



ΕΘΝΙΚΟ ΚΑΙ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Ανάκτηση Σημασιολογικής Πληροφορίας για
Υπηρεσίες του Παγκόσμιου Ιστού
Χρησιμοποιώντας Κατανεμημένους Πίνακες
Κατακερματισμού**

Σπυριδούλα Κ. Μηλιαράκη

Επιβλέπων: Αλέξιος Δελής, Αναπληρωτής Καθηγητής Ε.Κ.Π.Α.

**ΑΘΗΝΑ
ΝΟΕΜΒΡΙΟΣ 2005**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

**Ανάκτηση Σημασιολογικής Πληροφορίας για Υπηρεσίες του
Παγκόσμιου Ιστού Χρησιμοποιώντας Κατανεμημένους
Πίνακες Κατακερματισμού**

**Σπυριδούλα Κ. Μηλιαράκη
Α.Μ. Μ591**

Επιβλέπων:

Αλέξιος Δελής, Αναπληρωτής Καθηγητής Ε.Κ.Π.Α.

Εξεταστική Επιτροπή:

Αλέξιος Δελής, Αναπληρωτής Καθηγητής Ε.Κ.Π.Α.

Εμμανουήλ Κουμπάρκης, Αναπληρωτής Καθηγητής Ε.Κ.Π.Α.

Ημερομηνία εξέτασης 4/11/2005

Περίληψη

Παρουσιάζουμε το σύστημα ομότιμων κόμβων Atlas που χρησιμοποιείται για την αποθήκευση, την ενημέρωση και την αποτίμηση επερωτήσεων RDF δεδομένων. Το σύστημα Atlas έχει υλοποιηθεί χρησιμοποιώντας τον κατακερματισμένο πίνακα κατακερματισμού Bamboo [80]. Οι κόμβοι του Atlas αποθηκεύουν δεδομένα RDF(S) σε ένα δίκτυο Bamboo και θέτουν επερωτήσεις, παρόμοιες με τις επερωτήσεις της γλώσσας RQL, με στόχο να ανακτήσουν κάποια από αυτά τα δεδομένα. Στην τυπική εφαρμογή του Atlas που οραματιζόμαστε, τα δεδομένα RDF(S) θα χρησιμοποιούνται για να περιγράψουν πόρους που ανήκουν σε κόμβους του δικτύου (π.χ. υπηρεσίες). Κάθε φορά που ένας κόμβος θέτει μια επερώτηση, οι υπόλοιποι κόμβοι του δικτύου συνεργάζονται για να βρουν τα δεδομένα RDF(S) που σχηματίζουν την απάντηση σε αυτή. Περιγράφουμε την τρέχουσα υλοποίηση του συστήματος, καθώς και ένα σενάριο εφαρμογής του Atlas για την ανακάλυψη υπηρεσιών. Επιπλέον, η διπλωματική περιλαμβάνει μια ανασκόπηση της σχετικής έρευνας στις περιοχές όπου αναφέρεται η εργασία. Συγκεκριμένα, αρχικά γίνεται μια παρουσίαση των τεχνολογιών των δικτύων ομότιμων κόμβων και του πλαισίου RDF. Έπειτα, παρουσιάζονται οι εργασίες που αφορούν στην διαχείριση RDF δεδομένων σε δίκτυα ομότιμων κόμβων. Τέλος, παρουσιάζονται εργασίες σχετικές με την ανακάλυψη και το ταίριασμα υπηρεσιών στα πλαίσια του Σημασιολογικού Ιστού αλλά και στην περιοχή του Πλέγματος.

Επιστημονική περιοχή: Κατακερματισμένα Συστήματα

Λέξεις κλειδιά: δίκτυα ομότιμων κόμβων, κατακερματισμένοι πίνακες κατακερματισμού, RDF, Σημασιολογικός Ιστός, υπηρεσίες του παγκόσμιου ιστού

Abstract

We present Atlas, a peer-to-peer system for the distributed storage, updating and querying of RDF(S) metadata. Atlas is implemented on top of the Bamboo distributed hash table [80]. Atlas nodes store RDF(S) data in a Bamboo network and pose queries, similar to RQL queries, in order to retrieve these data. In the typical application we envision, the RDF(S) data will be used to describe resources that belong to the network nodes (e.g. services). Each time a node poses a query, the network nodes cooperate to find RDF(S) data that form the answer to the query. We describe the status of the current implementation of the system Atlas and describe a scenario where Atlas is used for the discovery of services. Moreover, this thesis includes a survey of the related work in this area. Firstly, we describe the technologies that made our work possible, particularly the technology of peer-to-peer networks and the RDF data model. Then, we present a comprehensive survey of the work in the management of RDF data in peer-to-peer systems. Finally, we present work in the area of discovery and matchmaking of Web services both in the context of the Semantic Web and in Grid environments.

SUBJECT AREA: Distributed Systems

KEYWORDS: peer-to-peer networks, distributed hash tables, RDF, Semantic Web, Web services

Στον μικρό μου αδερφό, Στέλιο.

Ευχαριστίες

Αρχικά θέλω να ευχαριστήσω τον επιβλέποντα Καθηγητή κ.Αλέξη Δελή. Έπειτα, θέλω να ευχαριστήσω ιδιαίτερα τον κ. Μανόλη Κουμπάρακη χωρίς την βοήθεια του οποίου η ολοκλήρωση αυτής της διπλωματικής δεν θα ήταν δυνατή. Επιπλέον, ευχαριστώ θερμά την Ματούλα Μαγγιρίδου που μου επέτρεψε να χρησιμοποιήσω την συνοπτική της παρουσίαση για το πλαίσιο RDF (Ενότητα 2.5) και την Εριέττα Λιάρου που μου επέτρεψε να χρησιμοποιήσω την παρουσίαση της για τα δίκτυα ομότιμων κόμβων που χρησιμοποιούν RDF δεδομένα (Ενότητα 3.1). Τέλος, θέλω να ευχαριστήσω ιδιαίτερα την Ζωή Καούδη για την επικοινωνιακή και ευχάριστη συνεργασία μας κατά την δημιουργία του συστήματος Atlas καθώς και για την συμπαράσταση της κατά της διάρκεια της διπλωματικής εργασίας.

Περιεχόμενα

1	Εισαγωγή	15
1.1	Οργάνωση του τόμου	15
2	Θεωρητικό Υπόβαθρο	17
2.1	Δίκτυα Ομότιμων Κόμβων και Κατανεμημένοι Πίνακες Κατακερματισμού: Μία σύντομη εισαγωγή	17
2.2	Τρία ισχυρά συστήματα δικτύων ομότιμων κόμβων: Napster, Gnutella και Freenet	18
2.3	Ιεραρχικά Δίκτυα Ομότιμων Κόμβων	19
2.4	Κατανεμημένοι Πίνακες Κατακερματισμού	19
2.4.1	Chord	20
2.4.2	Pastry	23
2.4.3	Bamboo	24
2.5	Το πλαίσιο RDF	25
2.6	Γλώσσες επερωτήσεων RDF	28
2.6.1	RQL	28
2.6.2	RDQL	30
2.6.3	SPARQL	30
2.6.4	N3	31
2.6.5	SeRQL	32
2.6.6	TRIPLE	32
2.6.7	QEL	33
2.6.8	rdfDB	33
2.6.9	Versa	33
2.7	Συμπεράσματα	34
3	Σχετικές εργασίες	35
3.1	Αποθήκευση και Υποβολή Επερωτήσεων RDF Δεδομένων σε Δίκτυα Ομότιμων Κόμβων	35
3.2	Ανακάλυψη και Ταίριασμα Υπηρεσιών του Παγκόσμιου Ιστού	44
3.3	Ανακάλυψη πόρων και υπηρεσιών σε Πλέγματα	52

3.3.1 Ανακάλυψη στα συστήματα Condor και Globus	55
3.3.2 Σημειολογικές τεχνικές	57
3.4 Συμπεράσματα	60
4 Atlas: Αποθήκευση, ενημέρωση και αποτίμηση επερωτήσεων RDF(S) δεδομένων χρησιμοποιώντας το ΚΠΚ Bamboο	63
4.1 Η αρχιτεκτονική του συστήματος Atlas	63
4.1.1 Η επιλογή του ΚΠΚ Bamboο	64
4.2 Μοντέλο Δεδομένων και Γλώσσα Επερωτήσεων	65
4.2.1 Επερωτήσεις σύζευξης	66
4.2.2 Επερωτήσεις διάζευξης	67
4.3 Πρωτόκολλα	68
4.3.1 Σύνδεση και αποχώρηση από το δίκτυο του Atlas	68
4.3.2 Αποθηκεύοντας RDF δεδομένα	70
4.3.3 Αποτίμηση επερωτήσεων	73
4.3.4 Πιθανές βελτιστοποιήσεις	77
4.4 Λεπτομέρειες υλοποίησης Atlas v0.5	78
4.4.1 Αρχιτεκτονική ενός κόμβου του Atlas	78
4.4.2 Atlas API	80
4.5 Κατάσταση υλοποίησης	82
4.5.1 Επερωτήσεις με περιορισμούς στις μεταβλητές	83
4.5.2 RDFS	84
4.5.3 Υποστήριξη για RQL και RUL	84
4.5.4 Δημοσίευση / Συνδρομή	85
4.6 Συμπεράσματα	85
5 Ο ρόλος του Atlas στο πρόγραμμα OntoGrid	87
5.1 Το Atlas και η Υπηρεσία Μεταδεδομένων	87
5.2 Αποθήκευση, Αποτίμηση Επερωτήσεων και Ενημέρωση Μεταδεδομένων	88
5.3 Σενάρια	90
5.3.1 Χρησιμοποιώντας το Atlas για ανακάλυψη υπηρεσιών	91
5.4 Άλλα θέματα	92
5.5 Συμπεράσματα	92
6 Επίλογος	93
6.1 Σύνοψη και Συμπεράσματα	93
6.2 Μελλοντική Εργασία	93
Βιβλιογραφία	99

Κατάλογος Σχημάτων

2.1	Παράδειγμα ενός δακτύλιου αναγνωριστικών με $m=6$ και 10 κόμβους	21
2.2	RDFS σχήμα για υπηρεσίες του παγκόσμιου ιστού	28
4.1	Αρχιτεκτονική του Atlas	64
4.2	Σχήμα RDFS για υπηρεσίες του παγκοσμίου ιστού	67
4.3	Αποθηκεύοντας ένα έγγραφο RDF	71
4.4	Αρχιτεκτονική ενός κόμβου του Atlas	79
5.1	Υπηρεσία μεταδεδομένων	88
5.2	Χρησιμοποιώντας το Atlas για ανακάλυψη υπηρεσιών	90
5.3	Σχήμα RDFS για υπηρεσίες παγκόσμιου ιστού	91

Κατάλογος Πινάκων

3.1	Δίκτυα ομότιμων κόμβων βασισμένα στο πλαίσιο RDF	43
3.2	Κεντριοποιημένες τεχνικές για ανακάλυψη υπηρεσιών του παγκόσμιου ιστού	53
3.3	Κατανεμημένες τεχνικές για ανακάλυψη υπηρεσιών του παγκόσμιου ιστού	54
3.4	Σημασιολογική ανακάλυψη πόρων και υπηρεσιών στο Πλέγμα	61
4.1	Κατάσταση της υλοποίησης του Atlas	82

Κεφάλαιο 1

Εισαγωγή

1.1 Οργάνωση του τόμου

Η εργασία αυτή αποτελείται από 6 κεφάλαια.

- Στο Κεφάλαιο 2 παρουσιάζεται η τεχνολογία των δικτύων ομότιμων κόμβων, γίνεται μια σύντομη περιγραφή των γνωστότερων συστημάτων καθώς και του συστήματος Bamboo το οποίο χρησιμοποιήσαμε για την υλοποίηση του συστήματος Atlas. Επιπλέον περιγράφουμε το πλαίσιο RDF καθώς και μια σύντομη περιγραφή γλωσσών επερωτήσεων για δεδομένα RDF.
- Στο Κεφάλαιο 3 γίνεται μια ανασκόπηση των σχετικών εργασιών σε συγγενείς περιοχές της περιοχής της εργασίας. Συγκεκριμένα, αναφερόμαστε σε εργασίες που χρησιμοποιούν δίκτυα ομότιμων κόμβων για RDF δεδομένα, εργασίες για ανακάλυψη και ταίριασμα υπηρεσιών του παγκόσμιου ιστού και τέλος για ανακάλυψη πόρων και υπηρεσιών στο Πλέγμα.
- Στο Κεφάλαιο 4 παρουσιάζεται λεπτομερώς το σύστημα Atlas.
- Στο Κεφάλαιο 5 περιγράφεται πως χρησιμοποιείται το σύστημα Atlas στα πλαίσια του προγράμματος OntoGrid.
- Στο Κεφάλαιο 6 γίνεται μια σύνοψη των όσων αναφέρθηκαν και αναφέρονται οι δυνατότητες μελλοντικής επέκτασης αυτής της εργασίας.

Τέλος, ακολουθεί η ορολογία που χρησιμοποιήθηκε, ο πίνακας συντμήσεων-αρκτικόλεξων και η σχετική βιβλιογραφία.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

Σε αυτό το κεφάλαιο, θα αναφερθούμε στην τεχνολογία που κατέστησε δυνατή αυτήν την εργασία. Συγκεκριμένα, στην τεχνολογία των *δικτύων ομότιμων κόμβων* (peer-to-peer networks) και ειδικά στους Κατακερματισμένους Πίνακες Κατακερματισμού (Distributed Hash Tables) και στο *πλαίσιο RDF*. Η περιγραφή του πλαισίου RDF γίνεται μαζί με τις γλώσσες επερωτήσεων που το συνοδεύουν. Το μοντέλο δεδομένων του RDF είναι αυτό που χρησιμοποιείται για την περιγραφή των δεδομένων που αποθηκεύονται στο σύστημα Atlas.

2.1 Δίκτυα Ομότιμων Κόμβων και Κατακερματισμένοι Πίνακες Κατακερματισμού: Μία σύντομη εισαγωγή

Στα δίκτυα ομότιμων κόμβων ένας μεγάλος αριθμός αυτόνομων υπολογιστικών κόμβων (peers) συγκεντρώνουν τους πόρους τους και βασίζονται ο ένας στον άλλον τόσο για δεδομένα όσο και για υπηρεσίες. Τα δίκτυα ομότιμων κόμβων έχουν εμφανιστεί σαν ένας φυσικός τρόπος διαμοιρασμού δεδομένων. Γνωστά συστήματα όπως το Napster¹ (πλέον παρέχεται ως εμπορική υπηρεσία), το Gnutella², το Freenet³, το Kazaa⁴, το Morpheus⁵ και άλλα συστήματα έχουν κάνει δημοφιλές το μοντέλο αυτό. Ιδέες από τα δίκτυα ομότιμων κόμβων μπορούν να εφαρμοστούν τόσο για τον απλό διαμοιρασμό δεδομένων όσο και για άλλες κατακερματισμένες εφαρμογές όπως υπολογισμοί σε Πλέγμα (Grid computation) (π.χ., οι εφαρμογές

¹<http://www.napster.com>

²Υπάρχουν διάφοροι πελάτες που υλοποιούν το πρωτόκολλο Gnutella ή παραλλαγές του. Για παράδειγμα βλέπε <http://www.limewire.com>.

³<http://freenet.sourceforge.net>

⁴<http://www.kazaa.com>

⁵<http://www.musiccity.com>

SETIHome⁶ και DataSynapse⁷), τα συνεργαζόμενα δίκτυα (collaboration networks, π.χ., Groove⁸) ακόμη και για νέους τρόπους σχεδιασμού της υποδομής του διαδικτύου έτσι ώστε να υποστηρίζει εξελιγμένα πρότυπα επικοινωνίας [96].

Ο πλούτος των επιχειρησιακών ευκαιριών που υπόσχονται τα δίκτυα ομότιμων κόμβων έχει προκαλέσει μεγάλο ενδιαφέρον από τη βιομηχανία και έχει δημιουργήσει διάφορα ερευνητικά και βιομηχανικά προγράμματα⁹, εταιρείες και ομάδες ειδικού ενδιαφέροντος¹⁰. Ερευνητές από τις περιοχές του καταμεμημένου υπολογισμού, των δικτύων, των συστημάτων πρακτόρων και των βάσεων δεδομένων έχουν επίσης ενθουσιαστεί με το όραμα που προσφέρουν τα δίκτυα ομότιμων κόμβων και άρθρα που ασχολούνται με ανοιχτά προβλήματα σε αυτήν την περιοχή έχουν αρχίσει να παρουσιάζονται σε τοποθεσίες υψηλού κύρους (όπως ICDCS, SIGCOMM, INFOCOM, CIDR, SIGMOD, VLDB, κ.ά.) αλλά και σε νέα ειδικευμένα συνέδρια και workshops¹¹.

2.2 Τρία ισχυρά συστήματα δικτύων ομότιμων κόμβων: Napster, Gnutella και Freenet

Τρία ισχυρά συστήματα δικτύων ομότιμων κόμβων είναι τα συστήματα Napster, Gnutella και Freenet. Αυτά τα τρία συστήματα έχουν έναν κοινό στόχο, να διευκολύνουν την ανακάλυψη και τον διαμοιρασμό διάφορων αρχείων (π.χ. εικόνες, τραγούδια και ταινίες) ανάμεσα σε ένα μεγάλο σύνολο από *κόμβους* (χρήστες υπολογιστών). Τα αρχεία που ανταλλάσσονται βρίσκονται αποθηκευμένα σε κόμβους και αφού ανακαλυφθούν από τον ενδιαφερόμενο, αυτός μπορεί έπειτα να τα αποκτήσει χρησιμοποιώντας ένα πρωτόκολλο αντίστοιχο του πρωτοκόλλου HTTP. Εκτός όμως από αυτόν τον κύριο στόχο υπάρχουν βασικές διαφορές μεταξύ των τριών συστημάτων όσον αφορά τα *μεταδεδομένα* (metadata) που διατηρούνται σε κάθε κόμβο του δικτύου, την τοπολογία του κάθε δικτύου ομότιμων κόμβων, την τοποθεσία αποθήκευσης των αρχείων που ανταλλάσσονται, τους αλγόριθμους δρομολόγησης τόσο για επερωτήσεις όσο και για απαντήσεις, τον βαθμό ιδιωτικότητας που παρέχεται στους χρήστες κ.ά.

⁶<http://www.setiathome.ssl.berkeley.edu>

⁷<http://www.datasynapse.com>

⁸<http://www.groove.net>

⁹Βλέπε τα Ευρωπαϊκά προγράμματα DIET (<http://www.dfki.de/diet>), BISON (<http://www.cs.unibo.it/bison/>), MMAPPS (<http://www.mmapps.org>), SWAP (<http://swap.semanticweb.org/>), EVERGROW (<http://www.evergrow.org>), DELIS (<http://delis.upb.de>) και το πρόγραμμα IRIS από τις ΗΠΑ (<http://iris.lcs.mit.edu/>).

¹⁰Για παράδειγμα δείτε τις δραστηριότητες της ομάδας που ασχολείται με συστήματα ομότιμων κόμβων του Global Grid forum στον σύνδεσμο <http://www.gridforum.org>.

¹¹Για παράδειγμα στα παρακάτω: IEEE International Conference on P2P Computing (<http://femto.org/p2p2004/>), International Workshop on Agents and P2P Computing (<http://p2p.ingce.unibo.it>), International Workshop on Peer-to-Peer Systems (<http://iptps04.cs.ucsd.edu/>) κ.ο.κ.

2.3 Ιεραρχικά Δίκτυα Ομότιμων Κόμβων

Μια προσέγγιση που μπορεί να ακολουθηθεί για να χειριστούμε την αδυναμία κλιμάκωσης συστημάτων, όπως το Gnutella, είναι να εισάγουμε μια ιεραρχία στους κόμβους διαχωρίζοντας τους σε *υπερ-κόμβους* (super-peers) και σε *πελάτες* (clients). Αυτήν την προσέγγιση ακολούθησαν οι επόμενες εκδόσεις του συστήματος Gnutella καθώς και άλλα νέα συστήματα όπως το Kazaa και το Morpheus. Σε ένα ιεραρχικό δίκτυο όλοι οι υπερ-κόμβοι είναι ισότιμοι και έχουν τις ίδιες ευθύνες. Κάθε υπερ-κόμβος εξυπηρετεί ένα μέρος των πελατών και κρατάει τα ευρετήρια με τους πόρους αυτών των πελατών. Οι υπερ-κόμβοι επικοινωνούν ακολουθώντας ένα πρωτόκολλο της επιλογής τους (π.χ. ένα συμμετρικό πρωτόκολλο όπως του Gnutella, ένα δομημένο όπως του Napster ή ένα πρωτόκολλο κατανεμημένων πινάκων κατακερματισμού). Οι πελάτες μπορεί να εκτελούνται σε υπολογιστές χρηστών και οι πόροι (π.χ. τα αρχεία σε μια εφαρμογή ανταλλαγής αρχείων) να αποθηκεύονται στους κόμβους αυτών των πελατών. Οι πελάτες είναι ισότιμοι μεταξύ τους δεδομένου ότι το λογισμικό που τρέχει σε κάθε κόμβο πελάτη παρέχει την ίδια λειτουργικότητα. Οι πελάτες μαθαίνουν για πόρους θέτοντας επερωτήσεις σε υπερ-κόμβους, ενώ ανακτούν τους πόρους κατευθείαν από άλλους κόμβους πελάτες.

Τα θέματα που περιλαμβάνονται στον σχεδιασμό και την υλοποίηση ιεραρχικών δικτύων ομότιμων κόμβων έχουν προσφάτως μελετηθεί από διάφορους ερευνητές. Για παράδειγμα, δείτε το άρθρο [113] καθώς και τα συστήματα Edutella [68] και P2P-DIET [50, 24].

2.4 Κατανεμημένοι Πίνακες Κατακερματισμού

Η επιτυχία των πρωτοκόλλων των δικτύων ομότιμων κόμβων και των εφαρμογών τους, όπως το Napster και το Gnutella, έχουν δώσει κίνητρο στους ερευνητές από την περιοχή των κατανεμημένων συστημάτων, των δικτύων και των βάσεων δεδομένων να μελετήσουν πιο προσεκτικά τους βασικούς μηχανισμούς αυτών των συστημάτων και να ανακαλύψουν πως μπορούν αυτοί να υποστηριχτούν με έναν τρόπο που διέπεται από συγκεκριμένες αρχές. Από αυτή την έρευνα προέκυψε ένα νέο κύμα κατανεμημένων πρωτοκόλλων που ονομάστηκαν *κατανεμημένοι πίνακες κατακερματισμού (ΚΠΚ)* και που είχαν ως πρωταρχικό στόχο την περαιτέρω ανάπτυξη των εφαρμογών των δικτύων ομότιμων κόμβων [97, 79, 82, 1, 46, 65, 4]. Οι ΚΠΚ αποτελούν *δομημένα* δίκτυα ομότιμων κόμβων και στοχεύουν στην επίλυση του ακόλουθου *προβλήματος αναζήτησης*:

Έστω X ένα αντικείμενο δεδομένων αποθηκευμένο σε ένα κατανεμημένο δυναμικό δίκτυο από κόμβους. Βρες το αντικείμενο δεδομένων X .

Η βασική ιδέα σε όλους τους ΚΠΚ είναι να λύσουν αυτό το πρόβλημα αναζήτησης προσφέροντας κάποια μορφή κατανεμημένης λειτουργικότητας μέσω ενός πίνα-

κα κατακερματισμού: θεωρώντας ότι αυτά τα αντικείμενα δεδομένων μπορούν να αναγνωριστούν χρησιμοποιώντας *μοναδικά αριθμητικά κλειδιά*, οι κόμβοι του ΚΠΚ συνεργάζονται για να αποθηκεύσουν τα κλειδιά μεταξύ τους (τα αντικείμενα δεδομένων μπορεί να είναι τα πραγματικά δεδομένα ή δείκτες σε δεδομένα). Οι υλοποιήσεις των ΚΠΚ παρέχουν μια πολύ απλή διεπαφή που αποτελείται από δύο βασικές λειτουργίες:

- `put (ID, item)`. Αυτή η λειτουργία εισάγει ένα αντικείμενο με κλειδί `ID` και τιμή `item` στον ΚΠΚ.
- `get (ID)`. Αυτή η λειτουργία επιστρέφει έναν δείκτη στον κόμβο του ΚΠΚ που είναι υπεύθυνος για το κλειδί `ID`.

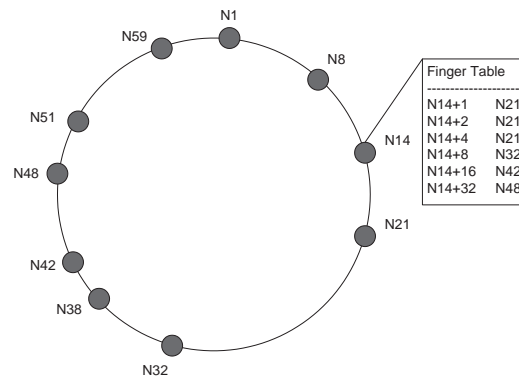
Έστω κι αν οι διαθέσιμοι κατακερματισμένοι πίνακες κατακερματισμού στην βιβλιογραφία διαφέρουν ως προς τις τεχνικές λεπτομέρειες, γενικά ασχολούνται με τις ακόλουθες βασικές ερωτήσεις:

- *Πως αντιστοιχούμε κλειδιά σε κόμβους;* Τα κλειδιά και οι κόμβοι αναγνωρίζονται με έναν δυαδικό αριθμό. Τα κλειδιά αποθηκεύονται σε έναν ή περισσότερους κόμβους με *κοντινά* αναγνωριστικά στο αναγνωριστικό του κλειδιού μέσα στον χώρο των αναγνωριστικών.
- *Πως δρομολογούμε ερωτήσεις για κλειδιά;* Όποιος κόμβος λάβει μια ερώτηση για το κλειδί k , αν διαθέτει το k , επιστρέφει το αντικείμενο δεδομένων X που είναι συσχετισμένο με το k , αλλιώς προωθεί την ερώτηση για το k στον κόμβο με το *κοντινότερο* αναγνωριστικό χρησιμοποιώντας μόνο κάποια τοπική πληροφορία που διαθέτει.
- *Πως διαχειριζόμαστε την δυναμικότητα του δικτύου;* Οι ΚΠΚ μπορούν να προσαρμόζονται σε εισαγωγές, αναχωρήσεις και αποτυχίες κόμβων και να ανανεώνουν τους πίνακες δρομολόγησης με μικρή προσπάθεια.

Οι απαντήσεις στις παραπάνω ερωτήσεις μας δίνουν μια καλή κατηγοριοποίηση υψηλού επιπέδου των υπάρχοντων ΚΠΚ [9, 5]. Στην υπόλοιπη ενότητα εξηγούμε κάποια από τα θέματα που αναφέραμε επικεντρώνοντας σε δύο από τα πρώτα και ισχυρά συστήματα, το Chord [97] και το Pastry [82]. Επιπλέον, παρουσιάζουμε το ΚΠΚ Bamboo δεδομένου ότι είναι αυτό που επιλέξαμε για χρήση στην υλοποίηση του συστήματος Atlas.

2.4.1 Chord

Το Chord χρησιμοποιεί μια παραλλαγή της τεχνικής του *συνεπή κατακερματισμού* (consistent hashing) [51] για να αντιστοιχήσει τα κλειδιά σε κόμβους. Στον συνεπή κατακερματισμό, αντιστοιχείται ένα αναγνωριστικό m -δυφίων σε κάθε κόμβο και αντικείμενο δεδομένων, όπου το m πρέπει να είναι αρκετά μεγάλο



Σχήμα 2.1: Παράδειγμα ενός δακτύλιου αναγνωριστικών με $m=6$ και 10 κόμβους

έτσι ώστε να αποφευχθεί η πιθανότητα διαφορετικά αντικείμενα να έχουν το ίδιο αναγνωριστικό (μια κρυπτογραφική συνάρτηση κατακερματισμού όπως η SHA-1 μπορεί να χρησιμοποιηθεί για αυτό το σκοπό). Το αναγνωριστικό ενός κόμβου μπορεί να υπολογιστεί με τον κατακερματισμό της IP διεύθυνσης του. Για να προκύψει το αναγνωριστικό ενός αντικείμενου δεδομένων πρέπει αρχικά να αποφασίσουμε τι θα αποτελέσει το κλειδί και μετά να το κατακερματίσουμε για να προκύψει το αναγνωριστικό. Για παράδειγμα, σε μια εφαρμογή ανταλλαγής αρχείων το κλειδί μπορεί να είναι το όνομα του αρχείου (αυτό είναι μια απόφαση που εξαρτάται από την αντίστοιχη εφαρμογή). Τα αναγνωριστικά ταξινομούνται σε έναν ιδεατό *δακτύλιο αναγνωριστικών* module 2^m , δηλαδή από 0 έως $2^m - 1$. Το Σχήμα 2.1 δείχνει έναν δακτύλιο αναγνωριστικών με 64 αναγνωριστικά ($m = 6$) αλλά μόνο 10 κόμβους.

Η αντιστοίχιση των κλειδιών στους κόμβους του ιδεατού δακτύλιου αναγνωριστικών γίνεται όπως περιγράφεται παρακάτω. Έστω ότι H είναι η συνεπής συνάρτηση κατακερματισμού που χρησιμοποιείται. Το κλειδί k αντιστοιχίζεται στον πρώτο κόμβο που έχει αναγνωριστικό που είναι ίσο ή ακολουθεί την τιμή $H(k)$ στον χώρο των αναγνωριστικών. Για παράδειγμα, στο δίκτυο του Σχήματος 2.1, ένα κλειδί με αναγνωριστικό 8 θα αποθηκευόταν στον κόμβο $N8$. Με άλλα λόγια, ένα κλειδί k αντιστοιχεί στον κόμβο του οποίου το αναγνωριστικό είναι το πρώτο αναγνωριστικό ακολουθώντας την φορά των δεικτών ρολογιού στον δακτύλιο αναγνωριστικών ξεκινώντας από το $H(k)$. Αυτός ο κόμβος καλείται ο *διάδοχος κόμβος* (successor) του αναγνωριστικού $H(k)$ και συμβολίζεται με $successor(H(k))$. Συχνά θα λέμε ότι αυτός ο κόμβος είναι *υπεύθυνος* για ένα κλειδί k . Στο παράδειγμα μας, ο κόμβος $N32$ είναι υπεύθυνος για όλα τα κλειδιά στο διάστημα $(21, 32]$.

Εάν κάθε κόμβος γνωρίζει τον διάδοχο του, μια ερώτηση για την εύρεση του κόμβου που είναι υπεύθυνος για το κλειδί k μπορεί να απαντηθεί πάντα σε $O(N)$ βήματα όπου N είναι το πλήθος των κόμβων του δικτύου. Για να μπορέσουμε να βελτιώσουμε αυτό το όριο, το σύστημα Chord διατηρεί σε κάθε κόμβο έναν

πίνακα δρομολόγησης που ονομάζεται *πίνακας δεικτών* (finger table) με το πολύ m εγγραφές. Κάθε εγγραφή i στον πίνακα δεικτών του κόμβου n δείχνει στον πρώτο κόμβο s στον δακτύλιο αναγνωριστικών που ακολουθεί το αναγνωριστικό $H(n) + 2^{i-1}$. Αυτοί οι κόμβοι (δηλαδή ο $successor(H(n) + 2^{i-1})$ όταν $1 \leq i \leq m$) ονομάζονται *δείκτες* (fingers) του κόμβου n . Δεδομένου ότι οι δείκτες δείχνουν σε επαναληπτικά διπλάσιες αποστάσεις από τον n μπορούν να επιταχύνουν την αναζήτηση για τον κόμβο που είναι υπεύθυνος για ένα κλειδί k . Εάν οι πίνακες δεικτών έχουν μέγεθος $O(\log N)$ τότε η εύρεση του διαδόχου του κόμβου n μπορεί να γίνει με μεγάλη πιθανότητα σε $O(\log N)$ βήματα [97].

