τα μεταδεδομένα του πόρου που ταιριάζει με το προφίλ του κόμβου και ζητάει να του επιστραφούν ως αποτελέσματα. Να σημειώσουμε πως τα πεδία που επιστρέφονται ως αποτελέσματα, στέλνονται μέσα σε <>, όπως γίνεται με το μήνυμα αποτελεσμάτων ερώτησης (4.3.4). Όπως εξηγήθηκε και στην ενότητα 3.4, το μήνυμα αυτό μπορεί να σταλεί σε διάφορες περιπτώσεις. Μπορεί να είναι απάντηση του μηνύματος δήλωσης προφίλ (4.2.4), όταν τα μεταδεδομένα του πόρου ενός κόμβου που είναι συνδεδεμένο με το δίκτυο, ταιριάζουν με το προφίλ τη στιγμή που αυτό δηλώνεται. Επίσης μπορεί να σταλεί μετά το μήνυμα δημοσίευσης πόρων (4.2.5), όταν ο κόμβος που δήλωσε το προφίλ είναι συνδεδεμένο με το δίκτυο και ένας κόμβος δημοσιεύει ένα πόρο, του οποίου τα μεταδεδομένα ταιριάζουν με το προφίλ. Εάν δεν είναι συνδεδεμένο εκείνη τη στιγμή, το μήνυμα ειδοποίησης στέλνεται στον κόμβο μετά το μήνυμα σύνδεσης (4.2.7), ως αποθηκευμένη ειδοποίηση.

4.3.7 Μήνυμα Αίτησης Ανεβάσματος

μορφή : <απομακρυσμένος χρήστης><όνομα πόρου>

Το μήνυμα αίτησης ανεβάσματος στέλνεται από τον διακομιστή, αφού ο κόμβος, με όνομα χρήστη το πρώτο πεδίο του μηνύματος, στείλει το μήνυμα αίτησης κατεβά-

σματος (4.2.8) του πόρου, με όνομα το δεύτερο πεδίο του μηνύματος. Αναλαμβάνει δηλαδή ο διακομιστής να στείλει αυτό το μήνυμα για να ρωτήσει ένα κόμβο, εάν μπορεί να μεταφέρει αντίγραφο ενός πόρου του σε ένα άλλο απομακρυσμένο κόμβο. Επίσης ο διακομιστής στέλνει αυτό το μήνυμα, όταν ο κόμβος συνδεθεί και βρεθεί ότι κάποιος άλλος κόμβος, έχει στείλει το μήνυμα αίτησης ραντεβού (4.2.12), ζητώντας από το διακομιστή να αναλάβει να μεταφέρει αντίγραφο του πόρου με όνομα το δεύτερο πεδίο του μηνύματος, από το κόμβο που στέλνεται αυτό το μήνυμα. Στην περίπτωση αυτή, ως <απομακρυσμένος χρήστης> στο μήνυμα αναφέρεται το αλφαριθμητικό *server*, διακομιστής δηλαδή.

4.3.8 Μήνυμα Αποσυνδεδεμένου Χρήστη

μορφή : <όνομα απομακρυσμένου χρήστη><όνομα πόρου>

Το μήνυμα αποσυνδεδεμένου χρήστη στέλνεται ως απάντηση του μηνύματος αίτησης κατεβάσματος (4.2.8), καθώς ο κόμβος με όνομα χρήστη το πρώτο πεδίο του μηνύματος είναι αποσυνδεδεμένο. Δηλαδή το μήνυμα στέλνεται στον κόμβο που ήθελε να του μεταφερθεί αντίγραφο του πόρου με όνομα το δεύτερο πεδίο του μηνύματος, αλλά ο κόμβος που προσφέρει αυτό το πόρο δεν είναι συνδεδεμένο και οπότε δεν μπορεί να γίνει η ανταλλαγή.

4.3.9 Μήνυμα Αποδεκτού Κατεβάσματος

μορφή : <απομακρυσμένος χρήστης><διεύθυνση διαδικτύου χρήστη><αριθμός θύρας><όνομα πόρου>

Το μήνυμα αποδεκτού κατεβάσματος στέλνεται από τον διακομιστή σε ένα κόμβο ως απάντηση του μηνύματος αποδεκτού ανεβάσματος (4.2.9), που στέλνει ο κόμβος με όνομα χρήστη το πρώτο πεδίο του μηνύματος. Δηλαδή ο κόμβος, στο οποίο στέλνεται αυτό το μήνυμα, έχει την έγκριση να μεταφέρει ένα πόρο, με όνομα το τελευταίο πεδίο του μηνύματος, που ανήκει σε ένα απομακρυσμένο κόμβο. Ο διακομιστής περιέχει στο μήνυμα και τη διεύθυνση πρωτοκόλλου του διαδικτύου (*IP address*) και τον αριθμό θύρας ανταλλαγής δεδομένων του απομακρυσμένου κόμβου, ώστε οι δύο κόμβοι να εγκαταστήσουν σύνδεση.

4.3.10 Μήνυμα Αποτυχημένου Κατεβάσματος

μορφή : <απομακρυσμένος χρήστης><όνομα πόρου>

Το μήνυμα αποτυχημένου κατεβάσματος στέλνεται από το διακομιστή, αμέσως μετά το μήνυμα μη διαθέσιμου πόρου (4.2.10), που έχει στείλει ο κόμβος με όνομα χρήστη το πρώτο πεδίο του μηνύματος. Δηλαδή ο κόμβος, στο οποίο στέλνεται αυτό το μήνυμα, δεν μπορεί να μεταφέρει αντίγραφο του πόρου, με όνομα το δεύτερο πεδίο του μηνύματος, από το απομακρυσμένο κόμβο.

4.3.11 Μήνυμα Ξεκινήματος Ανεβάσματος

μορφή : <απομακρυσμένος χρήστης><διεύθυνση διαδικτύου χρήστη><αριθμός θύρας><όνομα πόρου>

Το μήνυμα ξεκινήματος ανεβάσματος στέλνεται από το διακομιστή, μετά την αποστολή του μηνύματος ξεκινήματος κατεβάσματος (4.2.11) που στέλνει ο κόμβος με όνομα χρήστη το πρώτο πεδίο του μηνύματος. Ο διακομιστής περιέχει σε αυτό το μήνυμα τη διεύθυνση πρωτοκόλλου του διαδικτύου και τον αριθμό θύρας ανταλλαγής δεδομένων του απομακρυσμένου κόμβου, ώστε ο κόμβος στο οποίο στέλνεται αυτό το μήνυμα, να εγκαταστήσει σύνδεση με το απομακρυσμένο κόμβο για να μεταφέρει αντίγραφο του πόρου του, με όνομα το τελευταίο πεδίο του μηνύματος.

4.3.12 Μήνυμα Λάθους

μορφή : <λάθος>

Το μήνυμα λάθους στέλνεται από το διακομιστή σε ένα κόμβο σε περιπτώσεις που κάποιο γενικό λάθος έχει γίνει. Ένα τέτοιο παράδειγμα είναι όταν στο μήνυμα σύνδεσης (4.2.7), ο κόμβος έχει δώσει όνομα χρήστη που δεν υπάρχει ή έχει δώσει λάθος κωδικό πρόσβασης. Ένα άλλο παράδειγμα είναι όταν στο μήνυμα δημοσίευσης πόρων (4.2.5), ο κόμβος στέλνει ένα RDF αρχείο με συντακτικό ή σημασιολογικό λάθος, οπότε δεν μπορεί να αποθηκευτεί.

4.3.13 Μήνυμα Απάντησης Ραντεβού

μορφή : <απομακρυσμένος χρήστης><αριθμός θύρας><όνομα πόρου>

Το μήνυμα απάντησης ραντεβού στέλνεται από τον διακομιστή σε ένα κόμβο ως απάντηση του μηνύματος αίτησης ραντεβού (4.2.12), που έχει στείλει προηγουμένως αυτός ο κόμβος. Δηλαδή ο διακομιστής ενημερώνει ότι έχει κατεβάσει το πόρο με όνομα το τρίτο πεδίο του μηνύματος, από το κόμβο με όνομα χρήστη το πρώτο πεδίο του ονόματος, όπως ακριβώς είχε ζητήσει να γίνει ο κόμβος στον οποίο στέλνεται αυτό το μήνυμα. Ο κόμβος μπορεί στη συνέχεια να κατεβάσει τον πόρο από το διακομιστή, μέσω της θύρας ανταλλαγής δεδομένων του διακομιστή που δίνεται στο δεύτερο πεδίο του μηνύματος.

4.3.14 Μήνυμα Συνδεδεμένου Χρήστη

μορφή : <απομακρυσμένος χρήστης><όνομα πόρου>

Το μήνυμα συνδεδεμένου χρήστη στέλνεται από το διακομιστή σε ένα κόμβο ως απάντηση του μηνύματος αίτησης ραντεβού (4.2.12), που έχει στείλει προηγουμένως αυτός ο κόμβος. Ο διακομιστής ενημερώνει το χρήστη ότι δεν μπορεί να ζητήσει ραντεβού για να αποθηκεύσει το πόρο, με όνομα το δεύτερο πεδίο του μηνύματος, γιατί ο κόμβος, με όνομα χρήστη το πρώτο πεδίο του μηνύματος, είναι συνδεδεμένος στο δίκτυο. Στην περίπτωση αυτή, ο χρήστης μπορεί απλά να αποθηκεύσει το πόρο εκείνη τη στιγμή, στέλνοντας το μήνυμα αίτησης κατεβάσματος 4.2.8.