Για την απλοποίηση των αφίξεων και των αναχωρήσεων, κάθε κόμβος n διατηρεί έναν δείκτη στον *προκάτοχο* του (predecessor), δηλαδή στον πρώτο κόμβο με αντίθετη φορά από αυτή του ρολογιού στον δακτύλιο αναγνωριστικών ξεκινώντας από τον n . Όταν ένας κόμβος n θέλει να συνδεθεί σε ένα Chord δίκτυο βρίσκει έναν κόμβο n' που είναι ήδη στο δίκτυο χρησιμοποιώντας κάποιο εξωτερικό μέσο και μετά ζητάει από τον n' να τον βοηθήσει να βρει την θέση του στο δίκτυο βρίσκοντας τον διάδοχο κόμβο του [98]. Κάθε κόμβος εκτελεί περιοδικά έναν *αλγόριθμο σταθεροποίησης* (stabilization algorithm) για να μάθει για κόμβους που έχουν πρόσφατα συνδεθεί στο δίκτυο. Όταν ο n τρέχει το πρωτόκολλο σταθεροποίησης ζητάει από τον διάδοχο του να του πει τον προκατόχο του p . Αν ο p έχει πρόσφατα συνδεθεί στο δίκτυο τότε μπορεί να καταλήξει να είναι αυτός ο διάδοχος του n . Κάθε κόμβος εκτελεί περιοδικά δύο επιπλέον αλγορίθμους για να ελέγξει ότι ο πίνακας δεικτών και ο δείκτης στον προκατόχο του είναι έγκυροι [98]. Οι λειτουργίες σταθεροποίησης μπορεί να επηρεάσουν τις ερωτήσεις προκαλώντας καθυστέρηση στην εκτέλεση τους (δεδομένου ότι οι δείκτες στους διαδόχους είναι σωστοί αλλά οι εγγραφές του πίνακα δεικτών μπορεί να μην είναι σωστές) ή και να είναι λάθος (όταν οι δείκτες στους διαδόχους δεν είναι ακριβείς). Υποθέτοντας όμως ότι οι δείκτες στους διαδόχους είναι σωστοί και η ώρα που χρειάζεται για την διόρθωση των πινάκων δεικτών είναι λιγότερη από την ώρα που θέλει το δίκτυο να διπλασιαστεί σε μέγεθος, αποδεικνύεται ότι οι επερωτήσεις μπορούν να απαντηθούν σωστά με υψηλή πιθανότητα σε $O(\log N)$ βήματα [98].

Για τον χειρισμό των αποτυχιών κόμβων και για την αύξηση της ευρωστίας του συστήματος, κάθε κόμβος n του Chord διατηρεί μια *λίστα διαδόχων* μεγέθους r που περιέχει τους r πρώτους διαδόχους του n . Αυτή η λίστα χρησιμοποιείται όταν ο διάδοχος του n έχει αποτύχει. Στην πράξη, ακόμα και μικρές τιμές του r είναι αρκετές για να επιτευχθεί η ευρωστία του δικτύου [98]. Εάν ένας κόμβος επιλέξει να αποχωρήσει οικειοθελώς από το δίκτυο του Chord τότε μπορεί να ενημερώσει τον διάδοχο και τον προκατόχο του έτσι ώστε να μπορούν να αλλάξουν τους δείκτες τους και επιπλέον να μπορεί μεταφέρει τα κλειδιά στον διάδοχο του. Μπορεί να φανεί ότι με μεγάλη πιθανότητα όποιος κόμβος συνδέεται ή αποσυνδέεται από ένα δίκτυο Chord μπορεί να χρησιμοποιήσει $O(\log^2 N)$ μηνύματα για να δημιουργήσει σωστά όλους τους δείκτες σε διαδόχους, σε προκατόχους και τους πίνακες δεικτών [97].

2.4.2 Pastry

Οι κόμβοι και τα αντικείμενα δεδομένων στο Pastry [82] επίσης αντιστοιχούνται σε έναν κυκλικό χώρο αναγνωριστικών modulo 2^m χρησιμοποιώντας μια συνάρτηση κατακερματισμού όπως η SHA-1. Ένας κόμβος διατηρεί την τοπική κατάσταση με τρεις δομές δεδομένων: το σύνολο φύλλων (leaf set), τον πίνακα δρομολόγησης (routing table) και το σύνολο γειτόνων (neighbourhood set). Το σύνολο φύλλων ενός κόμβου n είναι το σύνολο των $2r$ κόμβων που ακολουθούν ή προηγούνται κατευθείαν του n στον κυκλικό χώρο αναγνωριστικών. Θα χρησιμοποιήσουμε το $L = \{L_i, -r \leq i \leq r\}$ για να αναπαραστήσουμε αυτό το σύνολο όπου το L_0 θα είναι ο κόμβος n . Το σύνολο φύλλων του Pastry αντιστοιχεί στο σύνολο διαδόχων του Chord. Όπως στο Chord, το σωστό σύνολο φύλλων αρκεί για να εξασφαλίσει την σωστή δρομολόγηση στο Pastry (εντός $O(N)$ βημάτων όπου το N είναι ο αριθμός των κόμβων στο δίκτυο).

Ας υποθέσουμε ότι τα αναγνωριστικά των κόμβων μπορούν να θεωρηθούν ακολουθίες ψηφίων βάσης 2^b . Ο πίνακας δρομολόγησης ενός κόμβου n στο Pastry οργανώνεται σαν ένας διδιάστατος πίνακας R με $\log_{2^b} N$ γραμμές και $2^b - 1$ στήλες. Έστω ότι με $R[l, i]$ αναφερόμαστε στο στοιχείο του πίνακα δρομολόγησης στη σειρά l και στη στήλη i . Ο κόμβος n επιλέγει την εγγραφή $R[l, i]$ του πίνακα δρομολόγησης να είναι ένας κόμβος με αναγνωριστικό που ταιριάζει με το αναγνωριστικό του n στα πρώτα l ψηφία και το $(l + 1)$ -στό ψηφίο είναι το i . Η επιλογή μεταξύ πολλών ενδεχομένως κόμβων με αυτήν την ιδιότητα μπορεί να γίνει χρησιμοποιώντας κάποιο μέτρο εγγύτητας.

Το σύνολο γειτόνων ενός κόμβου n είναι ένα σύνολο κόμβων που είναι κοντά στο n βάσει κάποιου κλιμακωτού μέτρου εγγύτητας, π.χ. τον αριθμό των βημάτων δρομολόγησης IP ή την γεωγραφική απόσταση. Η χρήση του συνόλου γειτόνων επιτρέπει στο Pastry να επιτυγχάνει καλές ιδιότητες τοπικότητας [82].

Η δρομολόγηση στο Pastry γίνεται ως εξής. Εάν ένας κόμβος n θέλει να εντοπίσει τον κόμβο με το κλειδί k τότε πρώτα ελέγχει αν το k περιλαμβάνεται στο δικό του σύνολο φύλλων. Εάν περιλαμβάνεται, τότε ο n στέλνει ένα μήνυμα αναζήτησης (lookup) στο μέλος του συνόλου φύλλων με αναγνωριστικό αριθμητικά κοντινότερο στο k (modulo 2^m). Εάν αυτός ο κόμβος είναι ο n , τότε η διαδικασία αναζήτησης τερματίζεται. Διαφορετικά, ο n υπολογίζει το μήκος l του μεγαλύτερου κοινού προθέματος του k και του δικού του αναγνωριστικού και χρησιμοποιεί τον πίνακα δρομολόγησης για να στείλει ένα μήνυμα αναζήτησης στον κόμβο $R[l, k[l]]$. Εάν αυτό δεν είναι δυνατό επειδή η εγγραφή $R[l, k[l]]$ του πίνακα δρομολόγησης είναι κενή, ο κόμβος n στέλνει ένα μήνυμα αναζήτησης στο μέλος του συνόλου φύλλων με αναγνωριστικό αριθμητικά κοντινότερο στο k . Ο κόμβος που λαμβάνει το μήνυμα αναζήτησης συνεχίζει με τον ίδιο τρόπο έτσι ώστε τελικά να φτάσουμε στον κόμβο που είναι υπεύθυνος για το k . Μπορεί να αποδειχτεί ότι αυτή η διαδικασία δρομολόγησης ολοκληρώνεται σε $O(\log_{2^b} N)$ βήματα υποθέτοντας ακριβείς πίνακες δρομολόγησης και απουσία πρόσφατων αποτυχιών κόμβων [82]. Τα σύνολα φύλλων χρησιμοποιούνται όταν οι πίνακες δρομολόγησης δεν

είναι πλήρεις με πιθανό κόστος μεγαλύτερα μονοπάτια αναζήτησης. Τα σύνολα φύλλων όμως προσθέτουν μεγάλη *προσαρμοστικότητα* (resilience) στο δίκτυο Pastry. Όπως φαίνεται στο άρθρο [38], εάν $r = 16$ και $N = 65536$, ακόμη και μετά από μια τυχαία αποτυχία 30% των συνδέσμων του δικτύου, υπάρχουν ακόμη συνδεδεμένα μονοπάτια μεταξύ κάθε ζεύγους κόμβων.

Όταν ένας νέος κόμβος n με κλειδί x θέλει να συνδεθεί σε ένα δίκτυο Pastry, πρέπει πρώτα να βρει έναν κόμβο n' του δικτύου που είναι κοντά του βάσει κάποιου μέτρου εγγύτητας. Τότε ο n ζητάει από τον n' να δρομολογήσει ένα μήνυμα *σύνδεσης* (*join*) στον κόμβο με κλειδί x . Εάν ο κόμβος n'' είναι ο κόμβος προορισμού του μηνύματος σύνδεσης τότε οι κόμβοι n', n'' και όλοι οι κόμβοι στο μονοπάτι από τον n' στο n'' στέλνουν τους πίνακες κατάστασης τους στον n . Ο κόμβος n χρησιμοποιεί αυτούς τους πίνακες για να δημιουργήσει τον πίνακα δρομολόγησης, το σύνολο φύλλων και το σύνολο γειτόνων καθώς και για να ενημερώσει τους άλλους κόμβους για την παρουσία του. Αυτή η διαδικασία ολοκληρώνεται σε $O(\log_{2b}^2 N)$ βήματα [82]. Το Pastry μπορεί να διαχειριστεί αποτυχίες ή αποχωρήσεις κόμβων (ακόμη και ταυτόχρονες) χρησιμοποιώντας κατάλληλα πρωτόκολλα [82]. Για παράδειγμα, εάν ένας κόμβος n ανακαλύψει ότι ο κόμβος L_i του συνόλου φύλλων του έχει αποτύχει, θα επικοινωνήσει με τον κόμβο με το μεγαλύτερο ευρετήριο σε σχέση με τον κόμβο που απέτυχε (π.χ., L_r εάν $i > 0$ και L_{-r} εάν $i < 0$) και θα ζητήσει ένα αντίγραφο του συνόλου φύλλων του. Μετά, αυτό το σύνολο θα χρησιμοποιηθεί από τον n για να βρεθεί ο κατάλληλος κόμβος που θα αντικαταστήσει τον L_i . Έπειτα, ο n στέλνει ένα μήνυμα σε όλους τους κόμβους από το σύνολο φύλλων του ενημερώνοντας τους για αυτήν την αλλαγή. Εκτός κι αν r κόμβοι με κοντινά αναγνωριστικά αποτύχουν ταυτόχρονα, αυτή η διαδικασία εγγυάται ότι οι κόμβοι μπορούν να ανακάμψουν όταν περιέχονται αποτυχημένοι κόμβοι στο σύνολο φύλλων τους. Λόγω της διαφορετικότητας των κόμβων με κοντινά αναγνωριστικά στο Pastry τέτοια αποτυχία δεν είναι πιθανό να συμβεί ακόμα και για μικρές τιμές του r [82].

2.4.3 Bamboo

Το Bamboo είναι ένα δίκτυο ΚΠΚ βασισμένο στο Pastry, από το οποίο κληρονομεί τον κυκλικό χώρο αναγνωριστικών και τον αλγόριθμο δρομολόγησης [80]. Το Bamboo υπερισχύει του συστήματος Pastry γιατί χρησιμοποιεί περισσότερο αυξητικούς αλγορίθμους για τις εισαγωγές κόμβων και την διαχείριση των γειτόνων. Αυτό επιτρέπει στο Bamboo να αντέχει σε πολύ δυναμικές αλλαγές του δικτύου δηλαδή προσαρμόζεται και σε υψηλούς ρυθμούς αποχωρήσεων και αφίξεων (churn). Ο όρος churn είναι ένας όρος που χρησιμοποιείται για να χαρακτηρίσει την συνεχή διαδικασία αφίξεων και αναχωρήσεων κόμβων που έχει παρατηρηθεί σε διάφορες εφαρμογές δικτύων ομοτίμων κόμβων όπως τα Napster, Gnutella και Kazaa [84, 91, 39]. Στο άρθρο [80] δοκιμάσανε μια υλοποίηση του Chord (MIT Chord) και του Pastry (FreePastry v1.3) από το Πανεπιστήμιο Rice. Δείξανε ότι κάτω από υψηλό ρυθμό αφίξεων και αναχωρήσεων αυτές οι υλοποιή-

σεις αποτύγχαναν να ολοκληρώσουν τις αιτήσεις αναζήτησης και επέστρεφαν μη συνεπή αποτελέσματα για την ίδια αναζήτηση από διαφορετικούς κόμβους ή επέστρεφαν συνεπή αποτελέσματα αλλά με μεγάλη αργοπορία λόγω της καθυστέρησης του δικτύου. Στο [80] έχουν δείξει χρησιμοποιώντας το δίκτυο Bamboo ότι τα ΚΠΚ μπορούν να διαχειριστούν υψηλότερους ρυθμούς αφίξεων και αναχωρήσεων και αναγνώρισαν τρεις παράγοντες που επηρεάζουν τους ΚΠΚ κάτω από τέτοιες συνθήκες.

Ο πρώτος παράγοντας είναι η *αντιδραστική* (reactive) σε σχέση με την *περιοδική* (periodic) ανάκαμψη από τις αποτυχίες. Στην αντιδραστική ανάκαμψη, εάν ένας κόμβος ανακαλύψει ότι ένας από τους κόμβους φύλλα έχει αποτύχει, δημιουργεί ένα νέο σύνολο φύλλων και μετά ενημερώνει όλα τα μέλη του συνόλου όπως αναφέρθηκε στην Ενότητα 2.4.2. Σε καταστάσεις χαμηλού ρυθμού αφίξεων και αναχωρήσεων αυτή η προσέγγιση μπορεί να κλιμακωθεί με επιτυχία και οδηγεί σε μηνύματα πολύ χαμηλού κόστους. Όμως, καθώς αυξάνεται ο ρυθμός (όπως σε σενάρια ανταλλαγής αρχείων που παρουσιάζονται στα άρθρα [84, 91, 39]), η περιοδική ανάκαμψη αποτελεί μια καλύτερη στρατηγική: ένας κόμβος *περιοδικά* μοιράζεται το σύνολο φύλλων με κάθε κόμβο στο σύνολο και κάθε κόμβος αντίστοιχα απαντάει με τον ίδιο τρόπο.

Ο δεύτερος παράγοντας είναι ο τρόπος επιλογής των *χρονικών ορίων των μηνυμάτων* (message timeouts). Οι κόμβοι πρέπει να εξασφαλίζουν ότι το χρονικό όριο μιας αίτησης επιλέγεται με βάσει συγκεκριμένα κριτήρια. Το άρθρο [80] δείχνει ότι η χρήση σταθερών χρονικών ορίων δεν αποτελεί καλή ιδέα και χρειάζεται κάποιος πιο *έξυπνος* υπολογισμός των χρονικών ορίων κάτω από συνθήκες υψηλού ρυθμού αφίξεων και αναχωρήσεων.

Τέλος, ο τελευταίος παράγοντας είναι η επιλογή του μέτρου εγγύτητας μεταξύ των γειτόνων. Στο άρθρο [80] προτείνεται η περιοδική εκτέλεση ενός αλγορίθμου συντήρησης για την ανανέωση των πινάκων δρομολόγησης ενός κόμβου n έτσι ώστε να συμπεριληφθούν οι γείτονες που είναι κοντινότερα στον κόμβο n βάσει του μέτρου εγγύτητας. Δείχνουν πειραματικά ότι η *καθοδική δειγματοληψία* (global sampling), χρησιμοποιώντας την λειτουργία αναζήτησης του ΚΠΚ, είναι μια τεχνική που δίνει καλά αποτελέσματα κάτω από υψηλούς ρυθμούς αφίξεων και αναχωρήσεων.

Η παραπάνω μελέτη συστημάτων δικτύων ομοτίμων κόμβων και καταμεμημένων πινάκων κατακερματισμού δύσκολα μπορεί να αποτελέσει μια πλήρη ανασκόπηση της περιοχής. Πιο λεπτομερείς ανασκοπήσεις περιλαμβάνονται στις δημοσιεύσεις [67, 29, 9, 106].

2.5 Το πλαίσιο RDF

Το πλαίσιο RDF (Resource Description Framework) επιτρέπει την αναπαράσταση πληροφορίας για πόρους του παγκόσμιου ιστού.

Αποτελείται από συστάσεις (recommendations) W3C¹² που καθιστούν δυνατή την κωδικοποίηση, την ανταλλαγή και την επαναχρησιμοποίηση δομημένων *μεταδεδομένων* παρέχοντας ένα τρόπο για δημοσίευση λεξιλογίων που μπορούν να διαβαστούν από ανθρώπους (human-readable) αλλά και να επεξεργαστούν από μηχανές (machine-processable). Οι τρέχουσες W3C συστάσεις για το πλαίσιο RDF μπορούν να χρησιμοποιηθούν σε ένα πλήθος από περιοχές, για παράδειγμα:

- στην *ανακάλυψη πόρων*, παρέχοντας ακριβή αποτελέσματα.
- στην *καταλογογράφηση*, για αποδοτική περιγραφή των πόρων του παγκόσμιου ιστού.
- από *έξυπνους πράκτορες λογισμικού*, για την βελτίωση της ανταλλαγής και του διαμοιρασμού γνώσης.
- στην *αξιολόγηση περιεχομένου*.
- για την περιγραφή *πνευματικών δικαιωμάτων* σελίδων του παγκόσμιου ιστού.
- για την περιγραφή των *επιλογών ιδιωτικότητας* ενός χρήστη καθώς και των *πολιτικών ιδιωτικότητας* ενός ιστοτόπου.

Έστω και αν το RDF είχε αρχικά προταθεί στα πλαίσια του Σηματολογικού Ιστού (Semantic Web), είναι πολύ λογικό να χρησιμοποιηθεί για την αναπαράσταση πληροφορίας που περιγράφει πόρους και σε άλλα πλαίσια π.χ. σε περιβάλλοντα Πλέγματος (Grid), σε περιβάλλοντα δικτύων ομότιμων κόμβων κ.ά.

Το μοντέλο δεδομένων του RDF χρησιμοποιεί τις ακόλουθες βασικές έννοιες:

- *Πόροι (resources)*: Στο RDF ένας πόρος είναι οτιδήποτε θέλουμε να περιγράψουμε. Ένας πόρος μπορεί να είναι μία ιστοσελίδα, ένα μέρος αυτής ή μια συλλογή από ιστοσελίδες. Επιπλέον, έναν πόρο μπορεί να αποτελεί ένα βιβλίο, ένας συγγραφέας, μια δημοσίευση ή ένα αρχείο υπολογιστή. Κάθε πόρος προσδιορίζεται μοναδικά από ένα *Universal Resource Identifier*(URI) [11]. Αναφέρουμε ότι ένα αναγνωριστικό δεν παρέχει υποχρεωτικά πρόσβαση στον πόρο που προσδιορίζει.
- *Ιδιότητες (properties)*: Μια ιδιότητα αποτελεί ένα χαρακτηριστικό ενός πόρου. Για παράδειγμα, η ιδιότητα *παροχέας* μπορεί να είναι η εταιρεία που παρέχει μια συγκεκριμένη υπηρεσία του παγκόσμιου ιστού. Οι ιδιότητες επίσης προσδιορίζονται από URIs.
- *Δηλώσεις (statements)*: Οι δηλώσεις είναι οι δομές που παρέχονται από το RDF για την αναπαράσταση πληροφοριών για ένα πεδίο. Μία δήλωση έχει τρία μέρη: τον πόρο για την οποίο γίνεται η δήλωση, την ιδιότητα του πόρου στην οποία αναφέρεται η δήλωση και την τιμή αυτής της

¹²<http://www.w3.org/>

ιδιότητας. Τα τρία μέρη μιας δήλωσης ονομάζονται αντίστοιχα, *υποκείμενο* (*subject*), *κατηγορημα* (*predicate*) και *αντικείμενο* (*object*). Το αντικείμενο μιας δήλωσης μπορεί να είναι ένας άλλος πόρος ή ένα *κυριολεκτικό* (*literal*), δηλαδή μια ατομική τιμή (π.χ. μια συμβολοσειρά). Για παράδειγμα, στην δήλωση: Η Amazon είναι η εταιρεία που είναι ιδιοκτήτρια του <http://www.amazon.com>, "Amazon" είναι το υποκείμενο, "είναι ιδιοκτήτρια" είναι το κατηγορημα και το "<http://www.amazon.com>" είναι το αντικείμενο.

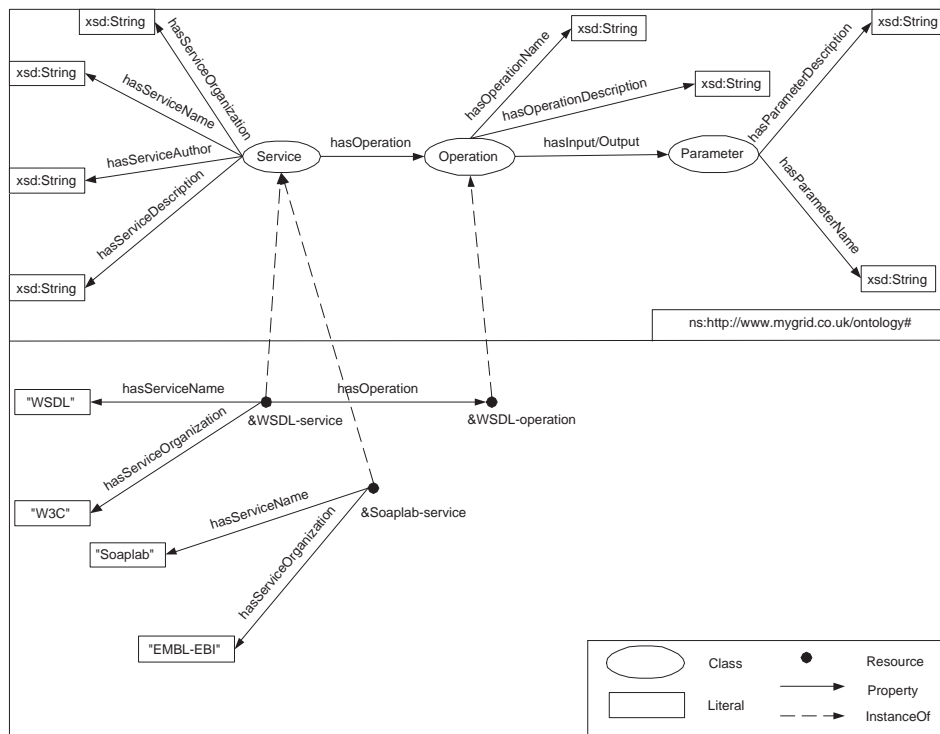
Ο ορισμός του RDF δεν περιγράφει μια συγκεκριμένη σύνταξη. Δύο πιθανές αναπαραστάσεις δεδομένων RDF είναι οι *γράφοι με ετικέτες* (*labeled graphs*) και οι λίστες τριπλετών (*triple lists*). Στην πρώτη αναπαράσταση μια οντότητα αναπαριστάται με έναν *κόμβο* και μια ιδιότητα με μια *ακμή*. Στην αναπαράσταση της λίστας τριπλετών όλες οι δηλώσεις αναπαριστώνται σαν τριπλέτες της μορφής *υποκείμενο-κατηγορημα-αντικείμενο* (πόρος-ιδιότητα-τιμή ιδιότητας). Μια εναλλακτική είναι η χρήση μιας κωδικοποίησης βασισμένης στην γλώσσα XML για την αναπαράσταση δεδομένων RDF.

Το μοντέλο δεδομένων του RDF παρέχει έναν απλό τρόπο για την περιγραφή των σχέσεων μεταξύ των πόρων με χρήση ονομασμένων ιδιοτήτων και τιμών αλλά δεν παρέχει μηχανισμούς για τον ορισμό των σχέσεων μεταξύ αυτών των ιδιοτήτων και άλλων πόρων. Αυτός είναι ο ρόλος του RDF σχήματος (RDF Schema (RDFS)) [15]. Το RDFS αποτελεί ένα είδους λεξικό, ορίζει τους όρους που θα χρησιμοποιηθούν στις RDF δηλώσεις και δίνει συγκεκριμένα νοήματα σε αυτούς. Το RDFS ορίζει όχι μόνο τις ιδιότητες ενός πόρου (π.χ. τίτλος, συγγραφέας, θέμα κ.ά.) αλλά μπορεί να ορίζει και τους τύπους των πόρων που περιγράφονται (άρθρο, ιστοσελίδα, βιβλίο κ.ά.). Επομένως, το RDFS παρέχει ένα βασικό σχήμα ή ένα μέσο ορισμού *τύπων* (*type definition facility*) όπως συνήθως αναφέρεται στις περιοχές των βάσεων δεδομένων ή στις γλώσσες προγραμματισμού [3].

Το πλαίσιο RDF και το RDFS συνεισφέρουν πολύ στην πραγματοποίηση του οράματος του σηματολογικού ιστού, δεδομένου ότι παρέχουν ένα πρότυπο μοντέλο για την περιγραφή γεγονότων για πόρους του παγκόσμιου ιστού καθώς και την δυνατότητα διερμηνείας αυτών των γεγονότων.

Στην συνέχεια παρουσιάζουμε ένα παράδειγμα σχήματος RDFS που περιγράφει πληροφορία για υπηρεσίες του παγκόσμιου ιστού. Αυτό το παράδειγμα είναι μέρος του βασικού μοντέλου που χρησιμοποιήθηκε στο πρόγραμμα *myGrid*¹³. Υπάρχουν τρεις κλάσεις (οντότητες) σε αυτό το σχήμα: *Service*, *Operation* και *Parameter*. Η βασική διαφορά μεταξύ των κλάσεων *Service* και *Operation* είναι ότι η πρώτη περιγράφει πώς δημοσιεύονται οι πόροι ενώ η δεύτερη αναφέρεται στην λειτουργικότητά τους. Γενικά, μια υπηρεσία μπορεί να παρέχει ένα σύνολο από σχετικές αλλά ανεξάρτητες λειτουργίες. Στο μοντέλο δεδομένων του *myGrid*, οι πληροφορίες που μπορούν να δημοσιευτούν για μια υπηρεσία περιλαμβάνουν ιδιότητες όπως το όνομα του παροχέα του οργανισμού, τον συγγραφέα

¹³<http://www.mygrid.org.uk>



Σχήμα 2.2: RDFS σχήμα για υπηρεσίες του παγκόσμιου ιστού

κ.ά. Οι δυνατότητες μιας υπηρεσίας (δηλαδή οι λειτουργίες της) μπορούν να χαρακτηριστούν από τις εισόδους, τις εξόδους και κάποια γνωρίσματα σχετικά με το συγκεκριμένο πεδίο στο οποίο αναφέρεται. Οι εισοδοι και οι έξοδοι μοντελοποιούνται μέσω της οντότητας *Parameter* [62]. Το σχήμα που περιγράφηκε παρουσιάζεται στο Σχήμα 2.2. Έχουμε επιλέξει το σχήμα *οβάλ* για την κλάση, ένα *απλό βέλος* για την ιδιότητα και ένα *διακεκομμένο βέλος* για την σχέση *στιγμιότυποΚλάσης* μεταξύ των κλάσεων και των στιγμιότυπων τους. Εάν μια τιμή είναι ένα κυριολεκτικό, το αναπαριστούμε με ένα *ορθογώνιο*.

2.6 Γλώσσες επερωτήσεων RDF

Έχουν προταθεί πλήθος από γλώσσες επερωτήσεων για το πλαίσιο RDF καθώς και συστήματα που υλοποιούν αυτές τις γλώσσες. Παρουσιάζουμε μια σύντομη περιγραφή και σύγκριση αυτών ακολουθώντας την προσέγγιση που παρουσιάστηκε στο άρθρο [41].

2.6.1 RQL

Η γλώσσα RQL (RDF Query Language) [52] είναι μια συναρτησιακή γλώσσα βασισμένη στη γλώσσα OQL [22] η οποία βασίζεται σε ένα τυπικό μοντέλο γράφων

το οποίο υποστηρίζει τις βασικές αρχές του μοντέλου RDF. Η RQL υιοθετεί τη λειτουργικότητα γλωσσών επερωτήσεων ημιδομημένων δεδομένων και XML δεδομένων (π.χ. εκφράσεις μονοπατιών) αλλά και επιπλέον επεκτείνει αυτές τις λειτουργίες έτσι ώστε να κάνει *ομοιόμορφα* επερωτήσεις τόσο σε περιγραφές RDF όσο και σε σχήματα RDF χρησιμοποιώντας ένα σύνολο βασικών επερωτήσεων και τελεστών. Ένα από τα βασικά χαρακτηριστικά της είναι ότι υποστηρίζει *γενικευμένες εκφράσεις μονοπατιών* (generalized path expressions) με μεταβλητές στις ετικέτες των κόμβων αλλά και των ακμών. Η καινοτομία της RQL έγκειται στην ικανότητα της να συνδυάζει επερώτηση σε σχήμα και δεδομένα με εύκολο τρόπο ενώ παράλληλα εκμεταλλεύεται την ιεραρχία των ετικετών και την πολλαπλή ταξινόμηση των πόρων. Η RQL επιπλέον ορίζει ένα τυπικό μοντέλο και ένα σύστημα τύπων για το RDF που έχει μερικές διαφορές από την θεωρία του RDF που έχει υιοθετηθεί από το W3C [44].

Η σύνταξη της RQL περιλαμβάνει ένα σύνολο βασικών επερωτήσεων (π.χ. Resource, SubClassOf() κ.ά.) και επερωτήσεων παρόμοιες με την επερώτηση SQL select - from - where. Πληροφορίες σχετικές με τον χώρο ονομάτων μπορούν να δοθούν χρησιμοποιώντας την πρόταση "USING NAMESPACE". Το αποτέλεσμα μιας επερώτησης είναι ένα σύνολο από δεσμευμένες μεταβλητές.

Η RQL υλοποιείται στο σύστημα RDF Suite του ICS-FORTH ¹⁴.

Ας θεωρήσουμε το RDFS σχήμα του Σχήματος 2.2 και ας υποθέσουμε ότι θέλουμε να βρούμε μια υπηρεσία χρησιμοποιώντας τα μεταδεδομένα της υπηρεσίας αυτής. Έστω ότι θέλουμε να κάνουμε την μια επερώτηση όπου η ζητούμενη υπηρεσία δημοσιεύεται από τον οργανισμό "W3C" και η περιγραφή της περιλαμβάνει την λέξη "bioinformatics". Η επερώτηση αυτή σε RQL εκφράζεται ως εξής:

```
SELECT X FROM {X;ns:Service}ns:hasServiceOrganization{Y},
{X}ns:hasServiceDescription{W} WHERE Y = "W3C" and X like
"*bioinformatics*" USING NAMESPACE ns =
http://www.mygrid.co.uk/ontology#
```

Μια άλλη πιθανή επερώτηση είναι να αναζητηθεί μια υπηρεσία βάσει των χαρακτηριστικών μιας παραμέτρου εισόδου π.χ., βρες μια υπηρεσία της οποίας η περιγραφή μιας παραμέτρου εισόδου περιέχει την φράση "DNA sequence". Αυτό εκφράζεται ως εξής στην RQL:

```
SELECT X FROM
{X;ns:Service}ns:hasOperation.ns:hasInput/Output{Y},
{Y}ns:hasParameterDescription{Z}
WHERE Z like "*DNA sequence*" USING NAMESPACE ns =
http://www.mygrid.co.uk/ontology#
```

¹⁴<http://139.91.183.30:9090/RDF/>

2.6.2 RDQL

Η γλώσσα RDF Data Query Language (RDQL) [89] αποτελεί ένα W3C submission. Αποτελείται από ένα πρότυπο γράφου, ο οποίος εκφράζεται σαν μια λίστα από τριπλέτες που αποτελείται από μεταβλητές και τιμές RDF (URIs και literals). Η σύνταξη της RDQL διαθέτει τον όρο SELECT παρόμοιο με την SQL αλλά όχι τον όρο FROM. Οι συντομεύσεις των χώρων ονομάτων μπορούν να χρησιμοποιηθούν με έναν ξεχωριστό όρο FROM. Το αποτέλεσμα μιας επερωτήσεως είναι ένας πίνακας με μεταβλητές μαζί τις πιθανές δεσμεύσεις τους.

Η RDQL αρχικά υλοποιήθηκε στο σύστημα Jena 1.0.2¹⁵.

Οι επερωτήσεις που παρουσιάστηκαν προηγουμένως μπορούν να εκφραστούν στην RDQL ως εξής:

```
SELECT ?x WHERE (?x, ns:hasServiceOrganization, ?y)
      (?x, ns:hasServiceDescription, ?w)
AND   ?y = "W3C" AND   ?w =~ "bioinformatics" USING ns FOR
<http://www.mygrid.co.uk/ontology#>
```

```
SELECT ?x WHERE (?x, ns:hasOperation, ?q)
      (?q, ns:hasInput/Output, ?y)
      (?y, ns:hasParameterDescription, ?z)
AND   ?z =~ "DNA sequence" USING ns FOR
<http://www.mygrid.co.uk/ontology#>
```

Κλείνουμε την παρουσίαση αυτή τονίζοντας ότι ακόμη και αν ο ορισμός της γλώσσας RDQL δεν περιλαμβάνει πληροφορία RDFS (π.χ. σύνδεσμοι στιγμιοτύπων μέσω ιεραρχιών υποκλάσεων δεν περιλαμβάνονται στις απαντήσεις των επερωτήσεων), οι υλοποιήσεις της RDQL όπως οι Jena και Sesame υποστηρίζουν την αποτίμηση επερωτήσεων RDQL με χρήση της σημασιολογίας της RDFS.

2.6.3 SPARQL

Η γλώσσα SPARQL έχει προσφάτως αποκτήσει την κατάσταση ενός W3C working draft. Είναι μία γλώσσα επερωτήσεων για γράφους RDF και έχει την δυνατότητα να εξαγει πληροφορίες τόσο για πόρους και ιδιότητες όσο και για RDF υπογράφους. Χρησιμοποιώντας την SPARQL είναι πιθανό να δημιουργηθούν νέοι RDF γράφοι χρησιμοποιώντας επερωτήσεις σε υπάρχοντες RDF γράφους. Η γλώσσα επερωτήσεων SPARQL βασίζεται στην ιδέα του ταιριάσματος προτύπων γράφων. Τα απλούστερα πρότυπα γράφων είναι τα πρότυπα τριπλετών που είναι σαν μια RDF τριπλέτα αλλά με την πιθανότητα ύπαρξης μεταβλητής σε κάποια από τις θέσεις του υποκειμένου, κατηγορήματος ή αντικειμένου. Ο συνδυασμός αυτών δίνει ένα βασικό πρότυπο γράφου όπου χρειάζεται ένα ακριβές ταιρίασμα σε ένα γράφο για να ικανοποιηθεί ένα πρότυπο. Είναι επίσης πιθανό να περιοριστούν οι

¹⁵<http://www.hpl.hp.com/semweb/jena.htm>

τιμές που επιτρέπονται στο ταίριασμα ενός προτύπου. Οι επερωτήσεις μπορούν να έχουν επίδραση σε περισσότερους τους ενός γράφου. Η σύνταξη της SPARQL ακολουθεί τις προτάσεις της SQL `select-from-where` με τις προσθήκες των όρων `CONSTRUCT`, `DESCRIBE` και `ASK`. Το σύνολο των αποτελεσμάτων ποικίλει ανάλογα με τον κύριο όρο της επερώτησης. Στην περίπτωση μιας επερώτησης με τον όρο `SELECT` το σύνολο αποτελεσμάτων είναι ένα σύνολο μεταβλητών και οι πιθανές δεσμεύσεις τους. Εάν όμως περιέχει τον όρο `CONSTRUCT` το αποτέλεσμα είναι ένας RDF γράφος που δημιουργείται με την αντικατάσταση των μεταβλητών με ένα σύνολο προτύπων τριπλετών. Τέλος, η χρήση του όρου `DESCRIBE` επιστρέφει έναν RDF γράφο που περιγράφει τους πόρους που ανακαλύπτονται και η επερώτηση με τον όρο `ASK` επιστρέφει ναι ή όχι ανάλογα με το αν το πρότυπο επερώτησης ταιριάζει ή όχι.

Ας παρουσιάσουμε πάλι τις παραδειγματικές επερωτήσεις που παρουσιάσαμε προηγουμένως. Η μετάφραση τους σε SPARQL επερωτήσεις είναι η ακόλουθη:

```
PREFIX ns: <http://www.mygrid.co.uk/ontology#> SELECT ?x WHERE {  
{?x ns:hasServiceOrganization ?y;  
  ns:hasServiceDescription ?z}.  
  FILTER ?y = "W3C" && regex(str(?z), "bioinformatics")}
```

```
PREFIX ns:  
<http://www.mygrid.co.uk/ontology#> SELECT ?x WHERE { ?x  
ns:hasOperation ?w.  
  ?w ns:hasInput/Output ?y.  
  ?y ns:hasParameterOutput ?z.  
  FILTER regex(str(?z), "DNA sequence") }
```

Δεν υπάρχει κάποια σταθερή υλοποίηση της γλώσσας SPARQL.

2.6.4 N3

Η γλώσσα Notation3 (N3) [12] αποτελεί μια σύνταξη κειμένου για δεδομένα RDF. Επιτρέπει τον ορισμό κανόνων που μπορούν να χρησιμοποιηθούν για επερωτήσεις. Η σημασιολογία του RDF πρέπει να παρέχεται από συγκεκριμένους κανόνες. Έστω και αν η γλώσσα N3 είναι πολύ πιο συνεκτική και ευκολότερα αναγνώσιμη σε σχέση με την XML, η χρήση της σαν μια γλώσσα επερωτήσεων για RDF προκαλεί αρκετά προβλήματα.

Η N3 υποστηρίζεται από δύο συστήματα που είναι διαθέσιμα ελεύθερα το CWM¹⁶ και το Euler¹⁷.

¹⁶<http://www.w3.org/2000/10/swap/doc/cwm.html>

¹⁷<http://www.agfa.com/w3c/euler/>

2.6.5 SeRQL

Η γλώσσα Sesame RDF Query language (SeRQL) [16] είναι μια γλώσσα επερωτήσεων βασισμένη στις υπάρχουσες γλώσσες RQL, RDQL και N3. Ο σκοπός της γλώσσας είναι να χρησιμοποιήσει ιδέες από αυτές και ταυτόχρονα να ικανοποιήσει συγκεκριμένους σχεδιαστικούς στόχους. Η SeRQL χρησιμοποιεί μια τυπική μετάφραση του RDF βασισμένη στη θεωρία του μοντέλου RDF [44].

Η σύνταξη της SeRQL βασίζεται σε αυτή της RQL έστω κι αν έχουν γίνει κάποιες αλλαγές και απλοποιήσεις. Τα βασικά χαρακτηριστικά της γλώσσας είναι η υποστήριξη γενικευμένων εκφράσεων μονοπατιού και το προαιρετικό ταίριασμα (optional matching). Η SeRQL επιπλέον υποστηρίζει τις δύο προτάσεις `select-from-where` και `construct-from-where`. Αυτές οι δύο προτάσεις μπορούν να περιέχουν και μια επιπλέον πρόταση `using namespace` για τον ορισμό διαφορετικών χώρων ονομάτων. Επιπλέον, μπορεί να χρησιμοποιηθεί η πρόταση `construct` για τον ορισμό απλών κανόνων ή μετατροπές γράφων.

Το σύστημα Sesame προσφέρει υλοποίηση της γλώσσας SeRQL¹⁸.

2.6.6 TRIPLE

Η TRIPLE [93] είναι μια γλώσσα επερωτήσεων, εξαγωγής συμπερασμάτων και μετατροπής για τον Σημαιολογικό Ιστό. Βασίζεται στην λογική Horn και δανείζεται κάποιες βασικές ιδιότητες της F-Logic [55]. Οι τριπλέτες RDF αναπαριστώνται σαν εκφράσεις σε F-logic ακόμη και εμφωλευμένες. Δεν υπάρχει όμως κάποια διάκριση μεταξύ των κανόνων και των επερωτήσεων εκτός από το ότι οι επερωτήσεις είναι κανόνες χωρίς κεφαλή (headless). Το αποτέλεσμα μιας επερωτήσης είναι οι δεσμεύσεις των ελεύθερων μεταβλητών. Η σημασιολογία του RDF δεν διερμηνεύεται από την αναπαράσταση της TRIPLE επομένως πρέπει αυτή να οριστεί μέσω ενός συνόλου κανόνων. Διαφορετικά μοντέλα RDF μπορούν να χρησιμοποιηθούν μέσα σε ένα σύνολο επερωτήσεων και κανόνων χρησιμοποιώντας το πρόθεμα "@model".

Οι επερωτήσεις TRIPLE αντίστοιχες με αυτές που παρουσιάστηκαν προηγουμένως είναι οι ακόλουθες:

```
FORALL X <- X[rdf:type -> Service;  
             hasServiceOrganization -> 'W3C';  
             hasServiceDescription -> 'bioinformatics']  
  
FORALL X <- X[rdf:type -> Service; hasInput/Output -> Y]  
AND Y[rdf:type -> Parameter; hasParameterDescription -> 'DNA  
sequence']
```

Η TRIPLE επίσης αποτελεί και το σύστημα υλοποίησης της γλώσσας επερωτήσεων και κανόνων¹⁹.

¹⁸<http://www.openrdf.org>

¹⁹<http://triple.semanticweb.org/>

2.6.7 QEL

Η γλώσσα Query Exchange Language (QEL) [68] έχει αναπτυχθεί ως μέρος του προγράμματος Edutella²⁰. Βασίζεται στην Datalog. Στο σύστημα Edutella, χρησιμοποιείται για να καταναίμει τις επερωτήσεις σε διάφορες αποθήκες RDF, όπου η επερώτηση μεταφράζεται στην γλώσσα που χρησιμοποιείται στην αντίστοιχη αποθήκη (π.χ. SQL, RDQL).

2.6.8 rdfDB

Η γλώσσα επερωτήσεων rdfDB [37] χρησιμοποιεί μια σύνταξη που μοιάζει με την σύνταξη της SQL και οι λειτουργίες έχουν αναπτυχθεί γύρω από την έννοια της τριπλέτας. Η απάντηση σε μια επερώτηση είναι ένα σύνολο από δεσμευμένες μεταβλητές για τις μεταβλητές που είναι παρούσες στα μέρη της επερώτησης. Το λεξιλόγιο RDF που χρησιμοποιείται μπορεί να ανήκει σε διαφορετικά σχήματα και επομένως υπάρχουν διαφορετικοί χώροι ονομάτων. Η γλώσσα rdfDB υποστηρίζει τόσο την ξεχωριστή δήλωση των URIs, όσο και την χρήση προθεμάτων των χώρων ονομάτων.

Η γλώσσα rdfDB αποτελεί μέρος του προγράμματος rdfDB²¹.

2.6.9 Versa

Η γλώσσα Versa είναι μια ειδικευμένη γλώσσα για δεδομένα RDF. Λειτουργεί σε ένα αφηρημένο μοντέλο γράφου του RDF, δηλαδή σε έναν γράφο που αποτελείται από κόμβους και ακμές με ετικέτες και επίσης παρέχει ένα μικρό σύνολο των βασικών τύπων δεδομένων. Μια επερώτηση στη γλώσσα Versa είναι ένας συνδυασμός κυριολεκτικών, πλοήγησης και φιλτραρίσματος, αναφορών σε μεταβλητές και κλήσεων συναρτήσεων. Οι πλοηγήσεις και τα φιλτραρίσματα αποτελούν εκφράσεις που κάνουν ταίριασμα προτύπων στο μοντέλο RDF. Το αποτέλεσμα μιας επερώτησης είναι μια λίστα των αντικειμένων/πόρων που ταιριάζουν με τα οριζόμενα κριτήρια. Η γλώσσα Versa επιτρέπει την χρήση μεταβλητών, συγκρίσεων και προκαθορισμένων συναρτήσεων.

Μια υλοποίηση της γλώσσας Versa μπορεί να βρεθεί στο 4Suite²², που αποτελεί ένα σύνολο XML και RDF εργαλείων.

Μια άλλη πρόσφατη ανασκόπηση σε διάφορες γλώσσες επερωτήσεων, περιλαμβανομένης της γλώσσας RDF, παρουσιάζεται στο παραδοτέο I4D1 στο πλαίσιο του Network of Excellence REVERSE²³.

²⁰<http://edutella.jxta.org/>

²¹<http://www.guha.com/rdfdb/>

²²<http://www.4suite.org>

²³<http://reverse.net/>

2.7 Συμπεράσματα

Στο Κεφάλαιο αυτό έγινε μια παρουσίαση του θεωρητικού υπόβαθρου που είναι απαραίτητο για την κατανόηση της εργασίας αυτής της διπλωματικής. Συγκεκριμένα, αναφερθήκαμε στην τεχνολογία των δικτύων ομότιμων κόμβων και ιδιαίτερα στους κατανεμημένους πίνακες κατακερματισμού. Επιπλέον, έγινε περιγραφή του πλαισίου RDF, του οποίου το μοντέλο δεδομένων χρησιμοποιήθηκε για την περιγραφή των δεδομένων που αποθηκεύονται στο σύστημα Atlas, το οποίο αποτελεί το αποτέλεσμα της δουλειάς που παρουσιάζεται σε αυτήν την διπλωματική εργασία.

Κεφάλαιο 3

Σχετικές εργασίες

Σε αυτό το κεφάλαιο παρουσιάζουμε τις σχετικές εργασίες που έχουν γίνει στις περιοχές που σχετίζονται άμεσα με αυτήν την διπλωματική. Αρχικά, παρουσιάζουμε τις σχετικές εργασίες που αφορούν στην διαχείριση RDF δεδομένων σε δίκτυα ομότιμων κόμβων. Έπειτα, γίνεται μια ανασκόπηση των εργασιών στην περιοχή της ανάκτησης σημασιολογικής πληροφορίας για υπηρεσίες του παγκόσμιου ιστού. Τέλος, κλείνουμε το κεφάλαιο αυτό με την παρουσίαση των σχετικών εργασιών για την ανακάλυψη πόρων και υπηρεσιών στο Πλέγμα (Grid).

3.1 Αποθήκευση και Υποβολή Επερωτήσεων RDF Δεδομένων σε Δίκτυα Ομότιμων Κόμβων

Ο συνδυασμός των τεχνολογιών του Σημασιολογικού Ιστού και των δικτύων ομότιμων κόμβων μπορεί να παρέχει ανάκτηση δεδομένων με ακρίβεια και αποδοτική αναζήτηση σε σενάρια κατανεμημένων εφαρμογών. Επομένως, έχει πρόσφατα αποτελέσει θέμα πολλών ερευνητικών άρθρων. Η πιο ολοκληρωμένη ανασκόπηση που έχει προκύψει από αυτή την έρευνα είναι το βιβλίο [95]. Τα δίκτυα ομότιμων κόμβων που είναι βασισμένα σε σχήμα επιτρέπουν πολύπλοκες και επεκτάσιμες περιγραφές πόρων, παρέχουν εξελιγμένες ευκολίες για επερωτήσεις και είναι δυνατόν να υποστηρίξουν ολοκλήρωση σχημάτων. Στις επόμενες παραγράφους παρουσιάζουμε κάποιες χαρακτηριστικές εργασίες σε αυτήν την ερευνητική περιοχή.

RDFPeers

Το άρθρο [19] μελετάει το πρόβλημα της αποτίμησης RDF επερωτήσεων πάνω από μια κατανεμημένη αποθήκη RDF δεδομένων με δυνατότητα κλιμάκωσης, που ονομάζεται RDFPeers. Το σύστημα RDFPeers υλοποιείται πάνω από το δίκτυο MAAN [18]. Το MAAN έχει δημιουργηθεί χρησιμοποιώντας την τεχνολογία των ΚΠΚ επεκτείνοντας το πρωτόκολλο του Chord [97] για την αποδοτική απάντηση

επερωτήσεων πολλών γνωρισμάτων (multi-attribute queries) καθώς και επερωτήσεων εύρους (range queries). Αυτό αποτέλεσε την πρώτη εργασία που ασχολήθηκε με επερωτήσεις RDF πάνω από ένα δομημένο δίκτυο

Στο σύστημα RDFPeers, κάθε κόμβος χρησιμοποιεί το μοντέλο δεδομένων του RDF για την δημιουργία περιγραφών πόρων, τις οποίες θέλει να κάνει διαθέσιμες στους υπόλοιπους κόμβους του δικτύου. Κάθε RDF τριπλέτα σε ένα έγγραφο RDF αποθηκεύεται τρεις φορές σε τρεις, ενδεχομένως διαφορετικούς, κόμβους. Αποθηκεύεται μια φορά στο διάδοχο κόμβο του αναγνωριστικού, που υπολογίζεται με τον κατακερματισμό της τιμής του υποκειμένου, και δύο ακόμη φορές χρησιμοποιώντας τις τιμές του κατηγορήματος και του αντικειμένου. Η συνάρτηση κατακερματισμού SHA-1 χρησιμοποιείται εάν η τιμή είναι μια συμβολοσειρά. Αν η τιμή είναι αριθμητική, τότε χρησιμοποιείται μια συνάρτηση κατακερματισμού που διατηρεί την διάταξη (order preserving hash function) και η οποία επιτρέπει την αποδοτική αποτίμηση των επερωτήσεων εύρους. Ένας κόμβος του συστήματος RDFPeers μπορεί να θέσει ατομικές επερωτήσεις για τριπλέτες, διαζευκτικές επερωτήσεις, επερωτήσεις εύρους καθώς και συζευκτικές ερωτήσεις πάνω στο κατηγορήμα. Στο άρθρο [19] προτείνουν διάφορους αλγορίθμους για την αποτίμηση αυτών των επερωτήσεων. Η γενική ιδέα αυτών των αλγορίθμων είναι ότι χρησιμοποιούν τα σταθερά μέρη των επερωτήσεων έτσι ώστε να δημιουργήσουν αναγνωριστικά που θα οδηγήσουν στους κόμβους που έχουν αποθηκευμένες τις σχετικές τριπλέτες. Επιπλέον, χρησιμοποιούν έναν απλό αλγόριθμο αντιγράφων για να βελτιώσουν την κατανομή του φόρτου. Τέλος, στο άρθρο [19] αναφέρονται κάποιες ιδέες σχετικά με σενάρια δημοσίευσης/συνδρομής στο σύστημα RDFPeers.

Οι ιδέες του [19] έχουν επηρεάσει σημαντικά την εργασία μας για απλές επερωτήσεις στο σύστημα Atlas, όπως αναφέρεται λεπτομερώς στο Κεφάλαιο 4. Επιπλέον, το άρθρο [61] έχει σημαντικά επεκτείνει και βελτιώσει τις ιδέες του [19] για να παρέχει λειτουργικότητα δημοσίευσης/συνδρομής για δεδομένα RDF πάνω από ΚΠΚ.

GridVine

Το GridVine [2] αποτελεί ένα σηματολογικό δίκτυο με δυνατότητα κλιμάκωσης που υποστηρίζει την δημιουργία τοπικών σχημάτων ενώ ταυτόχρονα παρέχει καθολική σηματολογική διαλειτουργικότητα. Ακολουθεί την αρχή της ανεξαρτησίας δεδομένων και διαχωρίζει το λογικό από το φυσικό στρώμα. Το λογικό στρώμα αποτελείται από το σηματολογικό στρώμα για την διαχείριση και την αντιστοίχιση δεδομένων και μεταδεδομένων σχημάτων, ενώ το φυσικό στρώμα αποτελείται από ένα δομημένο δίκτυο ομότιμων κόμβων που δρομολογεί μηνύματα με υψηλή απόδοση. Το δίκτυο αυτό χρησιμοποιείται για να υλοποιήσει διάφορες λειτουργίες στο λογικό στρώμα όπως αναζήτηση βάσει γνωρίσματος, διαχείριση σχημάτων και αντιστοίχιση σχημάτων.

Το GridVine χρησιμοποιεί το P-Grid [1] το οποίο αποτελεί ένα δομημένο δίκ-

το βασισμένο στις αρχές των ΚΠΚ. Οι κόμβοι στο GridVine έχουν την δυνατότητα να δημοσιεύουν τους διαθέσιμους πόρους δημιουργώντας RDF τριπλέτες (μεταδομένα). Μια RDF τριπλέτα αποθηκεύεται τρεις φορές στο δίκτυο χρησιμοποιώντας τρία διαφορετικά κλειδιά, βάσει των τιμών του υποκειμένου, του κατηγορήματος και του αντικειμένου όπως στο άρθρο [19]. Επομένως, τα είδη επερωτήσεων που μελετώνται στο [19] υποστηρίζονται επίσης από το σύστημα GridVine. Επιπλέον, υποστηρίζονται οι αναζητήσεις χρησιμοποιώντας το πρόθεμα μιας συμβολοσειράς π.χ. χρησιμοποιώντας την αρχή μιας συμβολοσειράς που αναπαριστά μια τιμή αντικειμένου. Αυτές οι αναζητήσεις υποστηρίζονται εύκολα χρησιμοποιώντας τους μηχανισμούς δρομολόγησης του P-Grid.

Το σύστημα GridVine επιτρέπει στους κόμβους να βρίσκουν νέα σχήματα από γνωστά βασικά σχήματα (χρησιμοποιώντας RDFS) παρέχοντας έτσι κληρονομικότητα σχήματος. Κάθε κόμβος έχει επίσης την δυνατότητα να δημιουργεί μια αντιστοίχιση μεταξύ δύο σχημάτων και έτσι δημιουργούνται οι σύνδεσμοι μετάφρασης μεταξύ των κόμβων του δικτύου (χρησιμοποιώντας OWL). Με αυτόν τον τρόπο οι επερωτήσεις στέλνονται από ένα σηματολογικό πεδίο σε ένα άλλο. Υπάρχουν δύο τεχνικές που χρησιμοποιούνται για την επίλυση μεταφραστικών συνδέσμων, η επαναληπτική και η αναδρομική επίλυση. Με την επαναληπτική επίλυση ο κόμβος που θέτει μια επερωτήση RDF προσπαθεί να βρει και να επεξεργαστεί όλους τους συνδέσμους της μετάφρασης μόνος του, ενώ με την αναδρομική επίλυση περισσότεροι του ενός κόμβοι χρησιμοποιούνται για την αποστολή της επερωτήσης και την μετάφραση της.

HyperCuP

Το άρθρο [86] προτείνει μια νέα τοπολογία γράφων για δίκτυα ομότιμων κόμβων που ονομάζεται HyperCup και επιτρέπει την αποδοτική αποστολή μηνυμάτων προς όλους (broadcast) και την λειτουργία αναζήτησης. Οι συγγραφείς περιγράφουν έναν αλγόριθμο αποστολής σε όλους τους κόμβους που εκμεταλλεύεται την προτεινόμενη τοπολογία για να φτάσει σε όλους τους κόμβους του δικτύου επιτυγχάνοντας το μικρότερο δυνατό αριθμό μηνυμάτων. Επιπλέον, δείχνουν πως μια καθολικά γνωστή ταξινόμηση μπορεί να χρησιμοποιηθεί για να οργανώσει τους κόμβους σε μια τοπολογία γράφων. Με αυτόν τον τρόπο επιτυγχάνεται αποδοτική αναζήτηση βάσει των εννοιών αυτής της ταξινόμησης.

Ο αλγόριθμος του HyperCup είναι δυνατόν να οργανώσει τους κόμβους ενός δικτύου ομότιμων κόμβων σε μια αναδρομική δομή γράφων από την οικογένεια γράφων Cayley. Η πιο γνωστή τοπολογία από την οικογένεια Cayley είναι ο υπερκύβος. Όλοι οι συμμετέχοντες κόμβοι είναι ίσοι (δεν υπάρχει κάποιος κεντρικός εξυπηρετητής ή υπερ-κόμβοι) και οργανώνονται σε μια τοπολογία υπερκύβου. Οι κόμβοι μπορούν να συνδεθούν και να αναχωρήσουν από το δίκτυο οποιαδήποτε στιγμή. Στην περίπτωση που κάποιοι κόμβοι απουσιάζουν, άλλοι κόμβοι στο δίκτυο θα απασχολήσουν περισσότερες της μιας θέσης στον υπερκύβο έτσι ώστε να διατηρηθεί η τοπολογία του υπερκύβου. Το HyperCup εγγυάται ότι μόνο χρήσιμα

μηνύματα θα δημιουργηθούν κατά την αποστολή σε όλους.

Οι συγγραφείς του άρθρου [86] παρατηρούν ότι στην περίπτωση των εφαρμογών του Σημαιολογικού Ιστού επιπλέον γνώση είναι διαθέσιμη και μπορεί κανείς να την εκμεταλλευτεί για την περαιτέρω βελτίωση της απόδοσης των δικτύων ομότιμων κόμβων. Επομένως, στο άρθρο [86] οι κόμβοι με ίδια ή παρόμοια ενδιαφέροντα οργανώνονται σε ομάδες ιδεών. Αυτές οργανώνονται αντίστοιχα σε μια τοπολογία υπερκύβου για να καταστεί δυνατή η δρομολόγηση σε συγκεκριμένες ομάδες στην τοπολογία. Οι ομάδες ιδεών είναι υπερκύβοι ή γραφήματα άστρων. Μια επερώτηση στέλνεται στις κατάλληλες ομάδες ιδεών (δηλαδή αυτές που αποθηκεύουν σχετικά δεδομένα σύμφωνα με την πληροφορία σχήματος) και προωθείται έπειτα σε όλους τους κόμβους της ομάδας.

Edutella

Ένα από τα πρώτα κατανεμημένα συστήματα για RDF δεδομένα που επηρέασαν την περιοχή είναι το σύστημα Edutella [70, 71]. Το Edutella παρέχει ένα πολύ γενικό πλαίσιο μεταδεδομένων RDF για εφαρμογές συστημάτων ομότιμων κόμβων.

Στα άρθρα [70, 71], θεωρούν ότι μια τοπολογία υπερκόμβων είναι η κατάλληλη τοπολογία για δίκτυα ομότιμων κόμβων βασισμένα σε πληροφορία σχήματος. Επομένως, μια τέτοια τοπολογία χρησιμοποιείται στο σύστημα Edutella. Σε ένα δίκτυο Edutella υπάρχουν δύο είδη κόμβων: οι υπερκόμβοι και οι πελάτες. Οι υπερκόμβοι οργανώνονται βάσει της τοπολογίας HyperCup, ενώ οι πελάτες είναι συνδεδεμένοι με τους υπερκόμβους βάσει μιας τοπολογίας άστρου. Κάθε πελάτης συνδέεται μόνο με έναν υπερκόμβο. Οι υπερκόμβοι χρησιμοποιούνται για να διαχειριστούν αποδοτικά όλες τις αιτήσεις των πελατών. Κατά την εγγραφή ενός πελάτη, αυτός παρέχει στον υπερκόμβο του τα μεταδεδομένα του, δηλαδή μια περιγραφή των μεταδεδομένων που έχουν δημιουργηθεί από τον πελάτη (υποστηριζόμενο σχήμα, χρησιμοποιούμενες τιμές κ.ά.). Τα πραγματικά μεταδεδομένα παραμένουν στον κόμβο πελάτη. Κάθε υπερκόμβος αποθηκεύει πληροφορία σχετική με την χρήση των μεταδεδομένων σε κάθε πελάτη με τον οποίο είναι άμεσα συνδεδεμένος. Επιπλέον, κάθε υπερκόμβος αποθηκεύει πληροφορία σχήματος για τους άμεσους γείτονες υπερκόμβους (δηλαδή, μια περιγραφή των μεταδεδομένων που χρησιμοποιούνται από τους πελάτες τους). Αυτή η πληροφορία χρησιμοποιείται για να δρομολογηθούν οι επερωτήσεις μόνο σε σχετικούς υπερκόμβους και πελάτες.

Οι ερευνητές του Edutella έχουν επίσης επικεντρωθεί στην ολοκλήρωση σχημάτων σε ιεραρχικά δίκτυα ομότιμων κόμβων βασισμένα σε σχήμα για δεδομένα RDF. Η γενική ιδέα είναι ότι οι υπερκόμβοι χρησιμοποιούνται σαν ενδιάμεσοι, δηλαδή διατηρούνε την πληροφορία αντιστοιχισμένη στο σχήμα, ενώ δρομολογούνε επερωτήσεις στο δίκτυο με στόχο να τις μεταφράσουν.

Ένα σενάριο εφαρμογής του συστήματος Edutella είναι το Elena [92], μια υποδομή μεσολαβητών για εκπαιδευτικές υπηρεσίες που ανακοινώνονται και

προσφέρονται με ηλεκτρονικά μέσα. Το Elena είναι ένα πρόγραμμα EU/IST που ακολουθεί το όραμα της δημιουργίας και της δοκιμής των ονομαζόμενων Smart Spaces for LearningTM. Αυτά αποτελούν ανοικτά περιβάλλοντα όπου οι μαθητευόμενοι μπορούν να επιλέξουν εκπαιδευτικές υπηρεσίες από ετερογενείς πηγές.

Δημοσίευση/Συνδρομή για δίκτυα ομότιμων κόμβων RDF δεδομένων

Στο άρθρο [24], περιγράφεται η παροχή λειτουργικότητας δημοσίευσης / συνδρομής σε ένα δίκτυο ομότιμων κόμβων που αποθηκεύει RDF δεδομένα. Αυτό το άρθρο εκμεταλλεύεται την εμπειρία που έχει αποκτηθεί από τα συστήματα Edutella και P2P-DIET. Οι συγγραφείς υποθέτουν ότι το σύστημα διαχειρίζεται διάφορους ψηφιακούς πόρους που αναγνωρίζονται από ένα URL και περιγράφονται με το μοντέλο δεδομένων του RDF. Οι κόμβοι του δικτύου μπορούν να δημοσιεύσουν τέτοιους πόρους ή να κάνουν συνδρομή με συνεχείς επερωτήσεις για να ειδοποιηθούν όταν σχετικοί πόροι εισάγονται στο δίκτυο. Οι συγγραφείς ορίζουν μια γλώσσα (*typed first-ordered language* \mathcal{L}) που αποτελεί ένα υποσύνολο της γλώσσας Query Exchange Language (QEL) και χρησιμοποιείται για να εκφράσει τις συνδρομές. Η γλώσσα \mathcal{L} είναι αρκετά παρόμοια με αυτήν που παρουσιάζεται στο άρθρο [19]. Η QEL είναι μια γλώσσα, που έχει δημιουργηθεί χρησιμοποιώντας την γλώσσα Datalog, για δεδομένα RDF και χρησιμοποιείται στο δίκτυο ομότιμων κόμβων Edutella [70].