4.4 Περίληψη

Σε αυτό το κεφάλαιο παρουσιάσαμε τα μηνύματα επικοινωνίας που χρησιμοποιούνται στο σύστημα RQL P2P, μεταξύ διακομιστή και των κόμβων. Στην αρχή παρουσιάσαμε ποια ακριβώς είναι τα πεδία από τα οποία αποτελείται ένα μήνυμα και με ποια σειρά το συνθέτουν. Στη συνέχεια, διαχωρίσαμε τα μηνύματα επικοινωνίας σε δύο κατηγορίες, ανάλογα με το αν είναι ο διακομιστής ή ο κόμβος, αποστολέας ή παραλήπτης. Παρουσιάσαμε όλα τα μηνύματα, την σειρά και τις περιπτώσεις στις οποίες ανταλλάσσονται αυτά, καλύπτοντας έτσι όλες τις λειτουργίες και συμπεριφορές του P2P δικτύου που υλοποιήσαμε. Ακολουθεί στο επόμενο κεφάλαιο η περιγραφή του πως υλοποιήθηκε το δίκτυο και ποια εργαλεία λογισμικού χρησιμοποιήθηκαν.

Κεφάλαιο 5

Η Υλοποίηση Του Συστήματος RQL P2P

Σε αυτό το κεφάλαιο θα παρουσιάσουμε την υλοποίηση του συστήματος RQL P2P. Δηλαδή θα αναφερθούμε στα πακέτα και εργαλεία λογισμικού που χρησιμοποιήθηκαν για τις διάφορες λειτουργίες, καθώς και στην αρχιτεκτονική του, τα τμήματα δηλαδή από τα οποία αποτελείται. Στη συνέχεια θα διαχωρίσουμε την παρουσίαση, παρουσιάζοντας πρώτα το διακομιστή και μετά το *dc peer*, που υλοποιήσαμε ως εφαρμογή για την διπλωματική αυτή.

5.1 Ο Διακομιστής

Ο διακομιστής του συστήματος RQL P2P έχει υλοποιηθεί πλήρως στην γλώσσα προγραμματισμού *Java*. Ο λόγος που χρησιμοποιήθηκε αυτή η γλώσσα, είναι το γεγονός πως το εργαλείο του σημασιολογικού διαδικτύου που χρησιμοποιήσαμε, το *ICS-FORTH RDFSuite* (υποενότητα 2.3.1), προσφέρει μεθόδους σε *Java* (*Java API*) για εισαγωγή και διαγραφή *RDF* τριάδων σε μία *PostgreSQL* βάση δεδομένων μέσω του *RSSDB*, καθώς και για εκτέλεση ερωτήσεων σε *RQL* πάνω στην *PostgreSQL* βάση δεδομένων μέσω του ερμηνευτή της *RQL*. Επίσης επιλέξαμε τη *Java* λόγω της φορητότητά της, δηλαδή της ικανότητας των προγραμμάτων της να εκτελεστούν σε οποιοδήποτε υπολογιστικό σύστημα, ανεξάρτητα από το υπολογιστικό σύστημα που αυτά αρχικά υλοποιήθηκαν.

Η επικοινωνία μεταξύ διακομιστή και κόμβου γίνεται με τη χρήση των *sockets* της *Java*. Ο διακομιστής εκτελείται σε συγκεκριμένο υπολογιστή και περιμένει μέσω του *socket* του, το οποίο είναι συνδεδεμένο με συγκεκριμένο αριθμό θύρας, για κάποιον πελάτη, που στη συγκεκριμένη περίπτωση είναι κάποιος κόμβος, να κάνει αίτηση για σύνδεση. Όταν λοιπόν κάποιος κόμβος κάνει αίτηση για σύνδεση και ο διακομιστής την αποδεχτεί, τότε κάνει χρήση των *Threads* της *Java*. Δηλαδή ένα καινούργιο *thread* δημιουργείται, το οποίο εκτελείται ανεξάρτητα από το διακομιστή και μέσω του *socket* που εγκαταστάθηκε, αναλαμβάνει να επικοινωνεί και να ανταλλάσσει μηνύματα με το συγκεκριμένο κόμβο. Κάθε κόμβος δηλαδή που συνδέεται με το δίκτυο επικοινωνεί με το διακομιστή μέσω ενός αντίστοιχου *thread*. Αυτό το *thread* θα το ονομάσουμε *περίοδο* (*session*). Η διάρκεια του είναι ίση με την διάρκεια που θα είναι συνδεδεμένο ο κόμβος.

Ο διακομιστής έχει υλοποιηθεί χρησιμοποιώντας επιπλέον τις κλάσεις του Java API του RSSDB, τις κλάσεις του Java API του διερμηνέα της RQL και τις κλάσεις του Java API του JDOM, καθώς και δύο PostgreSQL βάσεις δεδομένων.

5.1.1 Οι PostgreSQL Βάσεις Δεδομένων

Οι δύο PostgreSQL βάσεις δεδομένων που χρησιμοποιούμε είναι τα δύο εξωτερικά κομμάτια του διακομιστή του συστήματος RQL P2P. Ο ρόλος τους είναι η αποθήκευση των RDF περιγραφών των πόρων που προσφέρουν οι κόμβοι του δικτύου. Για την ακρίβεια χρησιμοποιούμε την κύρια PostgreSQL βάση δεδομένων, η οποία αποθηκεύει τα μεταδεδομένα των κόμβων που είναι συνδεδεμένα με το δίκτυο κάθε χρονική στιγμή, καθώς και την *καινούργιων μεταδεδομένων* PostgreSQL βάση δεδομένων, η οποία αποθηκεύει προσωρινά τα καινούργια μεταδεδομένα που έχει στείλει κάποιος κόμβος, μέχρι να συγκριθεί με όλα τα προφίλ των χρηστών.