Στο άρθρο [24], το δίκτυο οργανώνεται όπως το σύστημα Edutella [70]. Οι πελάτες στέλνουν διαφημίσεις στους υπερκόμβους. Οι διαφημίσεις χρησιμοποιούνται για να περιορίσουν τα σχήματα που χρησιμοποιούνται από έναν πελάτη, τα γνωρίσματα αυτών των σχημάτων ή ακόμη και τις τιμές αυτών των γνωρισμάτων. Οι διαφημίσεις αποκτώνται από τους υπερκόμβους, οι οποίοι τις εκμεταλλεύονται για να δρομολογήσουν αποδοτικά τις συνδρομές προς τα μέρη του δικτύου όπου είναι πιο πιθανό να δημοσιευτούν σχετικοί πόροι στο μέλλον. Κάθε υπερκόμβος αποθηκεύει πληροφορία για αυτές τις διαφημίσεις, όχι μόνο από τους πελάτες του, αλλά και από τους γείτονες υπερκόμβους με στόχο την επίτευξη της αποδοτικής δρομολόγησης των επερωτήσεων στο ιεραρχικό δίκτυο. Αυτή η πληροφορία συνθέτει τα *ευρετήρια δρομολόγησης διαφημίσεων* (advertisement routing indices). Επιπλέον, στο άρθρο [24] παρουσιάζονται αλγόριθμοι για τον αποδοτικό χειρισμό καταστάσεων όπου οι πελάτες δεν είναι συνδεδεμένοι, είτε όταν δημιουργείται μια ειδοποίηση για αυτούς, είτε όταν ένας άλλος πελάτης θέλει να ανακτήσει έναν πόρο από αυτούς. Στις δύο αυτές περιπτώσεις, το ιεραρχικό δίκτυο χρησιμοποιείται για να αποθηκεύσει προσωρινά την απαιτούμενη πληροφορία στους κατάλληλους υπερκόμβους.

Αποτίμηση των top- k επερωτήσεων

Οι συγγραφείς του άρθρου [69], εμπνεύστηκαν από την επιτυχία των αλγορίθμων κατάταξης στις μηχανές αναζήτησης στο διαδίκτυο, καθώς και των αλγορίθμων top- k ανάκτηση στις βάσεις δεδομένων. Προτείνουν έναν κατανεμημένο αλγόριθμο αποτίμησης για δίκτυα ομότιμων κόμβων, ο οποίος ανακτάει τις top- k , δηλαδή τις k πιο σημαντικές, απαντήσεις για κάθε επερώτηση. Αυτή η εργασία μπορεί να θεωρηθεί σημαντική, ειδικά όσον αφορά την επεκτασιμότητα. Αυτό προκύπτει από το γεγονός ότι αυξάνοντας το μέγεθος του δικτύου, αυξάνεται και το πλήθος των απαντήσεων δεδομένων επερωτήσεων αφού το δίκτυο έχει πιο πολλά δεδομένα.

Ο αλγόριθμος που προτάθηκε στο άρθρο [69] επιστρέφει τα k πιο σχετικά αποτελέσματα χωρίς να βασίζεται σε κάποιου είδους κεντρική γνώση και χωρίς την χρήση κάποιου ολοκληρωμένου κατανεμημένου ευρετηρίου. Η αρχιτεκτονική που θεωρείται για το δίκτυο ομότιμων κόμβων είναι η ιεραρχική αρχιτεκτονική του δικτύου Edutella, που έχει ήδη περιγραφεί. Ο προτεινόμενος αλγόριθμος επιτρέπει την βελτιστοποίηση της κατανομής των επερωτήσεων και της δρομολόγησης τους. Ο αλγόριθμος χρησιμοποιεί μεθόδους κατάταξης με στόχο να μειώσει το συνολικό πλήθος των απαντήσεων που επιστρέφονται. Επιπλέον, έχει την δυνατότητα να επιστρέψει σχεδόν ακριβείς απαντήσεις εάν το σύνολο των ακριβών απαντήσεων είναι κενό. Κάθε κόμβος υπολογίζει τοπικά τις κατατάξεις για μια συγκεκριμένη επερώτηση που λαμβάνει και στέλνει τα αποτελέσματα στον υπερκόμβο με τον οποίο είναι συνδεδεμένος. Κάθε υπερκόμβος ενώνει τα αποτελέσματα από τους κόμβους και τους γείτονες υπερκόμβους και προωθεί μόνο τα *καλύτερα* αποτελέσματα στον υπερκόμβο που είναι ο συνδεδεμένος ο κόμβος που έθεσε την επερώτηση. Αυτός ο υπερκόμβος ακολουθεί την ίδια διαδικασία και τις στέλνει πίσω στον πρώτο κόμβο. Ενώ τα αποτελέσματα δρομολογούνται μεταξύ των υπερκόμβων, ο αλγόριθμος διατηρεί στατιστικά σχετικά με το ποιοι κόμβοι επέστρεψαν τα καλύτερα αποτελέσματα. Χρησιμοποιεί αυτήν την πληροφορία για την κατανομή των επερωτήσεων μόνο στους *καλύτερους* κόμβους. Ο αλγόριθμος ελαχιστοποιεί το μέγεθος του συνόλου απαντήσεων και επομένως και την κίνηση του δικτύου.

SQPeer

Το σύστημα SQPeer [56] είναι ένα σύστημα για την δρομολόγηση και τον σχεδιασμό σύνθετων επερωτήσεων σε συστήματα βάσεων δεδομένων πάνω από δίκτυα ομότιμων κόμβων εκμεταλλευόμενο τα σχήματα των κόμβων. Στο σύστημα SQPeer, κάθε κόμβος παρέχει περιγραφές RDF/S για πόρους που θέλει να κάνει διαθέσιμους στο δίκτυο. Οι κόμβοι που χρησιμοποιούν το ίδιο σχήμα ανήκουν ουσιαστικά στο ίδιο υποκείμενο δίκτυο [105]. Οι επερωτήσεις στο SQPeer δημιουργούνται στην γλώσσα RQL, [52] σύμφωνα με το σχήμα RDF που υποστηρίζει ο κάθε κόμβος. Επιπλέον, κάθε κόμβος μπορεί να *διαφημίσει*

το περιεχόμενο της τοπικής βάσης του χρησιμοποιώντας την γλώσσα RVL [64]. Ο προτεινόμενος αλγόριθμος δρομολόγησης και επεξεργασίας επερωτήσεων μπορεί να ανακαλύψει τους σχετικούς κόμβους που μπορούν να απαντήσουν κάθε επερώτηση και παράγει πλάνα επερωτήσεων λαμβάνοντας υπόψη τα στατιστικά της κατανομής δεδομένων κ.ά. Το σύστημα SQPeer μπορεί να υλοποιηθεί χρησιμοποιώντας δύο εναλλακτικές αρχιτεκτονικές. Η μια είναι ένα υβριδικό δίκτυο ομότιμων κόμβων ενώ η δεύτερη είναι ένα δομημένο δίκτυο βασισμένο στους ΚΠ-Κ. Τα θέματα που αφορούν την υλοποίηση αυτού του συστήματος αφορούν την μελλοντική εργασία των συγγραφέων.

SWAP

Το πρόγραμμα SWAP¹ έχει μελετήσει τον συνδυασμό των τεχνολογιών του Σημαιολογικού Ιστού και των δικτύων ομότιμων κόμβων. Το όραμα αυτού του προγράμματος είναι να επιτρέψει στους κόμβους να διατηρούν προσωπικές όψεις της γνώσης του δικτύου και ταυτόχρονα να μπορούν να μοιράζονται αυτήν την γνώση με το υπόλοιπο δίκτυο.

Στο άρθρο [34], οι συγγραφείς θεωρούν μια τοπολογία δικτύων ομότιμων κόμβων όπου όλοι οι κόμβοι είναι ίσοι και προτείνουν ένα μοντέλο βασισμένο σε RDF μεταδεδομένα για την κωδικοποίηση της σημασιολογίας. Η σημασιολογία συνδυάζει ιδιότητες των οντολογιών με πλούσια μεταδεδομένα για την προέλευση της πληροφορίας και την αξιοπιστία των πηγών του κάθε κόμβου. Επιπλέον, προτείνουν διάφορες μεθόδους για την αυτόματη δημιουργία και την ολοκλήρωση των περιεχομένων από τοπικές αλλά και από εξωτερικές πηγές δεδομένων. Κάθε κόμβος του δικτύου πρέπει να ικανοποιεί συγκεκριμένες απαιτήσεις σχετικές με την ολοκλήρωση, την διαφορετικότητα, την ασυνέπεια και την ασφάλεια πληροφορίας. Με αυτόν τον τρόπο, ο κόμβος θα μπορεί να προσπελάσει διαφορετικές πηγές πληροφοριών. Το μοντέλο μεταδεδομένων επιτρέπει την διατήρηση (caching) της πληροφορίας και επομένως βελτιώνει την αποδοτικότητα του δικτύου. Οι επερωτήσεις κατανέμονται αναλόγως του περιεχομένου τους. Όταν ένας κόμβος λαμβάνει μια επερώτηση από έναν άλλο κόμβο, προσπαθεί να την απαντήσει ή να την προωθήσει. Αυτήν η απόφαση εξαρτάται από την γνώση για τους υπόλοιπους κόμβους.

Bibster

Το Bibster [42] είναι ένα σύστημα υλοποιημένο σαν ένα στιγμιότυπο μιας πλατφόρμας SWAP που βασίζεται στο JXTA². Αποτελεί ένα δίκτυο ομότιμων κόμβων για την ανταλλαγή βιβλιογραφικών εγγραφών (εγγραφών BibTex). Το Bibster επιτρέπει την αναζήτηση βιβλιογραφικών εγγραφών χρησιμοποιώντας αναζήτηση

¹<http://swap.semanticweb.org/>

²<http://www.jxta.org>

με λέξεις-κλειδιά, αλλά και πιο προχωρημένη αναζήτηση. Το Bibster χρησιμοποιεί οντολογίες για την εισαγωγή δεδομένων, την δημιουργία, την δρομολόγηση επερωτήσεων και την επεξεργασία των απαντήσεων. Συγκεκριμένα, χρησιμοποιεί την οντολογία SWRC³ και την ιεραρχία θεμάτων της ACM⁴. Κάθε κόμβος διαχειρίζεται μια τοπική αποθήκη δεδομένων RDF με βιβλιογραφικά δεδομένα και χρησιμοποιεί την ιεραρχία θεμάτων για να διαφημίσει μια σημασιολογική περιγραφή των δεδομένων (την ειδικότητα του) στο δίκτυο ομότιμων κόμβων. Επιπλέον, μπορεί να θέσει επερωτήσεις στην γλώσσα SeRQL [16]. Κατά την επεξεργασία επερωτήσεων, ένας κόμβος πρώτα αποτιμάει την επερώτηση τοπικά και μετά αποφασίζει εάν πρέπει να προωθηθεί. Αυτή η απόφαση βασίζεται στο θέμα της επερώτησης που ορίζει την απαραίτητη ειδικότητα για να απαντηθεί η επερώτηση και επομένως η επερώτηση θα προωθηθεί στους κόμβους με την κατάλληλη ειδικότητα.

SERSE

Ένα άλλο άρθρο που μελετάει την αναζήτηση στον σημασιολογικό ιστό είναι το [103]. Το άρθρο [103] παρουσιάζει το σύστημα SERSE (Semantic Routing System) που αποτελεί ένα πολυπρακτορικό σύστημα για την αναζήτηση στον ιστό. Αυτό το σύστημα συνδυάζει διαφορετικές τεχνολογίες όπως τα δίκτυα ομότιμων κόμβων, τις οντολογίες και την τεχνολογία των πολυπρακτορικών συστημάτων. Στο SERSE, οι πράκτορες δρομολόγησης έχουν τις ίδιες δυνατότητες και ευθύνες με τους κόμβους του δικτύου ομότιμων κόμβων. Οι διαθέσιμοι πόροι επισημειώνονται με σημασιολογική πληροφορία και οι πράκτορες είναι υπεύθυνοι για την ανάκτηση των πόρων χρησιμοποιώντας τις περιγραφές τους (επισημειώσεις). Οι περιγραφές των πόρων καθορίζουν μια σημασιολογική δομή στο δίκτυο ομότιμων κόμβων, όπου ο κάθε κόμβος μπορεί να επικοινωνήσει μόνο με τους κόμβους που ανήκουν στην ίδια *σημασιολογική γειτονιά* (semantic neighborhood). Δεν διατηρείται κάποια καθολική πληροφορία του δικτύου, συγκεκριμένα κάθε κόμβος γνωρίζει ακριβώς τους άμεσους γείτονες του και επιπλέον οι πράκτορες δεν μπορούν να μεταδώσουν μηνύματα σε όλο το δίκτυο.

Μια περίληψη των εργασιών που παρουσιάστηκαν δίνεται στον Πίνακα 3.1. Για κάθε άρθρο της ανασκόπησης αναφέρονται: η αρχιτεκτονική του δικτύου ομότιμων κόμβων που χρησιμοποιείται, το σημασιολογικό μοντέλο δεδομένων, η γλώσσα επερωτήσεων που χρησιμοποιείται, καθώς και τα σενάρια επερωτήσεων υποστηρίζονται.

³<http://www.semanticweb.org/ontologies/swrc-onto-2001-12-11.daml>

⁴<http://www.acm.org/class/1998/>

Εργασία	Αρχιτεκτονική	Μοντέλο δεδομένων	Γλώσσα επερωτήσεων	Σενάριο επερωτήσεων
<i>RDFPeers</i> [19]	MAAN (ΚΠΚ)	RDF	Ατομικές επερωτήσεις τριπλετών, διαζευκτικές και επερωτήσεις εύρους, Συζευκτικές επερωτήσεις στο κατηγορημα	Απλές και συνεχείς (ατομικές επερωτήσεις)
<i>GridVine</i> [2]	P-Grid (ΚΠΚ)	RDF/S + OWL	RDQL	Απλές
<i>HyperCuP</i> [86]	τοπολογία HyperCuP (Χωρίς υπερκόμβους και χωρίς κεντρικό εξυπηρετητή)	Οντολογίες	Βασισμένο σε οντολογία (π.χ., όνομα τάξης)	Απλές
<i>Edutella</i> [70, 71]	Υπερκόμβοι (τοπολογία HyperCuP)	RDF/S	QEL	Απλές
<i>Δημοσίευση /συνδρομή</i> [24]	Ιεραρχικό (τοπολογία HyperCuP)	RDF/S	$\mathcal{L} \subseteq \text{QEL}$	Απλές, Δημοσίευση /Συνδρομή
<i>Αποτίμηση top-k επερωτήσεων</i> [69]	δεν απαιτείται συγκεκριμένη τοπολογία	RDF/S	Δεν ορίζεται τυπικά (μπορεί να είναι όνομα κλάσης, λέξη-κλειδί ή συνδυασμός)	Απλές
<i>SQPeer</i> [56]	Υβριδικό ή δομημένο δίκτυο ομότιμων κόμβων	RDF/S	RQL/RVL	Απλές
<i>SWAP</i> [34]	-	RDF/S +OWL	SeRQL	Απλές
<i>SERSE</i> [103]	Δίκτυο ομότιμων κόμβων	Οντολογίες	Βασισμένη σε οντολογίες	Απλές

Πίνακας 3.1: Δίκτυα ομότιμων κόμβων βασισμένα στο πλαίσιο RDF

3.2 Ανακάλυψη και Ταίριασμα Υπηρεσιών του Παγκόσμιου Ιστού

Στο Κεφάλαιο 5 θα περιγράψουμε πως θα χρησιμοποιηθεί το σύστημα Atlas για την ανακάλυψη υπηρεσιών του παγκόσμιου ιστού. Για να μπορέσουμε να κατανοήσουμε την αξία του Atlas και να την συγκρίνουμε με την σχετική δουλειά, αυτό το κεφάλαιο αποτελεί μια ανασκόπηση σχετικών δημοσιεύσεων στην περιοχή της ανακάλυψης υπηρεσιών του παγκόσμιου ιστού.

Οι εργασίες αυτής της ανασκόπησης έχουν οργανωθεί αναλόγως με το αν χρησιμοποιούν έναν κεντρικοποιημένο ή έναν κατακερματισμένο αλγόριθμο για την ανακάλυψη και το ταίριασμα των υπηρεσιών.

Κεντρικοποιημένες προσεγγίσεις

Η πρόταση του *Universal Description Discovery and Integration (UDDI)* [107] ορίζει ένα πλαίσιο για την δημοσίευση και την ανακάλυψη διαθέσιμων υπηρεσιών του παγκόσμιου ιστού χρησιμοποιώντας ένα μητρώο που ονομάζεται *“business registry”*. Η αναπαράσταση των δεδομένων και των μεταδεδομένων για υπηρεσίες του παγκόσμιου ιστού είναι σημαντική για τους σκοπούς του UDDI. Οι επιχειρήσεις και οι παροχείς υπηρεσιών αναπαριστώνται με την δομή *“BusinessEntity”* η οποία περιλαμβάνει πληροφορίες όπως, όνομα, πολυγλωσσικές περιγραφές, πληροφορία επικοινωνίας και πληροφορία ταξινόμησης. Η δομή *“BusinessService”* χρησιμοποιείται για να αναπαραστήσει ένα σύνολο σχετικών υπηρεσιών που παρέχονται από μία *“BusinessEntity”*. Οι υπηρεσίες του παγκόσμιου ιστού αναπαριστώνται με το στοιχείο *“bindingTemplate”* που παρέχει τις τεχνικές πληροφορίες που χρειάζονται για την αλληλεπίδραση με την υπηρεσία, για παράδειγμα ένα σημείο πρόσβασης (access point). Τέλος, τα τεχνικά μοντέλα (technical models (tModels)) αποτελούν γενικές έννοιες του UDDI για την περιγραφή κάθε είδους προδιαγραφών. Για παράδειγμα, ένα πρωτόκολλο αλληλεπίδρασης μπορεί να αναπαρασταθεί με ένα tModel.

Ένα UDDI μητρώο παρέχει τα κατάλληλα APIs για να ταξινομήσει, να καταλογογραφήσει και να διαχειριστεί υπηρεσίες του παγκόσμιου ιστού με στόχο την ανακάλυψη και χρήση τους. Το UDDI αποτελείται από δύο βασικά APIs: το API αναζήτησης (Inquiry API) που επιτρέπει την εύρεση και την ανάκτηση πληροφοριών για εγγραφές του UDDI μητρώου και το API δημοσίευσης (Publication API) για την δημοσίευση και την ανανέωση των πληροφοριών που περιέχονται σε ένα UDDI μητρώο.

Το API αναζήτησης παρέχει τρεις μορφές επερωτήσεων οι οποίες χρησιμοποιούνται συνήθως στην χρήση μητρώων. Τα τρία μοτίβα που υποστηρίζονται είναι το *browse* μοτίβο, το *drill-down* μοτίβο και το *invocation* μοτίβο. Πολλαπλά κριτήρια αναζήτησης μπορούν να εκφραστούν με συζευκτική ή διαζευκτική μορφή. Το UDDI χρησιμοποιεί επίσης τον τελεστή LIKE της SQL99 που επιτρέπει το ταίριασμα συμβολοσειρών με την χρήση των ονομαζόμενων wildcards.

Τα UDDI μητρώα υλοποιούνται συνήθως σαν υπηρεσίες του παγκόσμιου ιστού και επομένως η αλληλεπίδραση μαζί τους γίνεται με την χρήση XML, WSDL και SOAP.

Στο άρθρο [109] των Wang και Stroulia παρουσιάζεται μία επέκταση του UDDI με ένα σύνολο μεθόδων αποτίμησης ομοιότητας χρησιμοποιώντας την γλώσσα WSDL. Η βασική ιδέα αυτής της δουλειάς είναι ότι ένας πιθανός τρόπος ερώτησης ενός UDDI μητρώου είναι η ‘επερωτήση με παράδειγμα’ (“query by example”). Αυτό επιτυγχάνεται παρέχοντας μια μερική περιγραφή σε WSDL της υπηρεσίας που επιθυμούμε. Το υποψήφιο σύνολο υπηρεσιών ανακτάται εφαρμόζοντας μια τυπική προσέγγιση ανάκτησης πληροφοριών (Information Retrieval) βασισμένη στο μοντέλο διανυσματικών χώρων (vector space). Έπειτα χρησιμοποιείται ένας αλγόριθμος ταιριάσματος δομών WSDL για να βελτιωθεί η ποιότητα του συνόλου που ανακτήθηκε. Για να δουλέψει το πρώτο βήμα οι περιγραφές των υπηρεσιών και οι επερωτήσεις αναπαριστώνται σαν διανύσματα λέξεων. Κάθε λέξη έχει ένα βάρος ανάλογο της συχνότητας εμφάνισης σε ένα έγγραφο και αντιστρόφως ανάλογο του πλήθους των εγγράφων στα οποία εμφανίζεται. Ένας κοινός δείκτης για την μέτρηση της σημαντικότητας μιας λέξης με αυτόν τον τρόπο στην ανάκτηση πληροφοριών είναι ο δείκτης TF/IDF. Σε μια επόμενη δημοσίευση [110], οι Wang και Stroulia επεκτείνουν το μοντέλο του [109] περιλαμβάνοντας ταιρίασμα με χρήση του λεξικού Wordnet. Έτσι, ενισχύουν τους αλγόριθμους τους με περισσότερη σηματολογική πληροφορία. Οπότε, εκτός από τις ακριβείς λέξεις που βρίσκονταν στις περιγραφές των υπηρεσιών και στις επερωτήσεις μπορούν να χρησιμοποιήσουν επιπλέον συνώνυμα, υπερώνυμα και υπόνομα αυτών των λέξεων, τα οποία παρέχει η λεξικολογική βάση του Wordnet. Ο αλγόριθμος ταιριάσματος δομών του [109] βασίζεται στην υπόθεση ότι αν δύο υπηρεσίες είναι εννοιολογικά παρόμοιες, είναι πιο πιθανό να είναι και δομικά παρόμοιες. Η σύγκριση δύο αρχείων WSDL περιλαμβάνει την σύγκριση των τύπων δεδομένων, των μηνυμάτων των υπηρεσιών και των λειτουργιών των υπηρεσιών. Το άρθρο [110] επεκτείνει τον αλγόριθμο ταιριάσματος δομών του [109] με ένα σηματολογικό αλγόριθμο που βασίζεται τόσο στην συντακτική όσο και στην σηματολογική ομοιότητα χρησιμοποιώντας το λεξικό Wordnet.

Οι Degwekar κ.ά. έχουν επεκτείνει την γλώσσα WSDL έτσι ώστε να περιλαμβάνει δομές που να επιτρέπουν την χρήση περιορισμών. Επιπλέον, θέλουν να μπορούν να εκφραστούν επιθυμητές τιμές για συγκεκριμένα γνωρίσματα υπηρεσιών σε μία αίτηση υπηρεσίας. Για τον ορισμό περιορισμών χρησιμοποιούν μια γλώσσα που επιτρέπει την έκφραση περιορισμών (constraint-based requirement specification language) από την εργασία [101]. Η γλώσσα WSDL επεκτείνεται έτσι ώστε να συμπεριλάβει τις δηλώσεις αυτής της γλώσσας σε μορφή XML. Πληροφορία σχετική με τους περιορισμούς περιλαμβάνεται στο έγγραφο υλοποίησης της υπηρεσίας του WSDL specification. Στο άρθρο [31] έχουν σχεδιάσει έναν αλγόριθμο ταιριάσματος περιορισμών βασισμένο στην επέκταση ενός υπάρχοντος εργαλείου [100]. Αυτό το εργαλείο ονομάζεται *constraint satisfaction processor (CSP)* και παίρνει ως είσοδο τα δύο αντικείμενα που θέλουμε να ταιριάξουμε ενώ

δίνει ως έξοδο τον βαθμό του ταιριάσματος. Στην εργασία αυτή, το έχουν συνδυάσει με κάποια επιπλέον εργαλεία όπως το μητρώο UDDI της IBM με στόχο τον σχηματισμό ενός μεσίτη που χρησιμοποιεί περιορισμούς (constraint-based broker). Δεδομένου ότι τα μητρώα UDDI δεν περιλαμβάνουν πληροφορία για περιορισμούς, στο άρθρο [31] χρησιμοποιούν ένα ξεχωριστό μητρώο για να τους αποθηκεύουν. Επομένως, η διαδικασία ανακάλυψης αποτελείται τώρα από δύο φάσεις. Αρχικά, γίνεται επικοινωνία με το UDDI μητρώο και επιστρέφεται ένα σύνολο υπηρεσιών του παγκόσμιου ιστού βάσει των κριτηρίων αναζήτησης. Έπειτα, χρησιμοποιώντας αυτό το σύνολο υπηρεσιών που επιστράφηκε, λαμβάνει χώρα το ταιρίασμα περιορισμών επικοινωνώντας με το μητρώο των περιορισμών και χρησιμοποιώντας τον CSP.

Οι Bilgin και Singh [14] παρουσιάζουν μια μέθοδο που λαμβάνει υπόψη την αναμενόμενη συμπεριφορά μιας υπηρεσίας, δηλαδή την ποιότητα της υπηρεσίας (quality of service) και όχι την λειτουργική περιγραφή αυτής της υπηρεσίας. Στην δουλειά του [14] αναπτύσσεται μια αποθήκη πληροφοριών που χρησιμοποιεί την γλώσσα DAML και επεκτείνει τα UDDI μητρώα έτσι ώστε να υποστηρίζουν γνωρίσματα ποιότητας υπηρεσιών. Για να μπορέσουν να πραγματοποιήσουν τέτοια μητρώα, δημιούργησαν μια γλώσσα επερωτήσεων και χειρισμού υπηρεσιών η οποία ονομάζεται Semantic Web Services Query and Manipulation Language (SWSQL). Η SWSQL είναι βασισμένη σε μια γλώσσα επερωτήσεων για δεδομένα DAML και η οποία έχει προταθεί από το άρθρο [7]. Επιπλέον, γίνεται χρήση μιας απλής οντολογίας για κατηγορίες υπηρεσιών και γνωρίσματα ποιότητας υπηρεσιών. Η SWSQL μπορεί να χρησιμοποιηθεί για επερωτήσεις τόσο στην οντολογία, δηλαδή στα μεταδεδομένα των γνωρισμάτων ποιότητας μίας υπηρεσίας του παγκόσμιου ιστού, όσο και στις τιμές που αποτελούν τα πραγματικά δεδομένα. Η οντολογία αποθηκεύεται σαν τριπλέτες στην μνήμη ενώ τα δεδομένα της περιγραφής της υπηρεσίας αποθηκεύονται σε μια σχεσιακή βάση δεδομένων. Τέλος, έχουν υλοποιήσει έναν μετατροπέα από SWSQL σε SQL (SWSQLtoSQL mapper) για την μετατροπή μιας SWSQL επερώτησης σε μια SQL επερώτηση.

Οι Sycara κ.ά. [102] υιοθετούν την οντολογία DAML-S (η οποία έχει πλέον εξελιχθεί στην οντολογία OWL-S) για την προσθήκη *σηματολογικής* πληροφορίας σε υπηρεσίες παγκόσμιου ιστού. Η οντολογία DAML-S περιγράφει μία υπηρεσία του παγκόσμιου ιστού χρησιμοποιώντας τα ακόλουθα στοιχεία: το προφίλ της υπηρεσίας (DAML-S service profile), το μοντέλο διεργασιών (DAML-S process model) και την υλοποίηση (DAML-S Grounding). Η βασική ιδέα της DAML-S είναι να αναπαραστήσει τις υπηρεσίες του παγκόσμιου ιστού με ένα σύνολο των δυνατοτήτων που προσφέρουν και που περιγράφονται από τις εισόδους που απαιτούν (inputs), τις εξόδους που παράγουν (outputs), τις συνθήκες (pre-conditions) και τα αποτελέσματα (post-conditions). Στο άρθρο [102] επεκτείνεται ένα UDDI μητρώο με το εργαλείο DAML-S/UDDI Matchmaker που χρησιμοποιεί την περιγραφή των δυνατοτήτων της DAML-S για να ενισχύσει το UDDI με την δυνατότητα ταιριάσματος δυνατοτήτων (capability matching). Η βασική ιδέα πίσω από την διαδικασία ταιριάσματος είναι η εξής: αναγνωρίζεται ταιρίασμα

μεταξύ μια δημοσίευσης και μιας αίτησης όταν η δημοσιευμένη υπηρεσία μπορεί να χρησιμοποιηθεί στην θέση της ζητούμενης υπηρεσίας. Το σηματολογικό ταίριασμα χρησιμοποιεί τις DAML οντολογίες με τις οποίες έχουν δημιουργηθεί οι δημοσιεύσεις και οι αιτήσεις. Η μηχανή ταιριάσματος (matching engine) χρησιμοποιεί μια μηχανή συλλογισμού (DAML+OIL reasoner) για να αναγνωρίσει σηματολογικά ταιριάσματα εκμεταλλευόμενη την ιεραρχία που παρέχεται. Η διαδικασία ταιριάσματος αναγνωρίζει ένα ταίριασμα όταν οι εξόδοι μιας δημοσίευσης είναι ισοδύναμες ή πιο γενικές από τις εξόδους μιας αίτησης και οι εισόδοι μιας αίτησης είναι ισοδύναμες ή πιο γενικές από τις εισόδους μιας δημοσίευσης. Εκτός από την μηχανή ταιριάσματος (DAML-S/UDDI Matchmaker), στο άρθρο [102] ανέπτυξαν και την εικονική μηχανή DAML-S Virtual Machine, που αποτελείται από ένα σύνολο στοιχείων λογισμικού που επιτρέπουν την αλληλεπίδραση των υπηρεσιών του παγκόσμιου ιστού χρησιμοποιώντας το μοντέλο διεργασιών της DAML-S. Τέλος, το άρθρο [102] περιλαμβάνει μια συζήτηση που αφορά στο πως η μηχανή ταιριάσματος (DAML-S/UDDI Matchmaker) και η εικονική μηχανή (DAML-S Virtual Machine) μπορούν να παρέχουν την βάση για την σύνθεση υπηρεσιών του παγκόσμιου ιστού (Web services composition).

Οι Li και Horrocks έχουν επίσης επιλέξει την οντολογία DAML-S για την αναπαράσταση δημοσιεύσεων των υπηρεσιών και των επερωτήσεων. Στο άρθρο [60] υποστηρίζουν ότι υπάρχει πάρα πολύ πληροφορία στον ορισμό της DAML-S που δυσκολεύει τον υπολογισμό των σηματολογικών ταιριασμάτων μεταξύ των περιγραφών DAML-S των υπηρεσιών. Η λύση που προτείνουν σε αυτό το πρόβλημα είναι να αφαιρεθεί η πληροφορία για το ποιος παρέχει ή ζητάει μια υπηρεσία από το προφίλ της υπηρεσίας και να το διατηρηθεί αυτή η πληροφορία ξεχωριστά. Στην εργασία [60] χρησιμοποιούν την μηχανή συλλογισμού περιγραφικής λογικής Racer [40] για τον υπολογισμό των σηματολογικών ταιριασμάτων μεταξύ των δημοσιεύσεων υπηρεσιών και των αιτήσεων για υπηρεσίες. Από την άποψη της περιγραφικής λογικής (description logic), οι δημοσιεύσεις υπηρεσιών και τα προφίλ είναι σχεδόν πανομοιότυπα και τα δύο υπάγονται στο προφίλ της υπηρεσίας. Για την διαδικασία ταιριάσματος το σύστημα Racer χρησιμοποιείται για τον υπολογισμό μιας ιεραρχίας των προφίλ υπηρεσιών όλων των δημοσιευμένων υπηρεσιών. Έπειτα, το σύστημα Racer κατηγοριοποιεί το προφίλ μιας υπηρεσίας για κάθε εισερχόμενη αίτηση και καθορίζει αν υπάρχει ένα ακριβές (exact) ταίριασμα ή ένα υποταίριασμα (subsume). Στο άρθρο [60], υλοποιήθηκε το σύστημα σαν ένα πολυπρακτορικό (multi-agent) σύστημα όπου οι πράκτορες παίζουν ένα σύνολο ρόλων περιλαμβανομένων των ρόλων του οικοδεσπότη, του διαφημιζόμενου και του αναζητητή. Η πλατφόρμα πρακτόρων JADE επιλέχθηκε με στόχο να απλοποιηθεί η ανάπτυξη και να εξασφαλιστεί η συμβατότητα με τα πρότυπα στην περιοχή των πρακτόρων.