5.1.2 Το Java API του RSSDB

Οι κλάσεις του Java API του RSSDB χρησιμεύουν για την αποθήκευση και διαγραφή των μεταδεδομένων, που δημοσιεύουν οι κόμβοι, στις PostgreSQL βάσεις δεδομένων. Για την ακρίβεια, κάθε περίοδος αναλαμβάνει να αποθηκεύσει μέσω του RSSDB τα μεταδεδομένα σε RDF, τα οποία δημοσιεύονται με μήνυμα από το αντίστοιχο της κόμβο, καθώς και να τα διαγράφει όταν χρειαστεί. Ο διακομιστής χρησιμοποιεί τις κλάσεις αυτές στο ξεκίνημα της λειτουργίας του, ώστε να αποθηκεύσει RDF/XML αρχεία που αντιστοιχούν σε RDF σχήματα γνωστών προτύπων μεταδεδομένων (π.χ Dublin Core, Vcard [31]), ώστε να μην χρειάζεται να κάνει αυτή τη δουλειά κάποιος κόμβος πριν στείλει RDF μεταδεδομένα που χρησιμοποιούν αυτά τα σχήματα. Επίσης ο διακομιστής αποθηκεύει τα αρχεία με τα μεταδεδομένα ενός κόμβου στην κύρια βάση δεδομένων, όταν αυτό επανασυνδεθεί

5.1.3 Το Java API Του RQL Ερμηνευτή

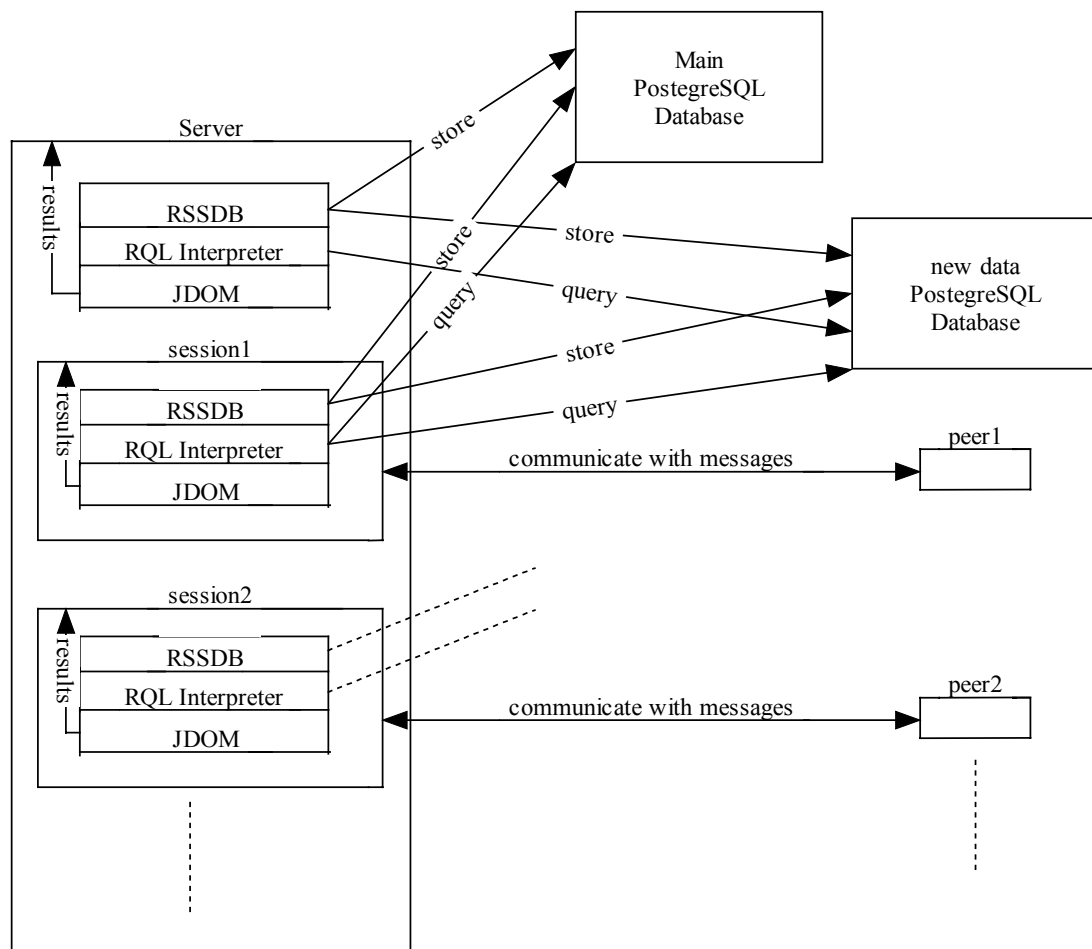
Όπως αναφέραμε στην υποενότητα 2.3.1, ο ερμηνευτής της RQL είναι υλοποιημένος σε C++. Όμως προσφέρει ένα Java API το οποίο χρησιμοποιεί το *Java Native Interface (JNI)*. Το JNI επιτρέπει σε Java κώδικα να λειτουργήσει με εφαρμογές και βιβλιοθήκες που είναι γραμμένες σε άλλες γλώσσες, όπως η C++ στην προκειμένη περίπτωση. Το Java API χρησιμοποιείται για την εκτέλεση RQL ερωτήσεων πάνω στις βάσεις δεδομένων. Για παράδειγμα, μία περίοδος το χρησιμοποιεί για εκτέλεση ερωτήσεων πάνω στην κύρια PostgreSQL βάση δεδομένων όταν ο κόμβος, της στέλνει μήνυμα αναζήτησης ή μήνυμα προφίλ. Επίσης και ο διακομιστής χρησιμοποιεί τις κλάσεις του ερμηνευτή της RQL, πάνω στην καινούργιων δεδομένων PostgreSQL βάση δεδομένων, όταν ένας κόμβος δημοσιεύει πόρους. Στην περίπτωση αυτή, ο διακομιστής εκτελεί τις ερωτήσεις-προφίλ των χρηστών που δεν είναι συνδεδεμένοι με το δίκτυο.

5.1.4 Το Java API Του JDOM

Τα αποτελέσματα των RQL ερωτήσεων που εκτελούνται μέσω του Java API του RQL ερμηνευτή επιστρέφουν τα αποτελέσματα σε αρχεία και σε RDF/XML σύνταξη. Το Java API του JDOM (αναφέρθηκε στην ενότητα 2.5), είναι πολύ χρήσιμο για τη μετατροπή των αποτελεσμάτων στη μορφή που θέλουμε (αναφέρεται στην υποενότητα 4.3.4), καθώς λαμβάνει τα attributes, τις τιμές και το περιεχόμενο αυτών των RDF/XML αρχείων-αποτελεσμάτων. Χρησιμοποιείται και για την γρήγορη εύρεση των χώρων ονομάτων και ιδιοτήτων που χρησιμοποιεί ένα αρχείο με RDF μεταδεδομένα, αφού αποθηκευτεί στις βάσεις δεδομένων, ώστε να ενημερωθεί ο πίνακας πληροφορίας σχήματος του διακομιστή.

5.1.5 Σχηματική Αναπαράσταση Του Λογισμικού

Στην συνέχεια στην εικόνα 5.1 έχουμε μία σχηματική αναπαράσταση του λογισμικού που χρησιμοποιήσαμε. Υποθέτουμε πως ένας αριθμός κόμβων είναι συνδεδεμένα με το δίκτυο, οπότε και ένας αντίστοιχος αριθμός περιόδων (sessions στο πίνακα) εκτελείται για την επικοινωνία μαζί τους. Από τον πίνακα φαίνεται πως ακριβώς χρησιμοποιούν ο διακομιστής και η κάθε περίοδος (η περίοδος1 στην εικόνα) τις διάφορες βιβλιοθήκες λογισμικού. Χρησιμοποιούμε τις διακεκομμένες γραμμές για να περιγράψουμε ότι πιθανώς μπορούν να εκτελούνται και άλλες περιόδους, οπότε να είναι συνδεδεμένα με το δίκτυο και άλλοι κόμβοι συγχρόνως.



Εικόνα 5.1 : Σχηματική αναπαράσταση του λογισμικού

5.2 Το RDF DC Peer

Όπως αναφέραμε στην ενότητα 3.2, το σύστημα RQL P2P είναι ένα P2P βασισμένο σε RDF και έτσι δεν περιορίζει τα σχήματα που μπορεί να χρησιμοποιήσει κάθε κόμβος για να περιγράψει το περιεχόμενό του. Για την ακρίβεια κάθε κόμβος μπορεί να χρησιμοποιήσει όποια σχήματα θέλει αρκεί να είναι αποθηκευμένα στις PostgreSQL βάσεις δεδομένων του διακομιστή. Επίσης κάθε κόμβος μπορεί να ορίσει τη διασύνδεση γραφικών με το χρήστη, τις RQL ερωτήσεις και το είδος των αποτελεσμάτων που θα επιστρέφει.