Οι Benatallah κ.ά. [10] χρησιμοποιούν επίσης οντολογίες υπηρεσιών σε DAML-S για να περιγράψουν υπηρεσίες του παγκόσμιου ιστού και προτείνουν έναν αλγόριθμο ταιριάσματος ο οποίος βασίζεται στην θεωρία γράφων. Στην δική τους προσέγγιση, οι DAML-S οντολογίες ορίζονται χρησιμοποιώντας ένα

υποσύνολο της γλώσσας DAML+OIL για το οποίο υπάρχει ένας δομικός αλγόριθμος υπαγωγής (structural subsumption algorithm). Ο αλγόριθμος ταιριάσματος παίρνει ως είσοδο την αίτηση μιας υπηρεσίας και μια οντολογία υπηρεσιών εκφρασμένη σε DAML-S. Στο άρθρο [10] προτείνουν ότι την διαδικασία της ανακάλυψης υπηρεσιών πρέπει να την δούμε σαν μια διαδικασία επαναγραφής μιας αίτησης μιας υπηρεσίας εκφρασμένης στην γλώσσα DAML+OIL ως μια σύζευξη εννοιών DAML+OIL, που αναπαριστούν διαθέσιμες υπηρεσίες του παγκόσμιου ιστού. Αναγνωρίζουν ότι αυτό το πρόβλημα είναι παρόμοιο με ένα πρόβλημα βέλτιστης κάλυψης (best covering) και περιγράφουν πως αυτό μπορεί να εφαρμοστεί σε DAML-S οντολογίες. Οι βέλτιστες καλύψεις προφίλ (best profile covers) υπολογίζονται χρησιμοποιώντας υπεργράφους και συγκεκριμένα με την εύρεση των ελάχιστων μονοπατιών με το ελάχιστο κόστος. Σε σύγκριση με παρόμοιες δουλειές εκτός από τον έλεγχο της υπαγωγής προτείνουν την χρήση μιας πράξης διαφοράς (difference) μεταξύ της επερώτησης και της επαναγραφής της. Έτσι, προκύπτει ένα πιο ελαστικό ταίριασμα.

Οι Aversano κ.ά. [8] εκμεταλλεύονται την σύνθεση υπηρεσιών σαν μια τεχνική για την υποστήριξη της ανακάλυψης υπηρεσιών. Ο αλγόριθμος σηματολογικού ταιριάσματος που παρουσιάζεται συνθέτει πολλαπλές υπηρεσίες για να μπορέσει να ικανοποιήσει μια συγκεκριμένη αίτηση υπηρεσίας. Στο άρθρο [8] χρησιμοποιείται η οντολογία DAML-S για την περιγραφή μιας υπηρεσίας του παγκόσμιου ιστού και ο αλγόριθμος που παρουσιάζεται βασίζεται στην περιγραφή της υπηρεσίας σε DAML-S και συγκεκριμένα του προφίλ της υπηρεσίας. Η τρέχουσα υλοποίηση που παρουσιάζουν εφαρμόζει ταίριασμα μεταξύ των εισόδων και των εξόδων χωρίς να λαμβάνει υπόψη τις συνθήκες και τα αποτελέσματα. Η βασική ιδέα είναι ότι ο αλγόριθμος παίρνει ως εισόδους τον στόχο που θέλουμε να επιτευχθεί, που ουσιαστικά αναφέρεται στις απαιτούμενες εξόδους, και μελετάει την πιθανότητα να τον ικανοποιήσει με μία μόνο υπηρεσία. Αν αυτό δεν είναι δυνατόν, χρησιμοποιεί έναν αλγόριθμο με προς τα πίσω αλυσίδα εκτέλεσης (backward chaining) έτσι ώστε να ικανοποιήσει την αίτηση με την σύνθεση πολλαπλών υπηρεσιών. Ο αλγόριθμος για το ταίριασμα των εισόδων και των εξόδων, δεδομένου ότι είναι οντότητες μιας συγκεκριμένης οντολογίας, χρησιμοποιεί εκτιμητές ομοιότητας όπως την ομοιότητα λέξεων με χρήση του λεξικού Wordnet, ομοιότητα γνωρισμάτων, χρησιμοποιώντας ιδιότητες των τάξεων της γλώσσας DAML+OIL, και ομοιότητα γειτόνων χρησιμοποιώντας τις σχέσεις στην γλώσσα DAML+OIL. Τέλος, ο αλγόριθμος ανακάλυψης υπηρεσιών του παγκόσμιου ιστού που προτείνεται λαμβάνει υπόψη κάποια γνωρίσματα, όπως την ποιότητα, για την διαδικασία επιλογής του καλύτερου συνόλου υπηρεσιών του παγκόσμιου ιστού.

Ακόμη μια κεντρικοποιημένη προσέγγιση παρουσιάζεται στο άρθρο [13]. Οι συγγραφείς του αναπαριστούν μια υπηρεσία του παγκόσμιου ιστού χρησιμοποιώντας έναν φορμαλισμό διεργασιών τον οποίο ορίζουν για αυτόν τον σκοπό. Οι περιγραφές υπηρεσιών (π.χ. τα μοντέλα διεργασιών) κατηγοριοποιούνται χρησιμοποιώντας μια οντολογία διεργασιών από το MIT Process Handbook project. Στο άρθρο [13], θεωρούν ότι η χρήση μιας οντολογίας είναι χρήσιμη καθώς βοηθάει να

μειωθεί το βάρος της μοντελοποίησης μιας υπηρεσίας κατά την εύρεση της περισσότερο παρόμοιας διεργασίας στην οντολογία και στην μετατροπή της. Επιπλέον, η προσεκτική χρήση των οντολογιών διεργασιών κατά την αναζήτηση υπηρεσιών του παγκόσμιου ιστού μπορεί να αυξήσει την ακρίβεια (precision) χωρίς να μειώσει την ανάκληση (recall). Δεδομένου ότι οι υπηρεσίες έχουν αντιστοιχηθεί στην οντολογία, κάποιος μπορεί να αναζητήσει υπηρεσίες του παγκόσμιου ιστού απλά αναζητώντας την οντολογία. Για να αποφευχθούν τυχόν δυσκολίες κατά την αναζήτηση της οντολογίας, το άρθρο [13] ορίζει μια λογική γλώσσα επερωτήσεων που καλείται PQL (Process Query Language) για την ανάκτηση των μοντέλων διεργασιών που έχουν αντιστοιχηθεί στην οντολογία. Ο αλγόριθμος ταιριάσματος για την ανακάλυψη υπηρεσιών είναι ουσιαστικά αποτίμηση PQL επερωτήσεων πάνω από μία βάση με τις περιγραφές των υπηρεσιών.

Οι Keller κ.ά. [53] παρουσιάζουν ένα εννοιολογικό μοντέλο για την αυτόματη εύρεση υπηρεσιών αναγνωρίζοντας τα ακόλουθα διαφορετικά βήματα στην εύρεση υπηρεσιών: ανακάλυψη στόχου (goal discovery), εκλέπτυνση στόχου (goal refinement), ανακάλυψη υπηρεσίας (service discovery) και σύνδεση με την υπηρεσία (contracting). Ακολουθώντας την προσέγγιση του WSMO⁵ (Web service modelling ontology) [81] ο καταναλωτής της υπηρεσίας εκφράζει την αίτηση του σε όρους του τι θέλει να επιτευχθεί. Αυτό γίνεται με την χρήση φυσικής γλώσσας ή άλλων μέσων. Το στοιχείο που αναπαριστά μια τέτοια επιθυμία στο πλαίσιο WSMO καλείται *στόχος (goal)*. Με την ανακάλυψη στόχων βρίσκεται ένας προκαθορισμένο στόχος έτσι ώστε να αναπαρασταθεί με περισσότερη ακρίβεια η επιθυμία του καταναλωτή. Η ανακάλυψη υπηρεσιών θεωρείται ότι είναι η αναγνώριση πιθανών υποψήφιων υπηρεσιών χρησιμοποιώντας κάποιους απλούς αλγόριθμους ταιριάσματος ενώ η σύνδεση με υπηρεσίες βασίζεται σε ακριβείς περιγραφές των δυνατοτήτων και μπορεί να περιλαμβάνει αλληλεπίδραση μεταξύ του παροχέα και του καταναλωτή υπηρεσιών. Στο άρθρο [53] αναλύονται επιπλέον διάφορες έννοιες ταιριάσματος αυτού του πλαισίου.

Κατανεμημένες προσεγγίσεις

Σε μια επακόλουθη δουλειά του άρθρου [102], οι Paolucci κ.ά. [75] επεκτείνουν τις ιδέες του άρθρου [102] παρουσιάζοντας πως μπορεί να χρησιμοποιηθεί η οντολογία DAML-S για την εκτέλεση της ίδιας αναζήτησης βάσει των δυνατοτήτων σε ένα κατανεμημένο περιβάλλον ομότιμων κόμβων. Το πρωτόκολλο ομότιμων κόμβων που χρησιμοποιούν είναι το πρωτόκολλο Gnutella. Κάθε κόμβος χρησιμοποιεί τα μηνύματα του Gnutella για να διατηρήσει την διασύνδεση στο δίκτυο, να δημοσιεύσει περιγραφές υπηρεσιών και να ανακαλύψει παροχείς υπηρεσιών. Τοπικά, κάθε κόμβος έχει μια παρόμοια αρχιτεκτονική με την κεντρικοποιημένη μηχανή ταιριάσματος (DAML-S/UDDI Matchmaker) και επομένως και παρόμοιες τοπικές δυνατότητες.

⁵<http://www.wsmo.org/>

Οι Papazoglou κ.ά. [76] συζητάνε πως μπορεί να υλοποιηθεί ένα σύστημα δημοσίευσης/συνδρομής (publish-subscribe) για την ανακάλυψη υπηρεσιών του παγκόσμιου ιστού σε ένα περιβάλλον ομοσπονδιών (federated environment). Αντί ο κάθε κόμβος να δημοσιεύει τις δικές του περιγραφές υπηρεσιών είτε τοπικά είτε κεντρικά, οι κόμβοι που μοιράζονται ενδιαφέροντα μπορούν να σχηματίζουν ομοσπονδίες από μητρώα υπηρεσιών βασισμένα στο UDDI. Σε κάθε μια από αυτές τις ομοσπονδίες κόμβων ένας συγκεκριμένος κόμβος λειτουργεί σαν ένας υπερκόμβος παρέχοντας υπηρεσίες καταλόγου. Οι υπερκόμβοι συνδέονται μεταξύ τους με ένα μη δομημένο δίκτυο ομότιμων κόμβων. Το μοντέλο δεδομένων στην εργασία [76] βασίζεται στην γλώσσα WSDL και οι αλγόριθμοι των δικτύων ομότιμων κόμβων χρησιμοποιούν ιδέες από το γνωστό σύστημα δημοσίευσης/συνδρομής SIENA [21].

Μια άλλη κατακερματισμένη προσέγγιση για την ανακάλυψη υπηρεσιών του παγκόσμιου ιστού παρουσιάζεται στην δουλειά των Sivashanmugam κ.ά. [94]. Το επίκεντρο αυτής της δημοσίευσης είναι η δημιουργία ενός περιβάλλοντος ομοσπονδιών (federated registry). Τα σηματολογικά δεδομένα των μητρώων αυτής της υποδομής αποθηκεύονται σε μια οντολογία που καλείται Extended Registries Ontology (XTRO). Η οντολογία XTRO επιτρέπει την διατήρηση γνώσης για τα μητρώα και παρέχει απαντήσεις για επερωτήσεις που αναγνωρίζουν ένα μητρώο ή ένα σύνολο μητρώων έτσι ώστε να βρει μητρώα για την δημοσίευση και την ανακάλυψη υπηρεσιών του παγκόσμιου ιστού. Η υποδομή METEOR-S Web service discovery infrastructure (MWSDI) που παρουσιάζεται στο άρθρο [108] χρησιμοποιείται για την παροχή ενός μηχανισμού για την δημοσίευση και την ανακάλυψη σε αυτό το περιβάλλον των πολλαπλών μητρώων. Η υποδομή MWSDI υλοποιείται σαν ένα δίκτυο ομότιμων κόμβων, χρησιμοποιώντας το πλαίσιο JXTA. Δύο βασικοί τύποι κόμβων είναι ο κόμβος χειριστής (operator) και ο κόμβος πελάτης (client). Ο κόμβος χειριστής λειτουργεί το UDDI μητρώο και παρέχει υπηρεσίες για αυτό ενώ ο κόμβος πελάτης επιτρέπει στους χρήστες να χρησιμοποιήσουν τις δυνατότητες του MWSDI. Το MWSDI χρησιμοποιεί τα t-Models, μία δομή του UDDI, για να προσθέσει σηματολογικές επισημειώσεις (annotations) στις UDDI οντότητες. Επομένως, στο άρθρο [94] υποστηρίζονται μόνο UDDI μητρώα. Κάθε ομοσπονδία θα έχει έναν κατάλογο με tModel, που είναι ένα ειδικού σκοπού UDDI μητρώο, που περιέχει μόνο δεδομένα σχετικά με το συγκεκριμένο tModel. Κατά την δημοσίευση υπηρεσιών σε μια ομοσπονδία, ο χρήστης του MWSDI πρέπει πρώτα να καθορίσει τα κριτήρια για την επιλογή μητρώου και μετά να δημοσιεύσει την υπηρεσία. Κατά την ανακάλυψη υπηρεσιών εάν ένα μητρώο που είναι μέρος μιας ομοσπονδίας λάβει την αίτηση για την ανακάλυψη μιας υπηρεσίας τότε μεταδίδει την αντίστοιχη επερώτηση εντός της ομοσπονδίας. Ο κόμβος χειριστής συνεχίζει την διαδικασία της ανακάλυψης υπηρεσιών εξυπηρετώντας τον κόμβο πελάτη. Για να εξασφαλίσουμε ομοιόμορφη σηματολογική μετάφραση της επερώτησης σε όλα τα μητρώα, χρησιμοποιείται ο κατάλογος tModel για την μετάφραση της επερώτησης.

Οι Schmidt και Parashar [87] χαρακτηρίζουν τις υπηρεσίες του παγκοσμίου

ιστού με ένα σύνολο λέξεων-κλειδιά. Χρησιμοποιούν αυτές τις λέξεις-κλειδιά για να ευρετηριάσουν τις περιγραφές των υπηρεσιών σε κόμβους του υποκείμενου δικτύου που δημιουργείται ως ο ΚΠΚ Chord [97]. Στο άρθρο [87] προσπαθούν να διατηρήσουν την τοπικότητα μέσα σε αυτόν τον χώρο των λέξεων-κλειδιά όχι με την απόλυτη έννοια αλλά χρησιμοποιώντας μια αντιστοίχιση διατήρησης τοπικότητας (*locality-preserving mapping*) που καλείται *space filling curves* (SFC). Έτσι, αντιστοιχούνται τα στοιχεία δεδομένων που περιγράφουν τις υπηρεσίες του παγκόσμιου ιστού σε κόμβους. Υποστηρίζονται πολύπλοκες επερωτήσεις που αποτελούνται από λέξεις-κλειδιά, μερικές λέξεις-κλειδιά και *wildcards*. Η μηχανή επερωτήσεων αποτιμάει αυτές τις επερωτήσεις μεταφράζοντας την κάθε μία σε σχετικές συστάδες του χώρου που είναι βασισμένος σε SFC. Έπειτα, θέτει την επερώτηση στους αντίστοιχους κόμβους. Αυτή η προσέγγιση είναι σημαντική γιατί ασχολείται με την ανομοιομορφία των κατανομών των δεδομένων (π.χ. κάποιες υπηρεσίες που αναπαριστώνται με συγκεκριμένες λέξεις-κλειδιά μπορεί να είναι πιο δημοφιλείς από άλλες) με έναν κομψό τρόπο.

Άλλη μια δουλειά σχετική με την ανακάλυψη υπηρεσιών του παγκόσμιου ιστού χρησιμοποιώντας δίκτυα ομότιμων κόμβων παρουσιάζεται από τους Emekci κ.ά. [35]. Το άρθρο [35] παρουσιάζει μια μέθοδο ανακάλυψης υπηρεσιών του παγκόσμιου ιστού που λαμβάνει υπόψη της όχι μόνο την λειτουργία της υπηρεσίας αλλά και την συμπεριφορά της. Σε αυτή την δουλειά έχει επίσης επιλεχθεί ο ΚΠΚ Chord σαν το υποκείμενο δίκτυο. Μια υπηρεσία αναπαριστάται σαν ένα πεπερασμένο αυτόματο (*finite automaton*) της περιγραφής της στην οντολογία BPEL/DAML-S και πεπερασμένα μονοπάτια αυτών των αυτόματων ευρετηριάζονται στον δακτύλιο του Chord. Επομένως, μπορούν να γίνουν επερωτήσεις στις περιγραφές των υπηρεσιών του παγκόσμιου ιστού χρησιμοποιώντας το πεπερασμένο αυτόματο της ζητούμενης υπηρεσίας. Εκτός από την ανακάλυψη υπηρεσιών, το άρθρο [35] χρησιμοποιεί μια μέθοδο για την αποτίμηση της ποιότητας των υπηρεσιών και ταξινομεί τα επιστρεφόμενα αποτελέσματα. Επιπλέον, χρησιμοποιείται ένα μοντέλο φήμης βασισμένο στην θεωρία σχεδίων *sketch theory*, όπου οι υπηρεσίες ψηφίζουν η μία την άλλη και παράγουν βαθμολογικές κατατάξεις για εμπιστοσύνη και ποιότητα. Ο προτεινόμενος σχεδιασμός απαιτεί ένα καθολικό λεξικό για την ονομασία των μεταβάσεων. Όλες οι υπηρεσίες χρησιμοποιούν μια DAML-S οντολογία για να εξασφαλιστεί ότι το σύστημα θα δουλέψει αποδοτικά.

Οι Schlosser κ.ά. [85] παρουσιάζουν μια κατανεμημένη αρχιτεκτονική ομότιμων κόμβων για την ανακάλυψη υπηρεσιών του παγκόσμιου ιστού βασισμένη στον ΚΠΚ HyperCup. Η βασική ιδέα είναι η χρήση καθολικών οντολογιών για τον διαμοιρασμό της τοπολογίας του δικτύου σε συστάδες κλάσεων και η εκμετάλλευση αυτής της τοπολογίας για να καταστεί δυνατή η δρομολόγηση σε συγκεκριμένες συστάδες κλάσεων. Στο άρθρο [85], χρησιμοποιούν οντολογίες και ταξινομούν τις υπηρεσίες του παγκόσμιου ιστού σαν στιγμιότυπα των κλάσεων οντολογιών. Για παράδειγμα, μια υπηρεσία πώλησης αυτοκινήτων μπορεί να περιγραφεί χρησιμοποιώντας κλάσεις από την οντολογία υπηρεσιών (οντολογία πώλη-

σεων) και από την οντολογία αυτοκινήτων (μάρκες και τύποι αυτοκινήτων). Κατά την ανακάλυψη υπηρεσιών του παγκόσμιου ιστού το δίκτυο HyperCup χρησιμοποιείται για να μεταδώσει μηνύματα στο δίκτυο με ένα αποδοτικό τρόπο εκμεταλλευόμενο την δομή του υπερκύβου. Στο άρθρο [85] υποστηρίζουν επερωτήσεις που αποτελούνται από λογικούς συνδυασμούς κλάσεων οντολογιών. Κάθε επερωτηση μεταδίδεται σε εκείνες τις συστάδες κλάσεων που περιέχουν κόμβους στους οποίους στοχεύει η επερωτηση. Έπειτα, μια μετάδοση γίνεται εντός καθεμιάς από τις συστάδες.

Οι Sapkota κ.ά. [83] χρησιμοποιούν ένα ιεραρχικό δίκτυο ομότιμων κόμβων για να διασυνδέσουν διαφορετικές πλατφόρμες σηματολογικών υπηρεσιών του παγκόσμιου ιστού (Semantic Web services platforms π.χ. οι πλατφόρμες WSMX, IRS-II). Χρησιμοποιούν την προσέγγιση της WSMO για την περιγραφή τόσο των αιτήσεων των χρηστών όσο και των υπηρεσιών του παγκόσμιου ιστού. Αυτές οι πλατφόρμες οργανώνονται σε συστάδες (clusters) βάσει της ομοιότητας των περιγραφών των υπηρεσιών του παγκόσμιου ιστού. Κάθε μια από αυτές τις συστάδες έχει τουλάχιστον ένα υπερ-κόμβο που ονομάζεται *διαχειριστής συστάδας* (ΔΣ) (cluster manager (CM)). Ο διαχειριστής επιλέγεται δυναμικά βάσει κάποιων μετρικών όπως την διαθεσιμότητα, την επεξεργαστική δύναμη και την δυνατότητα αποθήκευσης. Επιπλέον, ο διαχειριστής είναι υπεύθυνος για την ευρετηρίαση των υπηρεσιών του παγκόσμιου ιστού που έχουν εγγραφεί στην συστάδα του και για την επικοινωνία με τους υπόλοιπους διαχειριστές των άλλων συστάδων. Μια συστάδα που αποτελείται μόνο από διαχειριστές καλείται *υπερσυστάδα* (super cluster) και είναι ουσιαστικά ένα ιεραρχικό δίκτυο. Ο στόχος ενός χρήστη ικανοποιείται πάντα ως προς μια υπερσυστάδα. Για τους σκοπούς της ανακάλυψης υπηρεσιών του παγκόσμιου ιστού και του ταιριάσματος το άρθρο [83] χρησιμοποιεί το πλαίσιο που παρουσιάστηκε στο άρθρο [54] χρησιμοποιώντας ένα αλγόριθμο αποσύνθεσης στόχων. Η δουλειά του [83] εστιάζει στις υπηρεσίες που παρέχονται από περιβάλλοντα δικτύων ομότιμων κόμβων στις πλατφόρμες σηματολογικών υπηρεσιών υποστηρίζοντας την επικοινωνία μεταξύ τους.

Οι πίνακες 3.2 και 3.3 αποτελούν μια περίληψη της ανασκόπησης αυτής της ενότητας και παρουσιάζουν τις βασικότερες ιδιότητες των μεθόδων για ανακάλυψη υπηρεσιών του παγκόσμιου ιστού και ταιριάσματος που έχουν παρουσιαστεί στην σχετική βιβλιογραφία.

3.3 Ανακάλυψη πόρων και υπηρεσιών σε Πλέγματα

Αυτή η ενότητα συνεχίζει την ανασκόπηση της προηγούμενης, αλλά επικεντρώνεται περισσότερο στην ανακάλυψη πόρων και υπηρεσιών στο περιβάλλον του Πλέγματος (Grid). Ξεκινάμε την παρουσίαση με μια συζήτηση σχετική με την ανακάλυψη πόρων και υπηρεσιών σε καθιερωμένα συστήματα Πλέγματος και συνεχίζουμε

Άρθρο	Αρχιτεκτονική	Μοντέλο δημοσίευσης	Μοντέλο επερωτήσεων	Μέθοδος ταιριάσματος
<i>UDDI</i> [107]	Κεντριοποιημένη	service type definition (tModel)	Αναζήτηση με λέξεις-κλειδιά	Δεν έχει οριστεί στο αρχικό πλαίσιο
<i>Wang and Stroulia</i> [109, 110]	Κεντριοποιημένη	περιγραφή WSDL	Μερική περιγραφή WSDL + σύνολο διαθέσιμων περιγραφών WSDL	αλγόριθμος ταιριάσματος WSDL δομών
<i>Degwekar et al.</i> [31]	Κεντριοποιημένη (UDDI + αποθήκη περιορισμών)	WSDL επεκταμένη με πληροφορία περιορισμών	RSL βασισμένη σε περιορισμούς	CSP
<i>Bilgin and Singh</i> [14]	Κεντριοποιημένη (επέκταση στο UDDI)	WSDL + DAML	SWSQL (γνωρίσματα ποιότητας)	αποτίμηση επερωτήσεων SQL
<i>Sycara et al.</i> [102]	Κεντριοποιημένη	DAML-S (service profile, process model, grounding)	DAML-S (service profile, process model, grounding)	DAML-S/UDDI Matchmaker
<i>Li and Horrocks</i> [60]	Κεντριοποιημένη (πολυπρακτορικό σύστημα)	DAML-S	DAML-S	Racer (DL reasoner)
<i>Benatallah et al.</i> [10]	Κεντριοποιημένη	DAML-S	DAML-S (εισόδοι, εξόδοι)	Χρήση υπεργράφων
<i>Aversano et al.</i> [8]	Κεντριοποιημένη	DAML-S	DAML-S	Σύνθεση + Αλγόριθμος ταιριάσματος οντοτήτων
<i>Bernstein and Klein</i> [13]	Κεντριοποιημένη	Μοντέλο διεργασιών	Process Query Language (PQL)	αποτίμηση επερωτήσεων PQL
<i>Keller et al.</i> [53]	Κεντριοποιημένη	σύνολο προκαθορισμένων στόχων	Σύνολο αλλαγών συνόλων	βασισμένο στην θεωρία των συνόλων

Πίνακας 3.2: Κεντριοποιημένες τεχνικές για ανακάλυψη υπηρεσιών του παγκόσμιου ιστού

Άρθρο	Αρχιτεκτονική	Μοντέλο δημοσίευσης	Μοντέλο επερωτήσεων	Μέθοδος ταιριάσματος
<i>Paolucci et al.</i> [74]	Καταμεμημένη (Gnutella)	DAML-S (service profile, process model, grounding)	DAML-S (service profile, process model, grounding)	DAML-S/UDDI Matchmaker
<i>Papazoglou et al.</i> [76]	Καταμεμημένη (ιεραρχικό)	set of WSDL port-types	WSDL port-type και σύνολο λειτουργιών	Χρήση εννοιολογικών πλεγμάτων
<i>Sivashan mugam et al.</i> [94]	Καταμεμημένη (hybrid P2P)	WSDL	WSDL	MSWDI (SAWS αλγόριθμος)
<i>Schmidt and Parashar</i> [87]	Καταμεμημένη (Chord)	Περιγραφή (π.χ. WSDL) χαρακτηρισμένη με ένα σύνολο λέξεων κλειδιά	Σύζευξη λέξεων κλειδιά (υποστηρίζει χρήση wildcards)	Χώρος βασισμένος σε SFC
<i>Emekci et al.</i> [35]	Καταμεμημένη (Chord)	περιγραφή BPEL/DAML-S και αυτόματο της υπηρεσίας	περιγραφή BPEL/DAML-S σαν πεπερασμένο αυτόματο	Σύγκριση αυτόματων
<i>Schlosser et al.</i> [85]	Καταμεμημένη (HyperCuP)	Οντολογίες υπηρεσιών μαζί με οντολογίες πεδίων	Λογικός συνδυασμός εννοιών οντολογιών	Αντιστοίχιση μεταξύ οντολογιών
<i>Sapkota et al.</i> [83]	Καταμεμημένη (ιεραρχικό)	WSMO	WSMO	Δεν αναφέρεται λεπτομερώς

Πίνακας 3.3: Καταμεμημένες τεχνικές για ανακάλυψη υπηρεσιών του παγκόσμιου ιστού

με πιο πρόσφατες προσεγγίσεις που εκμεταλλεύονται την σημασιολογία για την περιγραφή των πόρων και των υπηρεσιών για την δημιουργία αποδοτικών αλγορίθμων ταιριάσματος.

3.3.1 Ανακάλυψη στα συστήματα Condor και Globus

Το άρθρο [77] περιγράφει ένα πλαίσιο για το ταίριασμα και την διαχείριση κατανεμημένων πόρων στο σύστημα Condor. Το Condor είναι ένα γνωστό περιβάλλον πλέγματος με σκοπό υπολογισμούς υψηλών απαιτήσεων (high throughput computing (HTC)) σε ένα μεγάλο πλήθος ετερογενών πόρων. Το πλαίσιο της εργασίας [77] χρησιμοποιεί ένα ημιδομημένο μοντέλο δεδομένων βασισμένο σε ζεύγη *γνωρισματιμής* για την περιγραφή των πόρων και των ιδιοτήτων τους. Με αυτόν τον τρόπο δεν χρειάζεται κάποιο προκαθορισμένο σχήμα και οι ετερογενείς πόροι μπορούν να μοντελοποιηθούν με φυσικό τρόπο. Αυτό το μοντέλο δεδομένων ονομάζεται μοντέλο των ταξινομημένων διαφημίσεων (*classified advertisements (classads)*). Οι οντότητες που παρέχουν ή απαιτούν μια υπηρεσία διαφημίζουν τα χαρακτηριστικά ή τις απαιτήσεις τους αντίστοιχα σε αυτές τις διαφημίσεις (*classads*). Το πλαίσιο ταιριάσματος του άρθρου [77] μπορεί να γίνει κατανοητό μέσω των ακόλουθων πέντε στοιχείων: τον *ορισμό του classad* που ορίζει την γλώσσα που χρησιμοποιείται για την έκφραση των χαρακτηριστικών και των απαιτήσεων, το *πρωτόκολλο δημοσίευσης* που περιγράφει τον τρόπο που ο υπεύθυνος για το ταίριασμα βρίσκει τις δημοσιεύσεις, τον *αλγόριθμο ταιριάσματος* που ορίζει την διαδικασία ταιριάσματος, το *πρωτόκολλο ταιριάσματος* που περιγράφει τον τρόπο που οι οντότητες ειδοποιούνται σε περίπτωση ταιριάσματος και το *πρωτόκολλο απαιτήσεων* που ορίζει τον τρόπο που ταιριασμένες οντότητες επικοινωνούν και διαπραγματεύονται μεταξύ τους. Δεδομένου ότι η διαδικασία ταιριάσματος γίνεται από τον διαχειριστή ενός Condor pool, η προσέγγιση θεωρείται κεντρικοποιημένη.

Στο άρθρο [78], η ομάδα του Condor προχωράει από το δυαδικό ταίριασμα των πόρων που παρουσίασε στο άρθρο [77], στο ταίριασμα που προκύπτει από το πρόβλημα όταν έχουμε μια εργασία που απαιτεί έναν σταθμό εργασίας και ένα δίπλωμα λογισμικού για να εκτελεστεί με επιτυχία. Επομένως, στο άρθρο [78], οι συγγραφείς μελετάνε το πρόβλημα του ταιριάσματος τριών συμμετεχόντων (π.χ. εργασίας, σταθμού εργασίας και διπλώματος) επεκτείνοντας τις ιδέες του άρθρου [77].

Το άρθρο [28] παρουσιάζει μια αρχιτεκτονική για υπηρεσίες πληροφοριών Πλέγματος που ασχολείται με τις απαιτήσεις των περιβαλλόντων Πλέγματος. Αυτή η αρχιτεκτονική έχει υλοποιηθεί ως η υπηρεσία παρακολούθησης και ανακάλυψης (Monitoring and Discovery Service (MDS-2)) της δεύτερης έκδοσης του Globus toolkit (GT2). Η υπηρεσία MDS-2 αποτελείται από τα ακόλουθα στοιχεία: τους *παροχείς πληροφοριών*, που παρέχουν πληροφορίες για συγκεκριμένες οντότητες, τους *καταναλωτές πληροφοριών* που θέτουν ερωτήσεις σε αυτές τις πληροφορίες και την *υπηρεσία αδροιστικής αναζήτησης* που διευκολύνει την ανακάλυψη

η και παρακολούθηση των πόρων. Η τελευταία είναι μια ιεραρχική υπηρεσία ανακάλυψης. Κάθε κατάλογος χρησιμοποιεί το μοντέλο δεδομένων του πρωτοκόλλου GRIP (Grid Information Protocol), την γλώσσα επερωτήσεων και το ίδιο το πρωτόκολλο και δρα σαν έναν παροχέας πληροφοριών που διαθέτει πληροφορίες για όλους τους πόρους που βρίσκονται χαμηλότερα στην ιεραρχία. Το πρωτόκολλο GRIP υποστηρίζει την αναζήτηση καθώς και λειτουργίες συνδρομής και βασίζεται στο LDAP [47].

Ενώ στην περίπτωση της ανακάλυψης πόρων του Πλέγματος, ενδιαφερόμαστε περισσότερο για τα χαρακτηριστικά του πόρου *σε κάποια συγκεκριμένη στιγμή*, κατά την παρακολούθηση των πόρων, ενδιαφερόμαστε περισσότερο για το πώς αυτά τα χαρακτηριστικά αλλάζουν *κατά την διάρκεια του χρόνου*. Επομένως, έστω κι αν η ανακάλυψη και η παρακολούθηση περιλαμβάνουν σε πολλές περιπτώσεις τους ίδιους τύπους πληροφορίας, προτιμάται να ζητείται πληροφορία, στην πρώτη περίπτωση ξεχωριστά, ενώ στην δεύτερη περίπτωση προτιμάται η πληροφορία να στέλνεται ασύγχρονα π.χ. όταν κάποιες τιμές αλλάζουν. Το πρωτόκολλο GRIP μπορεί να χρησιμοποιηθεί για να παρέχει και τις δύο λειτουργικότητες δεδομένου ότι υποστηρίζει και τον τρόπο *pull* για τους σκοπούς της ανακάλυψης και τον τρόπο *push* κατά την παρακολούθηση.

Η επόμενη γενιά της υπηρεσίας MDS-2, η MDS-3 [26] βασίζεται στο πρότυπο OGSi (Open Grid Services Infrastructure) για υπηρεσίες του παγκόσμιου ιστού. Χρησιμοποιεί την γλώσσα XML ως το μοντέλο δεδομένων και την γλώσσα Xpath σαν γλώσσα επερωτήσεων. Η υπηρεσία MDS-3 περιλαμβάνεται στην τρίτη έκδοση του Globus Toolkit(GT3). Τέλος, στην τελευταία έκδοση της υπηρεσίας MDS, την MDS-4 [25], ακολουθούνται τα πρότυπα WSRF και χρησιμοποιούνται οι ιδιότητες WS-Resource, οι οποίες είναι βασισμένες σε XML. Η υπηρεσία MDS-4 περιλαμβάνεται στο στην τέταρτη έκδοση του Globus Toolkit(GT4).

Στο άρθρο [17], γίνεται ένα πρώτο σημαντικό βήμα για να επιτευχθεί η *διαλειτουργικότητα* στο ταίριασμα πόρων του Πλέγματος. Μέσω της διαλειτουργικότητας, ένας μεσίτης πόρων (resource broker) μπορεί να ανακαλύψει πόρους σε διαφορετικά συστήματα Πλέγματος. Στο άρθρο [17], αναφέρονται στην εργασία που χρηματοδοτήθηκε από το πρόγραμμα Grid Interoperability (GRIP)⁶ με στόχο την δημιουργία ενός μεσίτη πόρων που θα μπορούσε να ανακαλύψει διαθέσιμους πόρους μιας ομοσπονδίας ενός συστήματος Unicore [36] ή ενός συστήματος Globus. Για το σύστημα Globus χρησιμοποιήθηκε σαν σχήμα περιγραφής πόρων το σχήμα GLUE. Το άρθρο [17] προτείνει να χρησιμοποιηθεί η ιεραρχία των τάξεων της Java στον ορισμό των αντικειμένων (Unicore Abstract Job Objects) μαζί με τον ορισμό του GLUE για να οριστούν οντολογίες που ορίζουν τυπικά αυτά τα μοντέλα και τις αντιστοιχίες ανάμεσα σε παρόμοιες έννοιες στις δύο οντολογίες. Αυτές οι αντιστοιχίες χρησιμοποιούνται στο άρθρο [17] από έναν μεταφραστή που αντιστοιχεί τις αιτήσεις πόρων του Unicore σε κατάλληλες επερωτήσεις X-Path που θα χρησιμοποιηθούν από την υπηρεσία MDS3. Με αυτόν τον τρόπο,

⁶<http://www.grid-interoperability.org/>

μπορεί να επιτευχθεί διαλειτουργικότητα μεταξύ των δύο αυτών συστημάτων του Πλέγματος.

3.3.2 Σηματολογικές τεχνικές

Οι Heine κ.ά. [45] παρουσιάζουν μια καταμεμημένη τεχνική για σηματολογική ανακάλυψη πόρων του Πλέγματος χρησιμοποιώντας την τεχνολογία των δικτύων ομότιμων κόμβων και συγκεκριμένα τον ΚΠΚ Chord. Εκτός από την κατανομή των περιγραφών των πόρων, στο άρθρο [45] κατανέμουν και τις οντολογίες που χρησιμοποιούνται για την έκφραση της σηματολογίας των πόρων. Με αυτόν τον τρόπο, η γνώση που αναφέρεται σε ένα συγκεκριμένο πεδίο μπορεί εύκολα να αποκτηθεί και αυτή η τεχνική δεν απαιτεί ότι κάθε κόμβος του δικτύου έχει γνώση κάποιας καθολικής οντολογίας. Κάθε κόμβος στο δίκτυο διαθέτει ένα τοπικό σύστημα Περιγραφικής Λογικής (Description Logic system) και εκτελεί τις προσθήκες στο A-Box και διαθέτει το δικό του T-Box. Το A-Box περιέχει πληροφορία για στιγμιότυπα εννοιών ενώ το T-Box για την ιεραρχία των εννοιών. Κάθε έννοια αποτελεί μια υπό-έννοια της κύριας έννοιας του πόρου, και κατακερματίζεται για να δημιουργήσει το αναγνωριστικό που οδηγεί στον κόμβο όπου θα αποθηκευτεί η γνώση A-Box και T-Box. Στο άρθρο [45], ασχολούνται μόνο με την ταξινόμηση εννοιών, χωρίς να ασχολούνται με θέματα που αφορούν τους ρόλους. Κάθε κόμβος έχει το τοπικό DAG ταξινόμησης, το οποίο υπολογίζεται από το τοπικό του σύστημα Περιγραφικής Λογικής. Το σύνολο των DAG όλων των κόμβων σχηματίζουν ένα ιδεατό DAG στο δίκτυο. Οι απλές επερωτήσεις, που περιέχουν μια έννοια, δρομολογούνται στον υπεύθυνο κόμβο για αυτήν την έννοια. Έπειτα, αυτός ο κόμβος επιστρέφει όλους τους κόμβους που αποθηκεύουν στιγμιότυπα αυτής της έννοιας. Στο άρθρο [45], υποστηρίζονται επερωτήσεις που περιέχουν σύζευξη, διάζευξη και άρνηση απλών εννοιών. Η λίστα των στιγμιότυπων δημιουργείται από κάτω προς τα πάνω, εφαρμόζοντας τομή ή ένωση αντίστοιχα, για τις δύο πρώτες περιπτώσεις. Στην άρνηση, είτε η λίστα των στιγμιότυπων αφαιρείται από την λίστα των στιγμιότυπων της κύριας έννοιας ή στην περίπτωση άρνησης μαζί με σύζευξη, οι δύο λίστες στιγμιότυπων αφαιρούνται. Όπως είναι αναμενόμενο, στο άρθρο [45] θεωρούν ότι έστω κι εάν οι αρνήσεις που δεν εμφανίζονται σε συνδυασμό με συζεύξεις δεν είναι αποδοτικές, δεν είναι πιθανό αυτές να εμφανιστούν σε πραγματικές επερωτήσεις. Τέλος, στο άρθρο [45] θεωρείται ότι οι κόμβοι δεν αφήνουν το δίκτυο τυχαία αλλά ειδοποιούν το δίκτυο πριν αποχωρήσουν στέλνοντας κάποια μηνύματα και αφαιρώντας *μόνο* την γνώση A-Box (επομένως η πληροφορία ταξινόμησης διατηρείται στο σύστημα).

Στο άρθρο [23], οι Chen κ.ά. προτείνουν ένα γενικό πλαίσιο για μια συστηματική και ολοκληρωμένη τεχνική για την διαχείριση μεταδεδομένων πόρων του Πλέγματος, όπως τις υπηρεσίες του παγκόσμιου ιστού, τους υπολογιστικούς αλγόριθμους και συναρτήσεις. Αυτή η δουλειά εστιάζει στην μοντελοποίηση δεδομένων και μεταδεδομένων και στην σηματολογική επισήμειωση χρησιμοποιώντας οντολογίες πόρων και οντολογίες πεδίων, προχωρημένους μηχανισμούς συλλο-

γισμού καθώς και την εκμετάλλευση αυτού του πλαισίου σε μια πραγματική εφαρμογή Πλέγματος. Εκτός από τις *οντολογίες πεδίων*, που αποτελούν μέρος αυτού του πλαισίου, ένα άλλο βασικό κομμάτι είναι η *αποθήκη πόρων* (resource registry) που είναι υπεύθυνη για την απόδοση σημασιολογίας σε περιγραφές πόρων. Η αποθήκη προσθέτει μεταδεδομένα στις περιγραφές με την δημοσίευση αυτών σε μια *κεντρική αποθήκη γνώσης πόρων* (resource knowledge repository). Εκεί διατηρείται όλη η σημασιολογική πληροφορία των πόρων και η αποθήκη αυτή έχει αντίστοιχη λειτουργικότητα με ένα μητρώο UDDI. Αυτή η αποθήκη αποτελείται από δύο στοιχεία: το API, που παρέχει μια διεπαφή αναπαράστασης OWL για την περιγραφή των πόρων και την βάση γνώσης που χρησιμοποιεί έναν χώρο αποθήκευσης στιγμιοτύπων⁷ (συνδυασμός μιας μηχανής συλλογισμού Περιγραφικής Λογικής και μια σχεσιακής βάσης). Τέλος, το πλαίσιο του άρθρου [23] περιλαμβάνει μια υπηρεσία *αναζήτησης πόρων* που βοηθάει τον χρήστη να κάνει επερωτήσεις χρησιμοποιώντας μια γραφική διεπαφή βασισμένη σε οντολογίες. Μια εφαρμογή Πλέγματος στα πλαίσια του προγράμματος GEODISE⁸ επίσης μελετάται στο άρθρο [23]. Το προτεινόμενο πλαίσιο χρησιμοποιείται για να προσθέσει σημασιολογία σε συναρτήσεις του MatLab, που αποτελούν σε αυτήν την περίπτωση τους πόρους του Πλέγματος. Για αυτόν τον λόγο, μια οντολογία συναρτήσεων του GEODISE εκφρασμένη σε OWL χρησιμοποιείται για την επισημείωση περιγραφών συναρτήσεων. Αφού ολοκληρωθεί η διαδικασία επισημείωσης, οι επισημειώσεις δημοσιεύονται στην αποθήκη γνώσης πόρων. Η λειτουργικότητα επερωτήσεων υποστηρίζεται μέσω μιας γραφικής διεπαφής που επίσης βασίζεται σε οντολογίες.

Οι Wroe κ.ά. [112] μελετάνε την χρήση οντολογιών για την περιγραφή υπηρεσιών του παγκόσμιου ιστού σε ένα περιβάλλον Πλέγματος με στόχο την διευκόλυνση της διαδικασίας ταιριάσματος και ανακάλυψης. Αυτή η εργασία έγινε στα πλαίσια του προγράμματος *myGrid*⁹, το οποίο αναπτύσσει λογισμικό Πλέγματος στην περιοχή της Βιο-Πληροφορικής. Ο στόχος του προγράμματος *myGrid* είναι η αυτοματοποίηση της διαδικασίας δημιουργίας workflows συνδυάζοντας υπηρεσίες, μια διαδικασία που οι επιστήμονες της Βιο-Πληροφορικής συνήθιζαν να εκτελούν χειρωνακτικά.

Οι συγγραφείς υποστηρίζουν ότι η γλώσσα DAML+OIL [27] αποτελεί μια γλώσσα κατάλληλη για την έκφραση οντολογιών, δεδομένου ότι μπορεί να παρέχει ταξινόμηση υπηρεσιών, περιγραφή υπηρεσιών και δυνατότητες συλλογισμού. Χρησιμοποιώντας ως βάση την οντολογία DAML-S [6], στην εργασία [112] δημιουργείται μια οντολογία για την περιγραφή υπηρεσιών που είναι ειδικευμένη για το πεδίο της Βιο-Πληροφορικής. Κάποιες ιδιότητες έχουν προστεθεί για να ικανοποιήσουν τις ανάγκες των επιστημόνων αυτής της περιοχής, ενώ κάποιες δευτερεύουσες δεν χρησιμοποιούνται. Επιπλέον, χρησιμοποιούνται και οι οντολογίες πεδίων για

⁷<http://instancestore.man.ac.uk/>

⁸www.geodise.org/

⁹www.mygrid.org.uk

την περιγραφή των υπηρεσιών. Η αρχιτεκτονική του *myGrid* περιλαμβάνει έναν εξυπηρετητή οντολογιών (ontology server), ο οποίος παρέχει πρόσβαση στις οντολογίες DAML+OIL, την μηχανή συλλογισμού Fact, που είναι υπεύθυνη για την εκτέλεση των συλλογισμών, και έναν κατάλογο υπηρεσιών, που χρησιμοποιείται για την αποθήκευση περιγραφών υπηρεσιών (αποτελεί μια επέκταση του UDDI) και έναν αξιολογητή για την διαχείριση των υποψηφίων περιγραφών των υπηρεσιών.

Οι Lord κ.ά. [62] παρουσιάζουν μια αρχιτεκτονική για την σημασιολογική ανακάλυψη υπηρεσιών, ειδικά σχεδιασμένη για το πεδίο της Βιο-Πληροφορικής. Η μηχανή της σημασιολογικής ανακάλυψης των υπηρεσιών του άρθρου [62], ονομάζεται Feta, και έχει επίσης δημιουργηθεί στα πλαίσια του προγράμματος *myGrid*. Η προσέγγιση του άρθρου [62] θεωρείται απλούστερη και πιο αποδοτική σε σχέση με αυτήν που είχε προταθεί στο προηγούμενο άρθρο [112]. Η επισημείωση μιας υπηρεσίας γίνεται χειρωνακτικά χρησιμοποιώντας το εργαλείο Pedro που αποτελεί ένα εργαλείο εισαγωγής δεδομένων XML. Εκτός από τις περιγραφές των υπηρεσιών, η μηχανή Feta διαθέτει γνώση για τις οντολογίες πεδίων που είναι συσχετισμένες με αυτές τις περιγραφές. Αυτές οι οντολογίες είναι εκφρασμένες σε OWL, έτσι ώστε η διαδικασία συλλογισμού σε αυτές να είναι δυνατή κατά την ανάπτυξη και μετά να γίνεται εξαγωγή τους σε RDFS και η αποθήκευσή τους στην μηχανή Feta. Η μηχανή Feta παρέχει ένα σύνολο συγκεκριμένων RDDL επερωτήσεων για την ανακάλυψη υπηρεσιών μέσω μιας διεπαφής υπηρεσιών του παγκόσμιου ιστού και με την χρήση του εργαλείου Jena για την απάντησή τους. Στο άρθρο [62], παρέχεται επιπλέον ένα εργαλείο επερωτήσεων με γραφική διεπαφή για την υποβολή των αντίστοιχων επερωτήσεων.

Στο άρθρο [104], ασχολούνται με το πρόβλημα του ταιριάσματος πόρων στο Πλέγμα χρησιμοποιώντας οντολογίες. Η μηχανή ταιριάσματος που βασίζεται σε οντολογίες αποτελείται από τα εξής τρία βασικά στοιχεία: *οντολογίες, γνώση πεδίων και κανόνες ταιριάσματος*. Οι οντολογίες που χρησιμοποιήθηκαν είναι βασικά τρεις οντολογίες πεδίων: μια *οντολογία πόρων*, μια *οντολογία αιτήσεων πόρων* και μια *οντολογία πολιτικών* που είναι εκφρασμένες σε RDFS. Έχοντας διαφορετικές οντολογίες για τις περιγραφές πόρων και για τις αιτήσεις για πόρους και επομένως, χρησιμοποιώντας ασύμμετρες περιγραφές ενός πόρου και μιας αίτησης, δεν υπάρχει λόγος για τον συντονισμό μεταξύ των παροχών πόρων και των καταναλωτών πόρων αν γίνει κάποια αλλαγή στο λεξιλόγιο των περιγραφών. Η γνώση των πεδίων αποτελείται από ένα σύνολο κανόνων που έχει δημιουργηθεί χρησιμοποιώντας το λεξικό που έχει οριστεί από τις οντολογίες. Οι κανόνες έχουν εκφραστεί χρησιμοποιώντας το σύστημα κανόνων TRIPLE¹⁰. Τέλος, οι κανόνες ταιριάσματος ορίζουν τους περιορισμούς μεταξύ των περιγραφών των πόρων και των αιτήσεων και εκφράζονται χρησιμοποιώντας την γλώσσα επερωτήσεων TRIPLE. Χρησιμοποιώντας αυτές τις οντολογίες και τους κανόνες, υποστηρίζονται τα αντίστοιχα πρωτόκολλα για την διαφήμιση των πόρων, την υποβολή

¹⁰triple.semanticweb.org/

επερωτήσεων για πόρους και την διαπραγμάτευση με παροχείς πόρων.

Η τελευταία εργασία αυτής της ανασκόπησης, το άρθρο [43] επεκτείνει την εργασία [104] και παρέχει μια υπηρεσία ταιριάσματος που βασίζεται στις οντολογίες και καλείται OMMS (Ontology-based Matchmaking Service) ως μια *υπηρεσία Πλέγματος* (Grid Service). Το πλεονέκτημα αυτής της προσέγγισης είναι ότι η υπηρεσία OMMS μπορεί να εκμεταλλευτεί τις υπηρεσίες ανακάλυψης και παρακολούθησης (MDS) που παρέχονται σε ένα περιβάλλον Πλέγματος για να ανακαλύψει και να ενημερώσει τις πληροφορίες σχετικές με πόρους. Η υπηρεσία OMMS μπορεί να ‘εγγραφεί’ σε παροχείς υπηρεσιών με αυτή την λειτουργικότητα για να διαχειριστεί τις ενημερώσεις. Επιπλέον, η υπηρεσία OMMS ρωτάει περιοδικά τους παροχείς που έχουν αυτήν την δυνατότητα. Τέλος, η υπηρεσία OMMS χρησιμοποιεί έναν μηχανισμό *soft-state* για την διατήρηση πληροφοριών πόρων. Δηλαδή, εάν μια περιγραφή ενός πόρου δεν έχει ενημερωθεί κατά την διάρκεια ζωής του αφαιρείται από την βάση γνώσης της υπηρεσίας OMMS.

Οι παραπάνω σημαιολογικές προσεγγίσεις για την ανακάλυψη πόρων και υπηρεσιών συνοψίζονται στον Πίνακα 3.4.

3.4 Συμπεράσματα

Σε αυτό το κεφάλαιο παρουσιάσαμε ένα μέρος της σχετικής έρευνας στις περιοχές που σχετίζονται με αυτήν την διπλωματική εργασία. Ξεκινήσαμε την παρουσίαση με εργασίες σχετικές με την διαχείριση RDF δεδομένων σε δίκτυα ομότιμων κόμβων. Έπειτα παρουσιάσαμε τις *state of the art* εργασίες στην περιοχή της ανάκτησης σημαιολογικής πληροφορίας για υπηρεσίες του παγκόσμιου ιστού και της ανακάλυψης υπηρεσιών και πόρων στο Πλέγμα.

¹Περιγραφική Λογική

Εργασία	Μοντέλο περι-γραφής πόρου-υπηρεσίας	Μοντέλο αιτήσεων-επερωτήσεων	Τεχνική ταιριάσματος	Αρχιτεκτονική
<i>Heine et al.</i> [45]	προσθήκες ΠΛ	γλώσσα επερωτήσεων ΠΛ ¹	συλλογισμός μέσω του RAC-ER σε κάθε κόμβο	Καταμεμημένη (Chord)
<i>Chen et al.</i> [23]	OWL	Γραφική διεπαφή με χρήση οντολογιών	συλλογισμός μέσω του FaCT	Κεντρικοποιημένη
<i>Wroe et al.</i> [112]	οντολογίες DAML-S	γλώσσα επερωτήσεων ΠΛ	συλλογισμός μέσω του FaCT	Κεντρικοποιημένη
<i>Lord et al.</i> [62]	RDFS	συγκεκριμένα RDQL επερωτήσεις	απάντηση επερωτήσεων χρησιμοποιώντας το Jena	Κεντρικοποιημένη
<i>Harth et al.</i> [43, 104]	RDFS	γλώσσα επερωτήσεων TRIPLE	απάντηση επερωτήσεων χρησιμοποιώντας το σύστημα κανόνων TRIPLE	Κεντρικοποιημένη

Πίνακας 3.4: Σημαιολογική ανακάλυψη πόρων και υπηρεσιών στο Πλέγμα

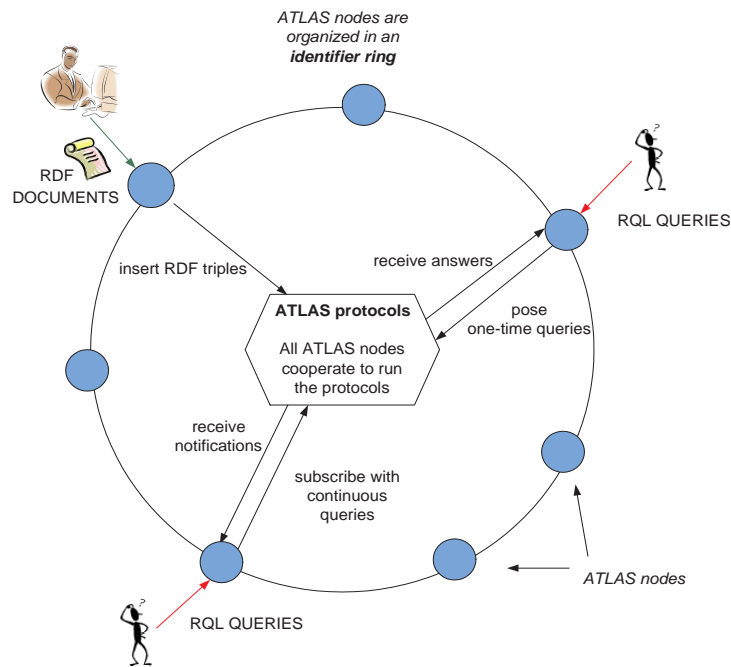
Κεφάλαιο 4

Atlas: Αποθήκευση, ενημέρωση και αποτίμηση επερωτήσεων RDF(S) δεδομένων χρησιμοποιώντας το ΚΠΚ Bamboo

Σε αυτό το κεφάλαιο θα παρουσιάσουμε το σύστημα Atlas v0.5. Το σύστημα Atlas είναι ένα σύστημα που χρησιμοποιεί ένα δίκτυο ομότιμων κόμβων για να αποθηκεύσει μεταδεδομένα RDF(S) και να απαντήσει επερωτήσεις σχετικές με αυτά με κατακευματισμένο τρόπο. Τα μεταδεδομένα RDF(S) περιγράφουν διάφορους πόρους. Χρησιμοποιούμε την τεχνολογία των κατακευματισμένων πινάκων κατακευματισμού για να μπορέσουμε να σχεδιάσουμε και να υλοποιήσουμε ένα κατακευματισμένο σύστημα με δυνατότητα κλιμάκωσης σε μεγάλα δυναμικά δίκτυα και απάντησης επερωτήσεων πάνω από ένα μεγάλο σύνολο δεδομένων RDF. Στο υπόλοιπο κεφάλαιο θα παρουσιάσουμε λεπτομερώς την κατακευματισμένη αρχιτεκτονική του Atlas, το μοντέλο δεδομένων που χρησιμοποιεί, την γλώσσα επερωτήσεων, τις λειτουργίες που προσφέρει και τα πρωτόκολλα που υλοποιούν αυτές τις λειτουργίες. Θα παρουσιάσουμε επίσης την κατάσταση της τρέχουσας υλοποίησης του συστήματος και θα συζητήσουμε την μελλοντική εργασία.

4.1 Η αρχιτεκτονική του συστήματος Atlas

Οι κόμβοι στο δίκτυο του Atlas οργανώνονται χρησιμοποιώντας το πρωτόκολλο του ΚΠΚ Bamboo που έχουμε ήδη περιγράψει στην Ενότητα 2.4.3. Οι κόμβοι μπορούν να εισάγουν RDF(S) δεδομένα στο δίκτυο και να θέτουν RQL επερωτήσεις. Στην τυπική εφαρμογή του Atlas που οραματιζόμαστε, τα RDF(S) δεδομένα θα χρησιμοποιούνται για να περιγράψουν πόρους που ανήκουν σε κόμβους του δικτύου (για παράδειγμα οι υπηρεσίες μπορούν να θεωρηθούν τέτοιοι πόροι, όπως θα δούμε στην Ενότητα 5.3). Το σύστημα Atlas υποστηρίζει δύο σενάρι-



Σχήμα 4.1: Αρχιτεκτονική του Atlas

α υποβολής επερωτήσεων. Το σενάριο όπου υποβάλλονται απλές επερωτήσεις (one-time queries) και το σενάριο όπου κάποιος κάνει συνδρομή με συνεχείς επερωτήσεις (continuous queries). Κάθε φορά που ένας κόμβος θέτει μία απλή επερώτηση, οι υπόλοιποι κόμβοι του δικτύου συνεργάζονται για να βρουν τα RDF(S) δεδομένα που συνθέτουν μια απάντηση σε αυτήν την επερώτηση. Στο σενάριο συνεχών επερωτήσεων ένας κόμβος θέτει μια συνεχή επερώτηση, αυτή η επερώτηση ευρετηριάζεται κάπου στο δίκτυο και κάθε φορά που δημοσιεύονται δεδομένα που απαντάνε αυτήν την επερώτηση, οι κόμβοι συνεργάζονται για να ειδοποιήσουν τον συνδρομητή κόμβο. Μία υψηλού επιπέδου άποψη της αρχιτεκτονικής του Atlas απεικονίζεται στο Σχήμα 4.1.

Στην υπόλοιπη ενότητα θα επικεντρωθούμε σε σενάρια απλών επερωτήσεων. Ιδέες σχετικά με τα σενάρια συνεχών επερωτήσεων αναλύονται στο άρθρο [61].

4.1.1 Η επιλογή του ΚΠΚ Bamboo

Στο αρχικό στάδιο του σχεδιασμού του συστήματος, έπρεπε να επιλέξουμε ένα κατάλληλο πρωτόκολλο ΚΠΚ για την υλοποίηση του Atlas. Οι κύριες επιλογές μας ήταν το FreePastry¹, που αποτελεί μια υλοποίηση του πρωτοκόλλου Pastry από το Πανεπιστήμιο Rice, η υλοποίηση του MIT του πρωτοκόλλου Chord² και

¹<http://www.cs.rice.edu/CS/Systems/Pastry/>

²<http://www.pdos.lcs.mit.edu/chord/>

το Bamboo³ από το Πανεπιστήμιο UC Berkeley. Υπήρχε επίσης ένα διαθέσιμο σύστημα από την Ευρώπη, το P-Grid⁴ [1]. Τα συστήματα DKS⁵ [4] και Tango [20] έγιναν διαθέσιμα αφού ξεκινήσαμε την υλοποίηση μας.

Η επιλογή μας επηρεάστηκε κυρίως από την τρέχουσα κατάσταση της υλοποίησης του κάθε διαφορετικού συστήματος και ιδιαίτερα από την γλώσσα προγραμματισμού που χρησιμοποιήθηκε καθώς και από την αναπτυξιακή υποστήριξη που παρέχόταν. Το Bamboo έχει υλοποιηθεί στην γλώσσα προγραμματισμού Java, γεγονός που καθιστά ευκολότερη την εκτέλεση του σε πολλαπλές μηχανές χωρίς περιορισμούς που να αφορούν στο λειτουργικό σύστημα. Επιπλέον, θεωρείται ότι αποτελεί μια ώριμη υλοποίηση ΚΠΚ, έχει πολύ καλή υποστήριξη μέσω μίας λίστας ηλεκτρονικού ταχυδρομείου καθώς και επαρκής τεκμηρίωση του κώδικα.

Όπως συζητήθηκε στην Ενότητα 2.4.3, ένα βασικό πλεονέκτημα του Bamboo, σε σύγκριση με άλλα δημοφιλή πρωτοκόλλα ΚΠΚ, είναι η δυνατότητα του να χειρίζεται καλύτερα υψηλούς ρυθμούς αφίξεων και αποχωρήσεων από το δίκτυο. Ο ρυθμός αυτός αποτελεί μια κρίσιμη παράμετρο ιδιαίτερα για συστήματα μεγάλης κλίμακας. Για παράδειγμα, το γεγονός ότι οι κόμβοι μπορούν να αποχωρήσουν από το δίκτυο την ώρα που γίνεται επεξεργασία μιας επερώτησης με καταμεμημένο τρόπο μπορεί να μην επιτρέπει την ανάκτηση των κατάλληλων απαντήσεων. Αυτό μπορεί να γίνει κυρίως λόγω αλλοιωμένων πινάκων δρομολόγησης (π.χ. οι πίνακες μπορεί να μην περιέχουν πλέον έγκυρες εγγραφές και να πρέπει να ανανεωθούν). Επομένως, το γεγονός ότι το Bamboo έχει όλες τις καλές ιδιότητες των ΚΠΚ σε συνδυασμό με την αντοχή σε υψηλούς ρυθμούς αφίξεων και αναχωρήσεων αποτέλεσε ακόμη ένα κίνητρο για την επιλογή του.

Ας αναφέρουμε τώρα κάποιες τεχνικές λεπτομέρειες σχετικά με την υλοποίηση του συστήματος Bamboo. Το Bamboo έχει δημιουργηθεί χρησιμοποιώντας ένα μονονηματικό προγραμματιστικό στυλ καθοδηγούμενο από τα γεγονότα (single-threaded event-driven programming style) [90]. Το Bamboo έχει κληρονομήσει αυτήν την δομή από την αρχιτεκτονική (SEDA (Staged Event-Driven Architecture)) [111] η οποία βασίζεται στην χρήση σταδίων (stages). Επομένως, κάθε μέρος μιας εφαρμογής του Bamboo αποτελεί ένα στάδιο. Αυτά τα στάδια επικοινωνούν μεταξύ τους στέλνοντας γεγονότα το ένα στο άλλο, ενώ η εφαρμογή παραμένει μονονηματική. Ένα στάδιο λέμε ότι εγγράφει τον εαυτό του στο Bamboo και με αυτόν τον τρόπο γίνεται συνδρομητής των γεγονότων που θέλει να λαμβάνει καθώς και να στέλνει μηνύματα σε άλλα στάδια ή άλλους κόμβους.

4.2 Μοντέλο Δεδομένων και Γλώσσα Επερωτήσεων

Οι κόμβοι του Atlas χρησιμοποιούν το μοντέλο δεδομένων του RDF (βλέπε Ενότητα 2.5) και παρέχουν τα δεδομένα τους με την μορφή RDF εγγράφων. Αυτά τα έγ-

³<http://bamboo-dht.org/>

⁴<http://www.p-grid.org/>

⁵<http://dks.sics.se/>

γραφα αποσυντίθενται σε RDF τριπλέτες, οι οποίες αποθηκεύονται σε διάφορους κόμβους του δικτύου. Χρησιμοποιούμε την τυπική έννοια του όρου τριπλέτα, δηλαδή μία τριπλέτα αναπαριστά μία δήλωση για ένα πεδίο και έχει την μορφή (υποκείμενο, κατηγορημα, αντικείμενο) [59].