Σε αυτό το σημείο, για το είδος του κόμβου που θα υλοποιούσαμε, ως παράδειγμα κόμβου που θα συνδεόταν στο σύστημα RQL P2P και θα επικοινωνούσε με το διακομιστή, είχαμε δύο επιλογές : α) να υλοποιήσουμε ένα κόμβο, το οποίο στις αναζητήσεις και στα αποτελέσματα που θα επιστρέφει, θα χρησιμοποιεί μεταδεδομένα που κάνουν χρήση συγκεκριμένου ή συγκεκριμένων σχημάτων β) να υλοποιήσουμε ένα κόμβο που δεν κάνει αναζήτηση με βάση κάποιο πρότυπο μεταδεδομένων. Προτιμήσαμε την πρώτη επιλογή, για τον λόγο ότι μπορούμε με αυτόν τον τρόπο να εκτελέσουμε συγκεκριμένες και πολύπλοκες ερωτήσεις, αξιοποιώντας τις δυνατότητες της γλώσσας RQL. Για την ακρίβεια, κάνοντας χρήση των μεταδεδομένων του σχήματος, ο χρήστης μπορεί να κάνει μία αναζήτηση με βάση συγκεκριμένες περιγραφές και να του επιστραφούν μεταδεδομένα που να περιγράφουν τα αποτελέσματα. Στη δεύτερη περίπτωση, το ψάξιμο θα στηριζόταν στην κλασική αναζήτηση με λέξεις-κλειδιά, χωρίς σε καμία περίπτωση να μπορούσαμε να εκμεταλλευτούμε την περιγραφική ικανότητα της RDF.

Το κόμβο που υλοποιήσαμε το ονομάσαμε RDF dc peer, γιατί οι ερωτήσεις του εκτελούνται σε RQL, χρησιμοποιώντας μεταδεδομένα του γνωστού Dublin Core σχήματος. Το RDF dc peer είναι και αυτό υλοποιημένο πλήρως σε Java και έχει την συμπεριφορά ενός κόμβου που συνδέεται στο σύστημα RQL P2P, όπως ακριβώς το περιγράψαμε πριν. Δηλαδή συνδέεται με τον διακομιστή χρησιμοποιώντας τα sockets, καθώς και στέλνει και δέχεται μηνύματα ακολουθώντας πιστά τα πρωτόκολλα επικοινωνίας, όπως αυτά περιγράφηκαν στο κεφάλαιο 4. Να σημειωθεί πως δεν υπάρχει περιορισμός σε αυτό το κόμβο όταν δημοσιεύει πόρους, σχετικά με τα σχήματα που θα χρησιμοποιούν τα μεταδεδομένα του. Για κάθε σχήμα που θα χρησιμοποιείται, αρκεί να είναι αποθηκευμένο το αντίστοιχο RDF αρχείο στο διακομιστή. Επίσης κάθε αρχείο RDF/XML που στέλνεται στο διακομιστή θα πρέπει να προσαρμόζεται με τους περιορισμούς του RSSDB (όπως αυτοί περιγράφονται στην υποενότητα 2.3.1).

Τα πρότυπα μεταδεδομένων του Dublin Core είναι ένα απλό αλλά όμως αποδοτικό σύνολο από στοιχεία για την περιγραφή ενός ευρύ φάσματος από δικτυωμένους πόρους. Το πρότυπο του Dublin Core αποτελείται από δεκαπέντε στοιχεία, η σημασιολογία των οποίων έχει καθιερωθεί μέσω της συναίνεσης μίας ομάδας επαγγελματιών από διάφορους κλάδους, όπως από την πληροφορική, τη βιβλιοθηκονομία, τη κωδικοποίηση κειμένων κ.τ.λ. Τα μεταδεδομένα του Dublin Core μπορούν να αναπαρασταθούν σε αρκετές διαφορετικές συντάξεις, όπως αυτή της RDF/XML, που χρησιμοποιήσαμε στο RDF dc peer. Κάθε στοιχείο αναπαριστάται ως διαφορετική ιδιότητα της RDF.

Στο RDF dc peer, ο χρήστης μπορεί να κάνει αναζήτηση με βάση τέσσερα από τα στοιχεία του προτύπου των μεταδεδομένων του Dublin Core. Είναι τα εξής :

dc : title	<i>Το όνομα που δίνεται στον πόρο</i>
dc : creator	<i>Μία οντότητα κυρίως υπεύθυνη για τη δημιουργία του περιεχομένου του πόρου</i>
dc : subject	<i>Το θέμα του περιεχομένου του πόρου</i>

dc : format *Η φυσική ή ψηφιακή εκδήλωση του πόρου*

5.3 Περίληψη

Σε αυτό το κεφάλαιο παρουσιάσαμε την υλοποίηση του συστήματος RQL P2P. Διαχωρίσαμε το κεφάλαιο, παρουσιάζοντας πρώτα το διακομιστή. Αναφερθήκαμε στις βιβλιοθήκες και πακέτα λογισμικού τα οποία χρησιμοποιούνται στην υλοποίηση του και σε ποια ακριβώς λειτουργία συμμετέχουν τα Java APIs του κάθε πακέτου. Η σχηματική αναπαράσταση δείχνει ξεκάθαρα το ρόλο κάθε τμήματος. Στη συνέχεια παρουσιάσαμε το RDF dc peer που υλοποιήσαμε, ως παράδειγμα κόμβου που συνδέεται με το δίκτυό μας.

Κεφάλαιο 6

Συμπερασματικά Σχόλια

Σε αυτό το κεφάλαιο θα συνοψίσουμε ότι έχουμε κάνει σε αυτή τη διπλωματική. Σε αυτή τη διπλωματική λοιπόν υλοποιήσαμε ένα βασισμένο σε σχήμα δίκτυο ομοβάθμιων υπολογιστικών μονάδων, το οποίο χρησιμοποιεί την γλώσσα RDF για την αναπαράσταση των μεταδεδομένων των πόρων και τη γλώσσα RQL για την εκτέλεση ερωτήσεων πάνω στις RDF περιγραφές. Αυτό το δίκτυο ομοβάθμιων υπολογιστικών μονάδων συνδυάζει τις ιδιότητες ενός κοινού P2P δικτύου και ενός συστήματος ειδοποίησης συμβάντων και το χρησιμοποιήσαμε για την υλοποίηση μιας εφαρμογής μοιράσματος αρχείων.

Αρχικά, στο κεφάλαιο 2, παρουσιάσαμε τις βασικές έννοιες και τεχνολογίες που χρησιμοποιήσαμε, κάνοντας λόγο για τα μεταδεδομένα και την RDF. Στη συνέχεια κάναμε μία εισαγωγή στο σημασιολογικό διαδίκτυο και παρουσιάσαμε τα δύο εργαλεία του σημασιολογικού διαδικτύου, το ICS-FORTH RDFSuite και το Jena 2, συγκρίνοντας τα ως προς τις διάφορες δυνατότητές τους.

Στο κεφάλαιο 3 παρουσιάσαμε την αρχιτεκτονική του συστήματος RQL P2P. Κάνουμε λόγο πρώτα για την τοπολογία του και γιατί την επιλέξαμε έτσι. Αμέσως μετά αναφερθήκαμε στις λειτουργίες και στις δυνατότητες που δίνει σε κάθε χρήστη και ποιες είναι οι ενέργειες που συντελούνται από το σύστημα για κάθε λειτουργία του. Πρόκειται για ένα υβριδικό δίκτυο ομοβάθμιων υπολογιστικών μονάδων, στο οποίο κάθε χρήστης δημοσιεύει πόρους σε RDF, εκτελεί ερωτήσεις σε RQL, στέλνει προφίλ, λαμβάνει ειδοποιήσεις, ζητάει ραντεβού και αποθηκεύει πόρους. Για να γίνουν περισσότερο κατανοητές όλες αυτές οι έννοιες, στο τέλος αυτού του κεφαλαίου παρουσιάζουμε τη γραφική διασύνδεση χρήστη του κόμβου που υλοποιήσαμε και το πώς εφαρμόζονται οι διάφορες λειτουργικότητες του RQL P2P. Επίσης αναφερόμαστε σε ένα πιθανό σενάριο χρήσης αυτού του συστήματος σε P2P εφαρμογή του διαδικτύου.

Στο κεφάλαιο 4 παρουσιάσαμε τα μηνύματα επικοινωνίας που χρησιμοποιούνται στο σύστημα RQL P2P, μεταξύ διακομιστή και των κόμβων. Τα μηνύματα επικοινωνίας τα διαχωρίσαμε σε δύο κατηγορίες, ανάλογα με το αν είναι ο διακομιστής ή ο κόμβος, αποστολέας ή παραλήπτης. Παρουσιάσαμε όλα τα μηνύματα, την σειρά και τις περιπτώσεις στις οποίες ανταλλάσσονται αυτά, καλύπτοντας έτσι όλες τις λειτουργίες και συμπεριφορές του P2P δικτύου που υλοποιήσαμε.

Τέλος, στο κεφάλαιο 5 παρουσιάσαμε την υλοποίηση του συστήματος RQL P2P. Αρχικά αναφερθήκαμε στο πως ακριβώς υλοποιήθηκε ο διακομιστής, δηλαδή ποιες βιβλιοθήκες και πακέτα λογισμικού χρησιμοποιήθηκαν. Στη συνέχεια παρουσιάστηκε η υλοποίηση του κόμβου που χρησιμοποιήθηκε ως εφαρμογή. Δηλαδή κατασκευά-

σαμε ένα κόμβο που χρησιμοποιεί το Dublin Core πρότυπο μεταδεδομένων στις αναζητήσεις και στα αποτελέσματα που δίνει.