Στην υπόλοιπη ενότητα θα ορίσουμε τους διαφορετικούς τύπους επερωτήσεων που μπορεί να διαχειριστεί το σύστημα Atlas v0.5. Το Atlas v0.5 υποστηρίζει διαζεύξεις επερωτήσεων σύζευξης. Ο στόχος είναι η επέκταση του συστήματος Atlas ώστε να υποστηρίζει ολόκληρη ή ένα μεγάλο υποσύνολο της γλώσσας RQL.

4.2.1 Επερωτήσεις σύζευξης

Αρχικά θα ορίσουμε τις επερωτήσεις σύζευξης. Μία επερωτήση σύζευξης έχει την ακόλουθη μορφή:

$$?x_1, \dots, ?x_n : (s_1, p_1, o_1) \wedge (s_2, p_2, o_2) \wedge \dots \wedge (s_n, p_n, o_n)$$

όπου s_1, \dots, s_n είναι μεταβλητές ή URIs, p_1, \dots, p_n είναι μεταβλητές ή URIs και o_1, \dots, o_n είναι μεταβλητές ή URIs ή κυριολεκτικά. $?x_1, \dots, ?x_n$ είναι μεταβλητές, τις οποίες ονομάζουμε μεταβλητές απάντησης (*answer variables*) και ισχύει

$$\{?x_1, \dots, ?x_n\} \subseteq \{s_1, \dots, s_n, p_1, \dots, p_n, o_1, \dots, o_m\}.$$

Οι μεταβλητές ξεκινάνε πάντοτε με τον χαρακτήρα “?”.

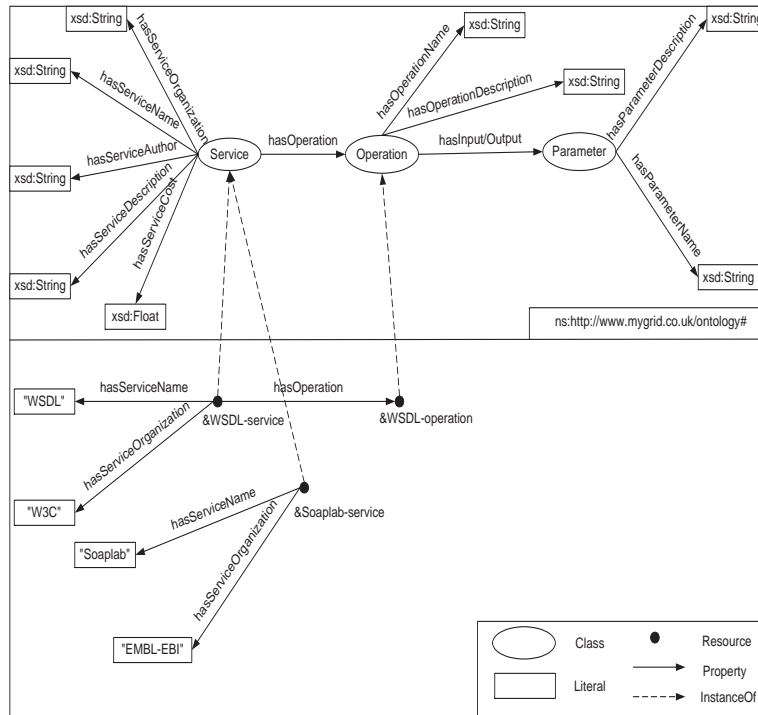
Όπως και στο άρθρο [19], οι ακολουθίες $(s_1, p_1, o_1), \dots, (s_n, p_n, o_n)$ ονομάζονται σχέδια τριπλέτων (triple patterns). Για να μπορέσουν να δουλέψουν σωστά τα πρωτόκολλα του Atlas, κάθε σχέδιο τριπλέτων πρέπει να περιέχει τουλάχιστον μια σταθερά, δηλαδή ένα από τα υποκείμενο, κατηγορημα και αντικείμενο της αντίστοιχης τριπλέτας να μην αποτελεί μεταβλητή. Η εξήγηση για αυτή την υπόθεση θα δοθεί στην συνέχεια.

Στην συνέχεια αναφέρουμε κάποια παραδείγματα επερωτήσεων σύζευξης. Ας θεωρήσουμε το σχήμα RDFS που παρουσιάζεται στο Σχήμα 4.2. Αυτό το σχήμα περιγράφει υπηρεσίες και αποτελεί μια επεκταμένη έκδοση του βασικού μοντέλου δεδομένων για υπηρεσίες που χρησιμοποιήθηκε από το πρόγραμμα *myGrid*⁶. Σε αυτό το σενάριο, οι κόμβοι του Atlas δημοσιεύουν μεταδεδομένα που περιγράφουν τις υπηρεσίες που παρέχουν. Οι κόμβοι που αναζητούν υπηρεσίες μπορούν να θέσουν μια κατάλληλη επερωτήση στο σύστημα Atlas σχετική με τα χαρακτηριστικά των υπηρεσιών για τις οποίες ενδιαφέρονται. Έπειτα θα λάβουν τις απαντήσεις οι οποίες θα προκύψουν με την συνεργασία πολλών κόμβων του Atlas. Η μορφή των απαντήσεων θα περιγραφεί στην Ενότητα 4.3.3.

Ακολουθούν παραδείγματα επερωτήσεων σύζευξης για αυτό το σενάριο:

- Βρες τις υπηρεσίες που προσφέρονται από τον οργανισμό “Fiat” και ο συγγραφέας τους ονομάζεται “John Smith”.

⁶<http://www.mygrid.org.uk>



Σχήμα 4.2: Σχήμα RDFS για υπηρεσίες του παγκοσμίου ιστού

$$?x : (?x, hasServiceOrganization, "Fiat") \wedge$$

$$(?x, hasServiceAuthor, "John Smith")$$

- Βρες τις παραμέτρους εισόδου απαιτεί η υπηρεσία *service89334*

$$?in : (service89334, hasOperation, ?op) \wedge (?op, hasOperationInput, ?in)$$

Μία ατομική επερώτηση αποτελεί μία ειδική περίπτωση των επερωτήσεων σύζευξης όταν αυτές περιλαμβάνουν ένα μόνο σχέδιο τριπλέτων. Ένα απλό παράδειγμα μια ατομικής επερώτησης είναι:

- Βρες την υπηρεσία με id 929208.

$$?x : (?x, service_id, 929208)$$

4.2.2 Επερωτήσεις διάζευξης

Ο άλλος τύπος επερωτήσεων που υποστηρίζεται από το Atlas v0.5 είναι οι επερωτήσεις διάζευξης. Οι επερωτήσεις διάζευξης προκύπτουν προσθέτοντας την διάζευξη σε μια επερώτηση σύζευξης. Επομένως, μια επερώτηση διάζευξης έχει την ακόλουθη μορφή

$$\phi_1 \vee \phi_2 \vee \dots \vee \phi_k$$

όπου οι $\phi_1, \phi_2, \dots, \phi_k$ είναι επερωτήσεις σύζευξης. Ας χρησιμοποιήσουμε το RDFS του Σχήματος 4.2 για να δώσουμε ένα παράδειγμα επερώτησης διάζευξης:

- Βρες τις υπηρεσίες που είτε ονομάζονται "TravelAgent" και έχουν ως συγγραφέα τον "John Smith" είτε ονομάζονται "TravelTour" και έχουν συγγραφέα τον "John Smith".

$?x : ((?x, hasServiceName, "TravelAgent") \wedge (?x, hasServiceAuthor, "JohnSmith")) \vee$

$((?x, hasServiceName, "TravelTour") \wedge (?x, hasServiceAuthor, "JohnSmith"))$

4.3 Πρωτόκολλα

Σε αυτήν την ενότητα θα περιγράψουμε τα διάφορα πρωτόκολλα που χρησιμοποιούν οι κόμβοι του Atlas για να παρέχουν την επιθυμητή λειτουργικότητα. Αρχικά, περιγράφουμε λεπτομερώς τον τρόπο με τον οποίο οι κόμβοι συνδέονται στο δίκτυο και τον τρόπο που αποχωρούν από αυτό. Έπειτα, περιγράφουμε τα πρωτόκολλα για την αποθήκευση RDF εγγράφων και την διαχείριση των διαφόρων τύπων επερωτήσεων. Τα πρωτόκολλα που περιγράφονται παρακάτω χρησιμοποιούνται στο σύστημα Atlas v0.5.

4.3.1 Σύνδεση και αποχώρηση από το δίκτυο του Atlas

Όπως έχουμε ήδη συζητήσει, οι κόμβοι του Atlas συνδέονται μέσω του δικτύου Bamboo. Στην τρέχουσα υλοποίηση του Atlas τα αρχικά πρωτόκολλα του Bamboo για την σύνδεση και την αποχώρηση κόμβων έχουν παραμείνει αναλλοίωτα. Στην Ενότητα 2.4.3 συζητήσαμε τους λόγους που οδήγησαν τους Rhea κ.ά. να αναπτύξουν τα συγκεκριμένα πρωτόκολλα σύνδεσης και αποχώρησης για το Bamboo βελτιώνοντας τα πρωτόκολλα του Pastry [80, 90]. Ακολουθεί περιγραφή των πρωτοκόλλων.

Κάθε φορά που ένας νέος κόμβος x θέλει να συνδεθεί (join) στο δίκτυο, χρειάζεται να βρει την IP διεύθυνση και τον αριθμό της port ενός κόμβου y που είναι ήδη στο δίκτυο, δηλαδή ενός κόμβου που είναι ήδη κόμβος του Atlas. Ένας τέτοιος κόμβος μπορεί να βρεθεί με διάφορους τρόπους, π.χ. μέσω ενός ασφαλούς δικτυακού ιστότοπου που περιέχει τις IP διεύθυνσεις των κόμβων που είναι ήδη συνδεδεμένοι. Αυτός ο κόμβος y καλείται κόμβος πύλη (gateway) ή κόμβος εκκίνησης (bootstrap node) του κόμβου x . Αφού έχει βρει τον κόμβο πύλη y , ο x στέλνει το ακόλουθο μήνυμα σε αυτόν:

`JOIN-REQUEST(gateway_address, node_address, guid, rev_ttl)`

όπου $gateway_address$ είναι η IP διεύθυνση και ο αριθμός της port του κόμβου y και $node_address$ είναι η IP διεύθυνση και ο αριθμός της port του x . Τέλος, η παράμετρος $guid$ αποτελεί το αναγνωριστικό του κόμβου x που έχει υπολογιστεί ως η τιμή κατακερματισμού με χρήση της συνάρτησης SHA-1 [73] της IP διεύθυνσης του x .

Όταν ο κόμβος y λάβει ένα τέτοιο μήνυμα, το προωθεί στον κόμβο που έχει το κοντινότερο αναγνωριστικό στο $guid$. Η χρήση του $guid$ διευκολύνει τον κόμβο y στην προώθηση του μηνύματος γιατί δεν χρειάζεται να υπολογίσει το αναγνωριστικό του $node_address$. Η ίδια διαδικασία ακολουθείται για κάθε κόμβο που βρίσκεται στο μονοπάτι αναζήτησης μέχρι το μήνυμα να φτάσει στον κόμβο z . Τέλος, ο z στέλνει το ακόλουθο μήνυμα πίσω στον x :

JOIN-RESPONSE($return_address, path, leaf_set$)

όπου $return_address$ είναι η IP διεύθυνση του κόμβου x , $path$ είναι το μονοπάτι αναζήτησης των κόμβων μέχρι το μήνυμα JOIN-REQUEST να φτάσει στον προορισμό του και $leaf_set$ είναι το σύνολο φύλλων του κόμβου z . Όταν ο κόμβος x λάβει το μήνυμα, προσθέτει τους κόμβους από το μονοπάτι αναζήτησης $path$ στον πίνακα δρομολόγησης χρησιμοποιώντας τις πληροφορίες στο JOIN-RESPONSE μήνυμα και αποκτά το αρχικό σύνολο φύλλων από το $leaf_set$ το οποίο περιέχει το σύνολο φύλλων του κόμβου z . Έπειτα, ο x θεωρεί ότι έχει συνδεθεί με επιτυχία στο δίκτυο. Στην συνέχεια, ο πίνακας δρομολόγησης του x θα συμπληρωθεί και θα βελτιστοποιηθεί από μια συνεχή περιοδική διαδικασία που εκτελείται μέχρι ο κόμβος να αποχωρήσει από το δίκτυο.

Θα περιγράψουμε λεπτομερώς την συνεχή διαδικασία που είναι υπεύθυνη για την διατήρηση των πινάκων δρομολόγησης σε κάθε κόμβο. Για κάθε κόμβο x ένας αλγόριθμος συντήρησης χρησιμοποιείται περιοδικά για να ενημερωθεί ο πίνακας δρομολόγησης του x με τους γείτονες που είναι κοντινότερα σε αυτόν χρησιμοποιώντας κάποιο μέτρο εγγύτητας. Για να επιτευχθεί αυτό, χρησιμοποιούνται δύο τεχνικές. Αρχικά αναφέρουμε ότι με $R_l[i]$ αναφερόμαστε στην εγγραφή της γραμμής l και στήλης i στον πίνακα δρομολόγησης. Σε αυτή την εγγραφή ο γειτονικός κόμβος y έχει ένα αναγνωριστικό που ταιριάζει με ακριβώς l ψηφία του κόμβου x και το $(l + 1)$ -στό ψηφίο είναι το i (βλέπε Ενότητα 2.4.2). Η πρώτη τεχνική συντήρησης του πίνακα δρομολόγησης ονομάζεται *καθολικός συντονισμός* (global tuning). Όταν ο κόμβος x θέλει να συμπληρώσει ή να βελτιώσει μία εγγραφή $R_l[i]$, καλεί μια λειτουργία αναζήτησης για ένα αναγνωριστικό id τέτοιο ώστε τα πρώτα l ψηφία του id είναι τα ίδια με τα πρώτα l ψηφία του αναγνωριστικού του x . Το $l + 1$ -στό ψηφίο του id είναι το i και τα υπόλοιπα είναι τυχαία ψηφία. Ο κόμβος που επιστρέφεται, δεδομένου ότι είναι κοντινότερα από την υπάρχουσα εγγραφή, χρησιμοποιείται για την ανανέωση του πίνακα δρομολόγησης του x . Αυτή η τεχνική είναι παρόμοια με το *πρωτόκολλο σταθεροποίησης* (stabilization protocol) του Chord. Στην δεύτερη τεχνική που ονομάζεται *τοπικός συντονισμός* (local tuning) ένας κόμβος x επικοινωνεί περιοδικά με ένα τυχαίο μέλος $R_l[i]$ του πίνακα δρομολόγησης του x και ζητάει όλες τις εγγραφές από το επίπεδο

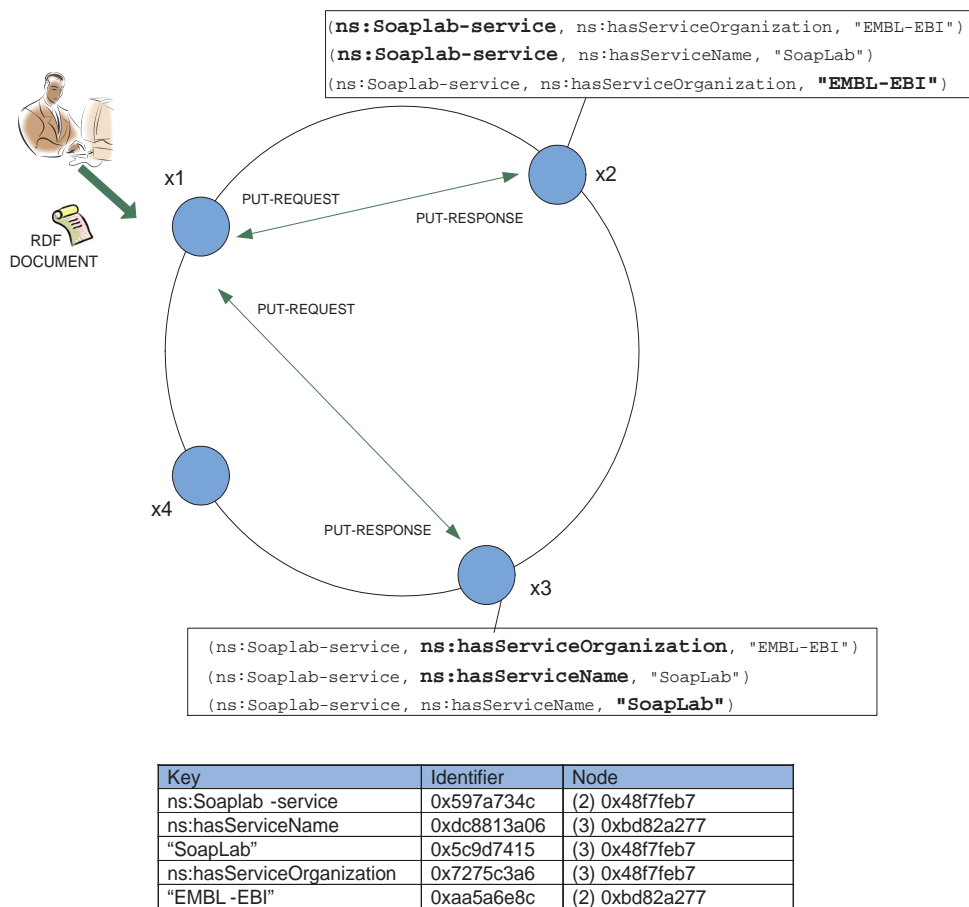
l. Αν μία από τις επιστρεφόμενες εγγραφές συμπληρώνει ένα κενό στον πίνακα δρομολόγησης του x τότε χρησιμοποιείται. Οι υπόλοιπες παρακολουθούνται για καθυστέρηση πριν την πιθανή αντικατάσταση των υπάρχοντων εγγραφών στον πίνακα δρομολόγησης του x . Αυτή η τεχνική είναι παρόμοια με την συντήρηση των πινάκων δρομολόγησης στο πρωτόκολλο Pastry εξαιρώντας το γεγονός ότι το Bamboo την εκτελεί αυξητικά και συχνότερα.

Ο πρώτος κόμβος r ο οποίος δημιουργεί το δίκτυο θεωρείται ειδικός κόμβος και ξεκινάει με τον εαυτό του σαν κόμβο πύλη. Ο κόμβος r αποτελεί τον ριζικό κόμβο του δικτύου και συνδέεται άμεσα σε αυτό.

Οι κόμβοι του δικτύου μπορεί να αποτύχουν ή να αποχωρήσουν χωρίς ειδοποίηση. Ένας κόμβος x θεωρείται ότι έχει αποτύχει όταν οι άμεσοι γείτονες του δεν μπορούν να επικοινωνήσουν πλέον με αυτόν. Οι κόμβοι των οποίων οι πίνακες δρομολόγησης και τα σύνολα φύλλων περιέχουν τον κόμβο x αντικαθιστούν το κόμβο που απέτυχε με έναν άλλο. Το Bamboo διαφέρει από τα υπόλοιπα δίκτυα ΚΠΚ στον τρόπο που επιλέγονται τα χρονικά όρια (timeouts). Τα χρονικά όρια αναφέρονται στο χρονικό διάστημα που ένας κόμβος πρέπει να περιμένει για μια απάντηση προτού να θεωρήσει ότι ο παραλήπτης κόμβος έχει αποτύχει. Εάν αυτό το διάστημα είναι πολύ μικρό, ο κόμβος μπορεί να μην έχει προλάβει να λάβει το μήνυμα ή η απάντηση μπορεί να έχει προλάβει να τοποθετηθεί σε ουρά στο δίκτυο. Ενώ αν το διάστημα είναι πολύ μεγάλο, ένας κόμβος μπορεί να σπαταλήσει χρόνο να περιμένει για μια απάντηση από έναν κόμβο που έχει ήδη αποτύχει. Το Bamboo επιλέγει τα χρονικά όρια χρησιμοποιώντας δύο συμπληρωματικές τεχνικές. Η πρώτη τεχνική χρησιμοποιεί μια πρόσφατη εκτίμηση του χρόνου απόκρισης για κάθε έναν από τους γείτονες του κάθε κόμβου. Η δεύτερη τεχνική βασίζεται στην χρήση της λεγόμενης αναδρομικής δρομολόγησης (recursive routing) αντί της επαναληπτικής δρομολόγησης (iterative routing). Στην αναδρομική δρομολόγηση μια αίτηση προωθείται από κόμβο σε κόμβο μέχρι να φτάσει στο προορισμό της, ενώ στην επαναληπτική δρομολόγηση γίνεται άμεση επικοινωνία του αρχικού κόμβου με κάθε ενδιαμέσο. Η χρήση της αναδρομικής δρομολόγησης καθιστά δυνατή την δημιουργία ενός πρόσφατου εκτιμητή του χρόνου απόκρισης [80, 90].

4.3.2 Αποθηκεύοντας RDF δεδομένα

Για να αποθηκεύσει ένας κόμβος δεδομένα RDF, πρέπει να τα διαθέσει στην μορφή ενός RDF εγγράφου. Αυτό το έγγραφο χωρίζεται τοπικά από τον κόμβο στις RDF τριπλέτες από τις οποίες αποτελείται. Έπειτα, το δίκτυο διαχειρίζεται κάθε τριπλέτα ξεχωριστά έτσι ώστε να κατανοηθεί ο φόρτος αποθήκευσης του εγγράφου καθώς και της απάντησης επερωτήσεων σε ένα μεγάλο μέρος του δικτύου. Έχουμε υιοθετήσει τον αλγόριθμο ευρετηρίασης τριπλέτων που παρουσιάζεται στο [19] όπου κάθε τριπλέτα αποθηκεύεται *τρεις φορές* (ενδεχομένως σε τρεις διαφορετικούς κόμβους), μία βάσει του υποκειμένου της, μία βάσει του κατηγορήματος της και μία βάσει του αντικειμένου της. Δηλαδή, για κάθε αποθήκευση



Σχήμα 4.3: Αποθηκεύοντας ένα έγγραφο RDF

χρησιμοποιούμε ως κλειδί την τιμή του υποκειμένου, του κατηγορήματος και του αντικειμένου αντίστοιχα. Το κλειδί κατακερματίζεται (χρησιμοποιώντας την συνάρτηση κατακερματισμού SHA-1) για να δημιουργηθεί το αναγνωριστικό που θα οδηγήσει στον κατάλληλο κόμβο όπου θα αποθηκευτεί η τριπλέτα.

Τα αναγνωριστικά που δημιουργούνται με τον κατακερματισμό των κλειδιών περιλαμβάνουν επιπλέον πληροφορία για το αν τα κλειδιά αναφέρονται στο υποκείμενο, στο κατηγορήμα ή στο αντικείμενο. Αυτό επιτυγχάνεται προσθέτοντας την πληροφορία αυτή στο τέλος των αναγνωριστικών με την χρήση δύο επιπλέον δυφίων. Επομένως, γνωρίζοντας εκ των προτέρων αν το κλειδί αναφέρεται στο υποκείμενο, στο κατηγορήμα ή στο αντικείμενο, μειώνουμε τα αποτελέσματα που ενδεχομένως ταιριάζουν με μια επερώτηση.

Οι τριπλέτες που εισάγονται στο δίκτυο έχουν συγκεκριμένη *ημερομηνία λήξης* όπως στο άρθρο [19]. Ο κόμβος που τις αποθηκεύει είναι υπεύθυνος για την ανανέωση τους πριν αυτές *λήξουν*. Αυτό αποτελεί μία κλασική ιδέα που χρησιμοποιείται σε *soft-state* πρωτόκολλα ανανέωσης που χρησιμοποιούνται στα ΚΠΚ

[80] καθώς και σε υπηρεσίες παρακολούθησης και ανακάλυψης Πλέγματος όπως η υπηρεσία MDS [28].

Για κάθε τριπλέτα ενός RDF εγγράφου ο κόμβος x που αποθηκεύει τα δεδομένα στέλνει τρία διαφορετικά μηνύματα της ακόλουθης μορφής

PUT-REQUEST($id, triple, ttl_sec, node_address, requestid$)

όπου ttl_sec είναι το χρονικό διάστημα κατά το οποίο η αποθήκευση θεωρείται έγκυρη, $node_address$ είναι η IP διεύθυνση του x και $requestid$ είναι ένα μοναδικό αναγνωριστικό για αυτήν την αίτηση. Το πρώτο από τα τρία μηνύματα έχει σαν id το αναγνωριστικό που υπολογίζεται κατακερματίζοντας το υποκείμενο της τριπλέτας, το δεύτερο έχει σαν id το αναγνωριστικό που υπολογίζεται κατακερματίζοντας το κατηγορημα της τριπλέτας και το τρίτο μήνυμα έχει σαν id το αναγνωριστικό που υπολογίζεται κατακερματίζοντας το αντικείμενο της τριπλέτας. Η τριπλέτα που φέρει το κάθε μήνυμα αποθηκεύεται στον κόμβο που είναι αριθμητικά κοντινότερος στο id και ο οποίος ονομάζεται ριζικός κόμβος του συγκεκριμένου id . Επιπλέον, αποθηκεύεται και σε όλους, εκτός από έναν, από τους άμεσους $leaf_set_size$ προκατόχους και διαδόχους, όπου $leaf_set_size$ είναι μία διαμορφωτική παράμετρος του συστήματος Bamboo. Ο κόμβος μέλος του συνόλου φύλλων που παραλείπεται είναι ο τελευταίος διάδοχος αν το κλειδί προηγείται του ριζικού κόμβου ή ο πρώτος προκατόχος αν το κλειδί διαδέχεται τον ριζικό κόμβο. Επομένως, υπάρχουν $2 * leaf_set_size$ αντίγραφα (replicas) της κάθε τριπλέτας περιλαμβανομένης της τριπλέτας που είναι αποθηκευμένη στον ριζικό κόμβο.

Κάθε κόμβος που λαμβάνει ένα μήνυμα PUT-REQUEST απαντάει με το ακόλουθο μήνυμα

PUT-RESPONSE($result, requestid$)

για να ενημερώσει αν η αποθήκευση ολοκληρώθηκε με επιτυχία. Στο μήνυμα αυτό, η παράμετρος $result$ καθορίζει αν η αίτηση ήταν επιτυχής και η παράμετρος $requestid$ καθορίζει την αίτηση στην οποία αναφέρεται η απάντηση.

Στο Σχήμα 4.3, βλέπουμε ένα παράδειγμα αποθήκευσης ενός RDF εγγράφου που αποτελείται από τις ακόλουθες δύο τριπλέτες οι οποίες έχουν δημιουργηθεί χρησιμοποιώντας το RDFS του Σχήματος 4.2:

```
(<ns:Soaplab-service>, <ns:hasServiceName>, "SoapLab")  
(<ns:Soaplab-service>, <ns:hasServiceOrganization>, "EMBL-EBI")
```

Ο κόμβος x_1 είναι ο κόμβος που θέλει να αποθηκεύσει αυτό το RDF έγγραφο. Ο πίνακας που εμφανίζεται στο Σχήμα 4.3 δείχνει τα πέντε διαφορετικά κλειδιά και τα αντίστοιχα αναγνωριστικά που έχουν δημιουργηθεί. Τόσο τα κλειδιά όσο και τα αναγνωριστικά αναπαριστώνται στο δεκαεξαδικό σύστημα. Επίσης, εμφανίζονται οι υπεύθυνοι κόμβοι που θα αποθηκεύσουν τις τριπλέτες σύμφωνα με το πρωτόκολλο που παρουσιάστηκε. Για λόγους αναγνωσιμότητας, δείχνουμε μόνο

τους ριζικούς κόμβους των κλειδιών των τριπλετών και όχι τους γείτονες αυτών των κόμβων που επίσης θα αποθήκευαν τις τριπλέτες. Σε κάθε κόμβο μπορούμε να δούμε τις τριπλέτες που έχει αποθηκεύσει. Το κλειδί που έχει χρησιμοποιηθεί για την αποθήκευση της κάθε τριπλέτας εμφανίζεται με έντονη γραμματοσειρά.

4.3.3 Αποτίμηση επερωτήσεων

Όταν ένας κόμβος θέλει να ανακαλύψει έναν πόρο, πρέπει να θέσει μια επερώτηση στο δίκτυο του Atlas. Στο Atlas v0.5 χειριζόμαστε επερωτήσεις των τύπων που ορίστηκαν στην Ενότητα 4.2.

Η βασική ιδέα του πρωτοκόλλου επεξεργασίας επερωτήσεων είναι η ακόλουθη. Για κάθε σχέδιο τριπλέτων σε μια επερώτηση, χρησιμοποιούμε ένα σταθερό μέρος αυτού του σχεδίου ως κλειδί. Με τον κατακερματισμό του κλειδιού δημιουργούμε το αναγνωριστικό που θα οδηγήσει στον κόμβο που περιέχει τις σχετικές τριπλέτες. Έπειτα, αυτός ο κόμβος επιστρέφει τις απαντήσεις στον κόμβο ο οποίος έθεσε την επερώτηση.

Στην υπόλοιπη ενότητα περιγράφουμε λεπτομερώς πως το σύστημα Atlas διαχειρίζεται τους διαφορετικούς τύπους επερωτήσεων.

Ατομικές επερωτήσεις

Θα ξεκινήσουμε την περιγραφή των πρωτόκολλων επεξεργασίας επερωτήσεων για τον πιο απλό τύπο επερωτήσεων, τις ατομικές επερωτήσεις. Όταν ένας κόμβος x θέτει μια ατομική επερώτηση, στέλνει το ακόλουθο μήνυμα :

```
GET-REQUEST(id, triplePattern, node_address, variables, requestid)
```

όπου :

- *id* είναι το αναγνωριστικό που προκύπτει με κατακερματισμό μιας από τις σταθερές του *triplePattern*.
- *triplePattern* είναι το σχέδιο τριπλέτων της ατομικής επερώτησης.
- *node_address* είναι η διεύθυνση του κόμβου που έθεσε την επερώτηση.
- *variables* είναι μια λίστα με τις μεταβλητές απάντησης.
- *requestid* είναι το μοναδικό αναγνωριστικό της συγκεκριμένης αίτησης.

Για να δημιουργήσουμε το αναγνωριστικό *id* χρησιμοποιούμε ένα από τα σταθερά μέρη του σχεδίου τριπλετών. Ένα σχέδιο τριπλετών σε μια ατομική επερώτηση έχει τουλάχιστον ένα και το πολύ τρία σταθερά μέρη. Επομένως, έχουμε δυνατότητα επιλογής. Στην τρέχουσα υλοποίηση η σειρά προτίμησης είναι αντικείμενο, υποκείμενο και κατηγορημα. Αυτή η στρατηγική έχει στόχο να

καταναίμει τον φόρτο της επεξεργασίας επερωτήσεων όσο δυνατόν πιο ομαλά έτσι όπως προτείνεται στο άρθρο [19]. Αυτό επιτυγχάνεται γιατί ένα αντικείμενο γενικά θεωρείται ότι έχει ευρύτερο πεδίο τιμών από ότι ένα υποκείμενο ή ένα κατηγορημα και επομένως αναμένουμε ότι σε αυτή την περίπτωση θα συνεισφέρουν περισσότεροι κόμβοι για την επεξεργασία των επερωτήσεων σε σχέση με το πόσοι θα συνεισέφεραν εάν χρησιμοποιούσαμε το κατηγορημα ή το υποκείμενο.