Βιβλιογραφία

- [1] Resource Description Framework (RDF) Specification by the W3C.
<http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/>
- [2] RDF Vocabulary Description Language 1.0: RDF Schema by the W3C.
<http://www.w3.org/TR/2004/REC-rdf-schema-20040210/>
- [3] World Wide Web (W3C) Consortium Home Page. <http://www.w3.org/>
- [4] Extensible Markup Language (XML) 1.0 (Third Edition) by the W3C.
<http://www.w3.org/TR/2004/REC-xml-20040204/>
- [5] The Freenet Project Home Page. <http://freenet.sourceforge.net>
- [6] Kazaa Home Page. <http://www.kazaa.com>
- [7] Gnutella Home Page. <http://www.gnutella.com>
- [8] Learning Object Metadata (LOM) website. <http://ltsc.ieee.org/wg12>
- [9] IMS Metadata Standard (IMS). <http://www.imsglobal.org>
- [10] Edutella Open Source Project. <http://edutella.jxta.org>
- [11] The ICS-FORTH RDFSuite. <http://139.91.183.30:9090/RDF>
- [12] The Jena Semantic Web Toolkit. <http://www.hpl.hp.com/semweb/jena.htm>
- [13] The RDF Schema Specific DataBase (RSSDB).
<http://139.91.183.30:9090/RDF/RSSDB/index.html>
- [14] PostgreSQL Home Page. <http://www.postgresql.org>
- [15] Java Technology. <http://java.sun.com>
- [16] The ICS-FORTH Validating RDF Parser (VRP).
<http://athena.ics.forth.gr:9090/RDF/VRP/index.html>
- [17] The RDF Query Language (RQL).
<http://athena.ics.forth.gr:9090/RDF/RQL/index.html>

- [18] RQL v2.0 User Manual. <http://athena.ics.forth.gr:9090/RDF/RQL/Manual.html>
- [19] Stroustrup: C++. <http://www.research.att.com/~bs/C++.html>
- [20] HP Labs Semantic Web Research group. <http://www.hpl.hp.com/semweb>
- [21] Jena Semantic Web Framework. <http://jena.sourceforge.net>
- [22] An Introduction to RDF and the Jena RDF API.
http://jena.sourceforge.net/tutorial/RDF_API/index.html
- [23] OWL Web Ontology Language Guide by W3C.
<http://www.w3.org/TR/2004/REC-owl-guide-20040210>
- [24] The Jena 2 Ontology API. <http://jena.sourceforge.net/ontology/index.html>
- [25] DAML+OIL Reference Description by W3C. <http://www.w3.org/TR/daml+oil-reference>
- [26] RDQL Tutorial. <http://jena.sourceforge.net/tutorial/RDQL/index.html>
- [27] Open Source Database MySQL. <http://www.mysql.com>
- [28] Oracle Technologies Home. <http://www.oracle.com/technologies>
- [29] The open source Java-based solution JDOM. <http://www.jdom.org>
- [30] DCMI Documents. <http://dublincore.org/documents/>
- [31] Napster website. <http://www.napster.com>.
- [32] ICQ Home Page. <http://www.icq.com>
- [33] Groove Website. <http://www.groove.net>
- [34] Edutella whitepaper. <http://www2002.org/CDROM/refereed/597/index.html>
- [35] S. Idreos, M. Koubarakis and C. Tryfonopoulos, P2P-DIET: An Extensible P2P Service that Unifies Ad-hoc and Continuous Querying in Super-peer Networks, Proceedings of the ACM SIGMOD/PODS 2004 Conference. Maison de la Chimie, Paris, France, June 13-18, 2004. <http://www.intelligence.tuc.gr/publications/sigmod-pods-2004-paris.pdf>
- [36] S. Idreos and M. Koubarakis, P2P-DIET: Ad-hoc and Continuous Queries in Peer-to-Peer Networks using Mobile Agents, 3rd Hellenic Conference in Artificial Intelligence, Samos, Greece, May 5-8, 2004. In LNAI, Vol. 3025, pages 23-32. <http://www.intelligence.tuc.gr/publications/SETN2004.pdf>

[37] Stratos Idreos, Manolis Koubarakis and Christos Tryfonopoulos. P2P-DIET: One-Time and Continuous Queries in Super-peer Networks. Proceedings of the IX International Conference on Extending Database Technology (EDBT04), Heraklion, Crete, Greece, March 14-18, 2004. In LNCS, Vol. 2992, pages 851-853. <http://www.intelligence.tuc.gr/publications/EDBT%202004.pdf>

[38] S. Idreos, C. Tryfonopoulos, M. Koubarakis and Y. Drougas, Query Processing in Super-Peer Networks with Languages Based on Information Retrieval: the P2P-DIET Approach, Proceedings of International Workshop on Peer-to-Peer Computing and DataBases (P2P&DB 2004), Heraklion, Crete, Greece, March 14, 2004. In LNCS (forthcoming). <http://www.intelligence.tuc.gr/publications/p2p-db2004.pdf>

[39] Nejd, Wolfgang; Wolpers, Martin; Siberski, Wolf; Schmitz, Christoph; Schlosser, Mario; Brunkhorst, Ingo; Löser, Alexander. Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. Technical Report, November 2002, accepted for Publication at 12th International World Wide Web Conference (WWW2003), Budapest, Hungary, May 2003. http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2002/www2003_superpeer.pdf

[40] Wolfgang Nejd and Wolf Siberski and Michael Sintek: Design Issues and Challenges for RDF- and Schema-Based Peer-to-Peer Systems. In: SIGMOD Record, September 2003, Special Issue on Peer-to-Peer Data Management. <http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2003/sigmod03-final.pdf>

[41] DIET Home Page. <http://www.dfki.de/diet>.