Το μήνυμα `GET-REQUEST` δρομολογείται βάσει του αναγνωριστικού *id* στον κατάλληλο κόμβο *y*, ο οποίος είναι ο ριζικός κόμβος του *id*, χρησιμοποιώντας τον αλγόριθμο δρομολόγησης του Βαμβοο. Ο κόμβος *y* αρχικά υπολογίζει τις δεσμεύσεις των μεταβλητών, που περιλαμβάνονται στο σχέδιο τριπλετών, βάσει των τριπλετών που ταιριάζουν με το *triplePattern*. Αυτές οι δεσμεύσεις σχηματίζουν μια σχέση *R* που έχει για γνωρίσματα τις ζητούμενες μεταβλητές. Έπειτα, ο κόμβος *y* υπολογίζει την προβολή της *R* στις μεταβλητές που περιλαμβάνονται στην λίστα *variables* και εισάγει τα αποτελέσματα στην σχέση *result*. Τέλος, ο κόμβος *y* στέλνει ένα μήνυμα απάντησης

`GET-RESPONSE(return_address, result, requestid)`

όπου

- *return_address* είναι η διεύθυνση του κόμβου *q*
- *result* είναι η σχέση με τις δεσμευμένες μεταβλητές απάντησης.
- *requestid* είναι το αναγνωριστικό της αίτησης.

Επερωτήσεις σύζευξης

Οι επερωτήσεις σύζευξης αναλύονται όπως συνήθως γίνεται στις σχεσιακές βάσεις δεδομένων. Κάθε τριπλέτα μπορεί να θεωρηθεί μία πλειάδα σε μία σχέση *TRIPLE(S, P, O)* με γνώρισμα *S* για το υποκείμενο, *P* για το κατηγορημα και *O* για το αντικείμενο. Επισημαίνουμε ότι αυτό αποτελεί μια αφαίρεση για να γίνει πιο κατανοητή η περιγραφή του πρωτοκόλλου. Με αυτόν τον τρόπο, οι επερωτήσεις σύζευξης μπορούν να θεωρηθούν επερωτήσεις της μορφής *select-project-join* σε μία βάση δεδομένων που περιέχει μόνο την σχέση *TRIPLE*. Ακολουθεί ο πλήρης αλγόριθμος.

Έστω *x* ένας κόμβος που θέλει να θέσει μια επερωτήση σύζευξης της παρακάτω μορφής:

$?x_1, \dots, ?x_n : (s_1, p_1, o_1) \wedge (s_2, p_2, o_2) \wedge \dots \wedge (s_n, p_n, o_n)$

όπου $?x_1, \dots, ?x_n$ είναι οι μεταβλητές απάντησης και s_i, p_i, o_i είναι το υποκείμενο, το κατηγορημα και το αντικείμενο του κάθε σχεδίου τριπλέτας αντίστοιχα και μπορεί να είναι είτε σταθερές είτε μεταβλητές. Υπενθυμίζουμε ότι τουλάχιστον ένα από αυτά σε κάθε σχέδιο τριπλετών είναι σταθερά.

Ο κόμβος *x* στέλνει το μήνυμα

GETCON-REQUEST(*id*, *triplePattern*, *restTriplePatterns*,
partialResult, *variables*, *return_address*, *requestid*)

στο δίκτυο όπου :

- *id* είναι το αναγνωριστικό που προκύπτει από το κατακερματισμό μίας από τις σταθερές του *triplePattern*.
- *triplePattern* είναι το σχέδιο τριπλετών που θα αποτιμηθεί επόμενο.
- *restTriplePatterns* είναι η λίστα με τα υπόλοιπα σχέδια τριπλετών.
- *partialResult* είναι μια σχέση με τα αποτελέσματα, αρχικά κενή (*nil*).
- *variables* είναι η λίστα με τις μεταβλητές απάντησης $?x_1, \dots, ?x_n$.
- *return_address* είναι η IP διεύθυνση του κόμβου που έθεσε αυτήν την επερώτηση.
- *requestid* είναι το μοναδικό αναγνωριστικό αυτής της συγκεκριμένης αίτησης.

Επομένως, ο κόμβος x έχει δύο επιλογές, να επιλέξει την επόμενη πλειάδα που πρέπει να αποτιμηθεί (*triplePattern*) και μετά να επιλέξει τον κόμβο όπου θα γίνει η αποτίμηση (*id*). Ο τρόπος που θα γίνουν οι παραπάνω επιλογές εξαρτάται από το μέγεθος που θέλει κανείς να βελτιστοποιήσει κατά την επεξεργασία επερωτήσεων και ο τρόπος επιλογής θα συζητηθεί στην Ενότητα 4.3.4.

Το μήνυμα δρομολογείται σύμφωνα με τον αλγόριθμο δρομολόγησης του Bamboo και τελικά καταλήγει στον κόμβο y .

Όταν ο κόμβος x που είναι ο ριζικός κόμβος του *id* λάβει το μήνυμα GETCON-REQUEST κάνει τα ακόλουθα. Αρχικά υπολογίζει τις δεσμεύσεις των μεταβλητών που περιλαμβάνονται στο σχέδιο τριπλετών βρίσκοντας τις τριπλέτες που ταιριάζουν με το *triplePattern*. Αυτές οι δεσμεύσεις σχηματίζουν μία νέα σχέση R με γνωρίσματα τις ζητούμενες μεταβλητές. Αν το *partialResult* είναι *nil*, τότε ο κόμβος y θέτει το R ίσο με *partialResult*. Διαφορετικά, ο y υπολογίζει την φυσική ζεύξη του R και του *partialResult* και θέτει το *partialResult* ίσο με $\text{partialResult} \bowtie R$. Έπειτα, ο y δημιουργεί ένα νέο μήνυμα

GETCON-REQUEST(*id'*, *triplePattern'*, *restTriplePatterns'*,
partialResult', *variables*, *return_address*, *requestid*)

Το μήνυμα το λαμβάνει ο κόμβος z , ο ριζικός κόμβος του τρέχοντος αναγνωριστικού, και ακολουθείται η ίδια διαδικασία. Όλοι οι κόμβοι εκτός του x εφαρμόζουν ζεύξη στην σχέση R των δεσμεύσεων, που ανακτούν τοπικά, με την σχέση *partialResult* και στέλνουν κατάλληλο μήνυμα στον επόμενο κόμβο.

Αυτή η διαδικασία τερματίζει σε δύο περιπτώσεις. Είτε όταν αδειάσει η λίστα *restTriplePatterns* ή όταν γίνει κενή η σχέση *partialResult*. Το τελευταίο υπονοεί ότι το τρέχον σχέδιο τριπλετών δεν ταιριάζει με καμία από τις τριπλέτες που είναι αποθηκευμένες τοπικά στον κόμβο και επομένως για αυτό η σχέση *R* είναι κενή αφού η ζεύξη οδηγεί σε μία κενή σχέση. Σε κάθε περίπτωση, μία απάντηση με τα αποτελέσματα πρέπει να επιστραφεί στον κόμβο *x* που έθεσε την επερώτηση. Το πεδίο *return_address*, που χρησιμοποιείται για αυτό τον σκοπό, παραμένει σταθερό και αναφέρεται καθόλη την διαδικασία στην IP διεύθυνση του κόμβου *x*.

Ο κόμβος *w*, που προσδιορίζει ότι η διαδικασία έχει τελειώσει, υπολογίζει τις δεσμεύσεις των μεταβλητών απάντησης $?x_1, \dots, ?x_n$. Για να το κάνει αυτό ο *w* υπολογίζει την προβολή του *partialResult* βάσει των μεταβλητών που περιλαμβάνονται στην λίστα *variables* και εισάγει τα αποτελέσματα στην σχέση *variableBindings* δηλαδή, $variableBindings = \pi_{variables}(partialResult)$.

Έπειτα, ο *w* στέλνει το μήνυμα απάντησης

GETCON-RESPONSE(*return_address*, *variableBindings*, *requestid*)

στον κόμβο *x* όπου:

- *return_address* είναι η IP διεύθυνση του *x*.
- *variableBindings* είναι η σχέση με την απάντηση στην επερώτηση.
- *requestid* είναι το αναγνωριστικό της αίτησης στην οποία αναφέρεται η απάντηση.

Επερωτήσεις διάζευξης

Όταν ο κόμβος *x* θέτει μια επερώτηση διάζευξης, η επερώτηση αναλύεται τοπικά στο σύνολο των επερωτήσεων σύζευξης από τις οποίες αποτελείται. Έπειτα, ο κόμβος *x* αποτιμάει αυτές τις επερωτήσεις σύζευξης ξεχωριστά, προβάλλει κάθε αποτέλεσμα στις μεταβλητές απάντησης και επιστρέφει την ένωση αυτών των πράξεων προβολής. Εάν κάποια από τις επερωτήσεις είναι ατομική επερώτηση, αυτή η επερώτηση αποτιμάται με την αποστολή ενός μηνύματος GET-REQUEST και με την χρήση του πρωτοκόλλου που περιγράφηκε παραπάνω. Εάν όμως μια επερώτηση είναι μια επερώτηση σύζευξης, ο κόμβος στέλνει ένα μήνυμα GETCON-REQUEST και ακολουθείται το αντίστοιχο πρωτόκολλο.

Για παράδειγμα, εάν ο κόμβος *x* θέσει την επερώτηση

$$?x, ?z : (s_1, p_1, ?x) \vee ((s_2, p_2, ?y) \wedge (?y, p_3, ?z))$$

πρώτα την αναλύει στις ακόλουθες επερωτήσεις

$$?x : (s_1, p_1, ?x) \text{ ανδ } ?z : (s_2, p_2, ?y) \wedge (?y, p_3, ?z).$$

Έπειτα, στέλνει ένα μήνυμα GET-REQUEST στον κόμβο y , που είναι ο ριζικός κόμβος του αναγνωριστικού του s_1 , και ένα άλλο μήνυμα GETCON-REQUEST στον κόμβο z , που είναι ο ριζικός κόμβος του αναγνωριστικού του s_2 . Αφού ολοκληρωθεί η εκτέλεση των πρωτοκόλλων, επιστρέφεται η απάντηση στον κόμβο που έθεσε την επερώτηση.

4.3.4 Πιθανές βελτιστοποιήσεις

Σε αυτήν την ενότητα θα συζητήσουμε τι είδους βελτιστοποιήσεις μπορούν να γίνουν στα πρωτόκολλα επεξεργασίας επερωτήσεων που παρουσιάστηκαν και πως μπορούμε να τις εκμεταλλευτούμε αυτές στο σύστημα Atlas για να βελτιώσουμε την απόδοση του.

Τεχνικές δημιουργίας αντιγράφων. Στην Ενότητα 4.3.2 αναφέραμε ότι υπάρχουν $2 * leaf_set_size$ αντίγραφα κάθε τριπλέτας, όπου $leaf_set_size$ είναι μία ρυθμιστική παράμετρος της υλοποίησης του Bamboo. Αυτό γίνεται για την αύξηση της ανοχής σφαλμάτων (fault-tolerance) επειδή κάποιοι κόμβοι του Atlas μπορεί να αποτύχουν ή κάποιες δικτυακές συνδέσεις να πέσουν. Η δημιουργία αντιγράφων αποτελεί μια κλασσική μέθοδο στα κατανεμημένα συστήματα και ο ρόλος της στο σύστημα Atlas πρέπει να εκτιμηθεί προσεκτικά μαζί με τα υπόλοιπα θέματα που αναφέρονται παρακάτω.

Επεξεργασία ατομικών επερωτήσεων. Η επιλογή του αναγνωριστικού (κλειδιού) που χρησιμοποιείται για την απάντηση μιας ατομικής επερωτήσης μπορεί να έχει δραστική επίδραση στην επίδοση του συστήματος. Όπως συζητήθηκε αρχικά στο άρθρο [19], τα κλειδιά μπορεί να μην κατακερματίζονται ομοιόμορφα στον κυκλικό χώρο αναγνωριστικών του ΚΠΚ επειδή η κατανομή των δεδομένων δεν είναι ομοιόμορφη. Για παράδειγμα, στο άρθρο [19] δείχνουν ότι η μέτρηση της συχνότητας κατανομής των URIs και των κυριολεκτικών σε αρχεία RDF από το Open Directory Project (<http://dmoz.org/>) είναι πολύ ετεροκλινής. Ο χειρισμός ετεροκλινών κατανομών δεδομένων είναι ένα κοινό πρόβλημα στα περισσότερα συστήματα επεξεργασίας επερωτήσεων που έχουν χτιστεί πάνω από ΚΠΚ. Στο Atlas χρησιμοποιούμε μόνο το απλό ευριστικό που παρουσιάσαμε. Για να χειριστούμε ετεροκλινείς κατανομές χρειάζεται να χρησιμοποιηθούν πιο πολύπλοκες μέθοδοι.

Εξισορρόπηση φόρτου. Δεν έχουμε μελετήσει όλες τις δυνατότητες που υπάρχουν για την εξισορρόπηση του φόρτου στο σύστημα Atlas. Αυτό είναι στενά συνδεδεμένο με την περίπτωση των ετεροκλινών κατανομών (γιατί η ετεροκλινής πρόσβαση δημιουργεί ζεστά σημεία (hot spots)) και με τις διαθέσιμες τεχνικές δημιουργίας αντιγράφων.

Επεξεργασία επερωτήσεων βάσει εγγύτητας. Οι πίνακες δρομολόγησης στο Bamboo δημιουργούνται παίρνοντας υπόψη πληροφορία εγγύτητας μεταξύ των κόμβων. Αυτό μας δίνει την δυνατότητα να επεξεργαστούμε τις επερωτήσεις χρησιμοποιώντας αυτήν την πληροφορία εγγύτητας (*proximity-aware*) με στόχο να ελαχιστοποιηθεί το κόστος του δικτύου. Αυτό θα πρέπει να μελετηθεί περισσότερο ως δυνατότητα στο μέλλον.

Συσχέτιση με βελτιστοποίηση σε κατακερματισμένες βάσεις δεδομένων. Η κοινότητα των βάσεων δεδομένων έχει κάνει πολύ δουλειά στην περιοχή των κατακερματισμένων και των παράλληλων βάσεων δεδομένων στο παρελθόν [57, 33]. Η δουλειά πάνω σε αλγόριθμους ζεύξης βασισμένους σε κατακερματισμό για αρχιτεκτονικές παράλληλων και κατακερματισμένων βάσεων δεδομένων σχετίζεται με την εργασία μας [99, 66, 88, 32]. Οι κλασικοί αλγόριθμοι κατακερματισμένης ζεύξης έχουν πρόσφατα ξαναπροκαλέσει το ενδιαφέρον στο πλαίσιο αποτίμησης επερωτήσεων πάνω από συστήματα ΚΠΚ στο πρόγραμμα PIER [49]. Θα θέλαμε να ασχοληθούμε με τις ιδέες από το PIER με περισσότερη λεπτομέρεια και να δούμε κατά πόσο μπορούν να προσφέρουν κάτι χρήσιμο στην υλοποίηση του Atlas.

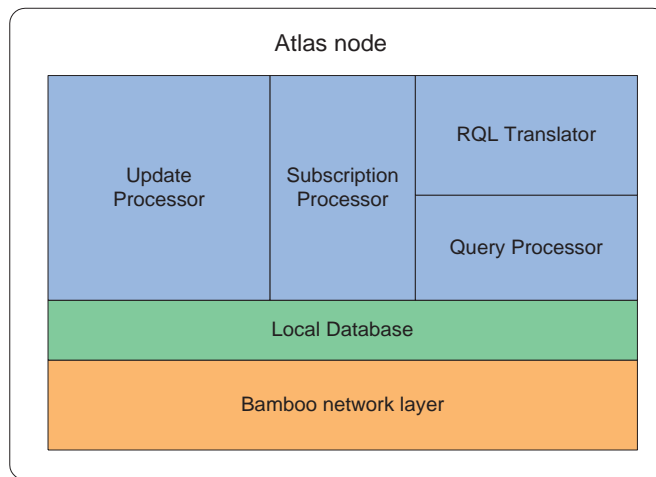
4.4 Λεπτομέρειες υλοποίησης Atlas v0.5

Το Atlas υλοποιήθηκε στην γλώσσα προγραμματισμού Java, την γλώσσα στην οποία είναι γραμμένο και το σύστημα Bamboo. Στην τρέχουσα υλοποίηση χρησιμοποιούμε την CVS έκδοση του Bamboo 01-07-2005⁷.

4.4.1 Αρχιτεκτονική ενός κόμβου του Atlas

Σε αυτήν την ενότητα, θα αναγνωρίσουμε τα στοιχεία τα οποία συνθέτουν ένα κόμβο του Atlas και υποστηρίζουν τα πρωτόκολλα που παρουσιάστηκαν προηγουμένως. Διακρίνουμε τα ακόλουθα έξι στοιχεία σε έναν κόμβο του Atlas: το στρώμα δικτύου του Bamboo που είναι υπεύθυνο για την δρομολόγηση και την διαχείριση των μηνυμάτων του δικτύου, τον επεξεργαστή ανανεώσεων που είναι υπεύθυνος για την αποθήκευση και την ανανέωση RDF δεδομένων, τον επεξεργαστή δημοσίευσης /συνδρομής, τον επεξεργαστή επερωτήσεων που είναι υπεύθυνος για την απάντηση επερωτήσεων των τύπων που έχουν οριστεί στην Ενότητα 4.2, την τοπική βάση δεδομένων που χρησιμοποιείται για την τοπική αποθήκευση δεδομένων σε κάθε κόμβο και τον μεταφραστή RQL που είναι υπεύθυνος για την μετατροπή των RQL επερωτήσεων στους τύπους επερωτήσεων που παρουσιάστηκαν στην Ενότητα 4.2. Θα περιγράψουμε τον ρόλο κάθε στοιχείου και την κατάσταση της υλοποίησης τους στις επόμενες ενότητες.

⁷<http://bamboo-dht.org/bamboo-cvs-2005-07-01.tgz>



Σχήμα 4.4: Αρχιτεκτονική ενός κόμβου του Atlas

Επεξεργαστής ενημερώσεων. Ο επεξεργαστής ενημερώσεων είναι υπεύθυνος για την αποθήκευση ενός RDF εγγράφου στο δίκτυο του Atlas. Λαμβάνει ως είσοδο ένα RDF έγγραφο σε μορφή RDF/XML ή N3, αποσυνθέτει το έγγραφο σε τριπλέτες και αποθηκεύει τις τριπλέτες στο δίκτυο. Επιπλέον, αυτός είναι υπεύθυνος για την αποθήκευση των τριπλέτων που λήφθηκαν μετά από μια αίτηση put-request ενός άλλου κόμβου. Κάθε τριπλέτα αποθηκεύεται στην τοπική βάση δεδομένων του αντίστοιχου κόμβου βάσει του αναγνωριστικού της τριπλέτας. Οι κόμβοι που είναι υπεύθυνοι για την αποθήκευση των τριπλετών προκύπτουν από την χρήση του πρωτοκόλλου που παρουσιάστηκε στην Ενότητα 4.3.2.

Ο επεξεργαστής ενημερώσεων είναι επίσης υπεύθυνος για την ενημέρωση και την διαγραφή τριπλετών που έχουν προηγουμένως αποθηκευτεί στο δίκτυο. Οι ενημερώσεις στο Atlas θα ακολουθήσουν στο μέλλον την σημασιολογία της γλώσσας RUL. Η απαιτούμενη λειτουργικότητα για την διαχείριση της RUL δεν υλοποιείται στο σύστημα Atlas v0.5.

Επεξεργαστής δημοσίευσης / συνδρομής. Ο επεξεργαστής δημοσίευσης / συνδρομής θα είναι υπεύθυνος για την παροχή της λειτουργικότητας δημοσίευσης / συνδρομής στο σύστημα Atlas. Θα αναπτυχθεί σύμφωνα με τις ιδέες που παρουσιάζονται στο άρθρο [61]. Δεν έχει υλοποιηθεί στο Atlas v0.5

Τοπική βάση δεδομένων. Η τοπική βάση δεδομένων είναι το μέρος που κάθε κόμβος αποθηκεύει τα ζεύγη (*key, value*) που του αντιστοιχούν. Στην υλοποίηση του Bamboo, χρησιμοποιείται η βάση δεδομένων Berkeley DB [48]. Το λογισμικό Berkeley DB αποτελείται από μια βιβλιοθήκη βάσεων δεδομένων (λογισμικό ανοικτού κώδικα) που παρέχει ένα απλό API για την προσπέλαση και διαχείριση δεδομένων. Δεν αποτελεί σχεσιακή ή αντικειμενοστραφής βάση δεδομένων (έστω κι εάν κάποιος μπορεί να την χρησιμοποιήσει για να δημιουργήσει μια σχεσι-

ακή βάση δεδομένων) και δεν υποστηρίζει κανένος τύπου γλώσσες επερωτήσεων υψηλού επιπέδου όπως την SQL. Ωστόσο, παρέχει ένα απλό API έτσι ώστε να μπορεί κανείς να αποθηκεύσει και να ανακτήσει δεδομένα με αποδοτικό τρόπο.

Τα δεδομένα σε μια βάση Berkeley DB αποθηκεύονται σε εγγραφές της μορφής ζευγών (*key, value*). Υποστηρίζονται οι ακόλουθες λειτουργίες στις εγγραφές: *εισαγωγή* ζεύγους σε έναν πίνακα, *διαγραφή* ενός ζεύγους από έναν πίνακα, *έυρεση* ενός ζεύγους σε έναν πίνακα με χρήση ενός κλειδιού και *αναέωση* ενός ζεύγους που έχει ήδη βρεθεί. Το Bamboo χρησιμοποιεί μια ασύγχρονη διεπαφή σε Java για επικοινωνία με την βάση δεδομένων Berkeley DB.

Επεξεργαστής επερωτήσεων. Ο επεξεργαστής επερωτήσεων λαμβάνει ως είσοδο τις επερωτήσεις που θέτονται από κάθε κόμβο. Αρχικά διαβάζει και ελέγχει την επερώτηση. Θεωρείται ότι μια επερώτηση είναι σε έγκυρη μορφή εάν είναι σε μια από τις μορφές που ορίστηκαν στην Ενότητα 4.2. Αφού ελεγχθεί ότι η επερώτηση είναι έγκυρη, ο επεξεργαστής επερωτήσεων αποτιμάει την επερώτηση χρησιμοποιώντας τα πρωτόκολλα που παρουσιάστηκαν ήδη λεπτομερώς στην Ενότητα 4.3.3.

Μεταφραστής RQL. Ο μεταφραστής RQL μεταφράζει τις επερωτήσεις RQL στους τύπους των επερωτήσεων που ορίζονται στην Ενότητα 4.2. Έπειτα, αυτές τις επερωτήσεις θα τις διαχειριστεί ο επεξεργαστής επερωτήσεων. Ο μεταφραστής δεν περιλαμβάνεται στην τρέχουσα υλοποίηση του Atlas v0.5.

4.4.2 Atlas API

Σε αυτήν την ενότητα, παρουσιάζουμε το API που υποστηρίζεται από το Atlas v0.5. Αφού παρουσιάσουμε το API αναγνωρίζουμε ένα API χαμηλού επιπέδου που προσφέρεται από το σύστημα Bamboo και τις επεκτάσεις του που είναι απαραίτητες για την υλοποίηση του συστήματος Atlas.

Το Atlas API είναι το ακόλουθο:

```
package org.ontogrid.ontokit.MetadataService.Atlas;  
UpdateProcessor.StoreMetadata(RDFDocument);  
QueryProcessor.QueryMetadata(Query);
```

Η λειτουργία `StoreMetadata` αποθηκεύει δεδομένα στο σύστημα Atlas. Το έγγραφο προς αποθήκευση μπορεί να είναι σε μορφή N3 (τριπλέτες) ή σε μορφή RDF/XML. Η λειτουργία επιστρέφει κατά πόσο η αποθήκευση ολοκληρώθηκε με επιτυχία. Υλοποιείται χρησιμοποιώντας το πρωτόκολλο για την αποθήκευση δεδομένων που περιγράφηκε στην Ενότητα 4.3.2. Η λειτουργία `QueryMetadata` παρέχει την λειτουργικότητα της επεξεργασίας απλών επερωτήσεων. Η επερώτηση πρέπει να είναι εκφρασμένη στην γλώσσα επερωτήσεων που παρουσιάστηκε στην

Ενότητα 4.2. Ο επεξεργαστής επερωτήσεων αποτιμάει την επερώτηση ακολουθώντας τα αντίστοιχα πρωτόκολλα που έχουν ήδη περιγραφεί. Έπειτα, λαμβάνει τις απαντήσεις στην επερώτηση σαν μια λίστα από δεσμευμένες μεταβλητές.

Το παραπάνω API θα προσφέρεται από κάθε κόμβο του δικτύου του Atlas. Σε ένα περιβάλλον Πλέγματος είναι δυνατόν κάθε συμμετέχοντας κόμβος να αποτελεί έναν κόμβο του Atlas. Εναλλακτικά, το Atlas μπορεί να υλοποιηθεί από ένα σύνολο *αφοσιωμένων* κόμβων. Αυτό θα ήταν μια έξυπνη λύση για να αποφευχθεί η υπερφόρτωση των κόμβων του Πλέγματος που θα έχουν να εκτελέσουν και άλλες εργασίες. Το ιδανικό μέγεθος ενός δικτύου του Atlas μπορεί να επιλεγθεί λαμβάνοντας υπόψη τις σχέσεις μεταξύ των μεταβλητών συστήματος του ΚΠΚ και το μέγεθος της δραστηριότητας που θα προκύψει. Εάν η απόδοση γίνει πρόβλημα (π.χ. όλοι οι κόμβοι υπερφορτωθούν), μπορούμε πάντα να προσθέσουμε νέους κόμβους στο δίκτυο για να διορθώσουμε την κατάσταση. Εάν οι κόμβοι του Atlas αποτύχουν ή δεν είναι προσωρινά διαθέσιμοι τα πρωτόκολλα του ΚΠΚ εγγυώνται ακριβώς για την ποιότητα των απαντήσεων που θα επιστραφούν.

Αλλαγές στο API του Bamboo. Θα περιγράψουμε τις λειτουργίες χαμηλού επιπέδου που υποστηρίζονται από το Bamboo, καθώς και κάποιες λειτουργίες που έχουν επεκταθεί έτσι ώστε να υποστηριχθούν πλήρως τα πρωτόκολλα του Atlas. Όπως έχει ήδη αναφερθεί προηγουμένως στην Ενότητα 2.4, οι βασικές λειτουργίες που παρέχονται από ένα δίκτυο ΚΠΚ είναι οι λειτουργίες `put` και `get`. Αυτές οι λειτουργίες υλοποιούνται πάνω από τις ακόλουθες λειτουργίες χαμηλού επιπέδου.

Γενικά, μια λειτουργία `put` αποθηκεύει ένα ζεύγος *αναγνωριστικό-τιμή* στο δίκτυο. Η αντίστοιχη λειτουργία που παρέχεται από το Bamboo είναι:

```
PUT(id, value, tll_sec)
```

όπου *id* είναι το αναγνωριστικό, *value* είναι η τιμή του αναγνωριστικού και *tll_sec* είναι ο χρόνος για τον οποίο η τιμή αυτή θα θεωρείται έγκυρη στο δίκτυο. Όλες οι λειτουργίες `PUT` εφαρμόζονται σαν λειτουργίες πρόσθεσης (`append`). Επομένως, δύο `PUT` λειτουργίες με το ίδιο αναγνωριστικό αλλά με διαφορετικές τιμές θα έχουν ως αποτέλεσμα την αποθήκευση και των δύο τιμών στο δίκτυο. Όλα τα μεταδεδομένα που εισάγονται με τις λειτουργίες `put` διατηρούνται στο δίκτυο για τόση χρονική διάρκεια όση ορίζεται στο πεδίο *tll_sec*. Για τους σκοπούς του *Atlas* και την τρέχουσα υλοποίηση του συστήματος, για την αποθήκευση εγγράφων RDF, η λειτουργία `PUT` όπως προσφέρεται από το Bamboo κρίνεται επαρκής.

Μια λειτουργία `GET` ανακτάει μια τιμή βάσει ενός αναγνωριστικού. Το Bamboo προσφέρει την λειτουργία:

```
GET(id, maxvals, placemark)
```

όπου *id* είναι το αναγνωριστικό, *maxvals* είναι το μέγιστο πλήθος των τιμών που θα ανακτηθούν και *placemark* είναι μια παράμετρος για την ανάκτηση περαιτέρ-

Λειτουργία	Κατάσταση
Αποθήκευση RDF εγγράφων	Ολοκληρώθηκε
Επερωτήσεις σύζευξης	Ολοκληρώθηκε
Επερωτήσεις διάζευξης	Μελλοντική εργασία
Υποστήριξη RQL και RUL	Μελλοντική εργασία
Υποστήριξη RDFS	Μελλοντική εργασία
Επερωτήσεις με περιορισμούς για επιλεγμένους τύπους δεδομένων	Μελλοντική εργασία
Δημοσίευση / Συνδρομή	Μελλοντική εργασία

Πίνακας 4.1: Κατάσταση της υλοποίησης του Atlas

ω τιμών. Η λειτουργία GET που παρέχεται από το Bamboo δεν αρκεί για την υλοποίηση της λειτουργίας απάντησης επερωτήσεων του συστήματος Bamboo.

Για αυτό το λόγο έχει επεκταθεί με τις ακόλουθες λειτουργίες

`GET(id, maxvals, triple_pattern, placemark)`

όπου *triple_pattern* είναι η επερώτηση που πρέπει να απαντηθεί μέσω της λειτουργίας GET. Περιλαμβάνοντας την επερώτηση στις παραμέτρους της GET, μειώνουμε τον φόρτο του δικτύου γιατί θα επιστραφούν πίσω στον κόμβο που έθεσε την επερώτηση μόνο οι τιμές που απαντάνε την επερώτηση. Εάν δεν περιλαμβανόταν το *triple_pattern*, όλες οι τιμές που είναι συσχετισμένες με το κλειδί θα επιστρέφονταν στον κόμβο και έπειτα ο κόμβος τοπικά θα έπρεπε να βρει το υποσύνολο των τιμών που ταιριάζουν με την επερώτηση.

Επιπλέον, για να υποστηρίζονται και επερωτήσεις σύζευξης, έχει δημιουργηθεί μια νέα λειτουργία βασισμένη στις ιδέες που παρουσιάστηκαν στην Ενότητα 4.3.3:

`GETCON(id, maxvals, triple_pattern, rest_triple_patterns, partialResult, placemark)`

Περισσότερες λεπτομέρειες για την χρήση των παραμέτρων που περιλαμβάνονται στις λειτουργίες GET και GETCON αναφέρονται στην προηγούμενη περιγραφή των πρωτοκόλλων που χρησιμοποιούνται για την επεξεργασία επερωτήσεων στην Ενότητα 4.3.3.

4.5 Κατάσταση υλοποίησης

Στον πίνακα 4.1, παρουσιάζουμε περιληπτικά τις λειτουργίες που έχουν ήδη υλοποιηθεί στο Atlas v0.5.

Στην υπόλοιπη ενότητα αναφερόμαστε στις λειτουργίες που δεν έχουν υλοποιηθεί ακόμη και συζητάμε ιδέες σχετικά με το πως θα μπορούσε να γίνει η υλοποίησή τους.

4.5.1 Επερωτήσεις με περιορισμούς στις μεταβλητές

Σε μια μελλοντική έκδοση του Atlas, στοχεύουμε να υποστηρίξουμε επερωτήσεις σύζευξης με περιορισμούς στις μεταβλητές που περιλαμβάνουν. Οι μεταβλητές μπορεί να έχουν ως τιμές URIs ή κυριολεκτικά. Στο μοντέλο του RDF, τα *απλά κυριολεκτικά* (plain literals) δεν παρέχουν πληροφορία για τον τύπο δεδομένων τους. Για παράδειγμα, το απλό κυριολεκτικό 24 δεν θεωρείται αριθμός αλλά συμβολοσειρά με δύο χαρακτήρες, τους 2 και 4. Τα *κυριολεκτικά με τύπο* (typed literals) είναι αυτά που χρησιμοποιούνται για να συσχετίσουν ένα κυριολεκτικό με έναν τύπο δεδομένων. Τα κυριολεκτικά με τύπο δημιουργούνται με τον συνδυασμό μιας συμβολοσειράς με ένα URI που αναγνωρίζει έναν συγκεκριμένο τύπο δεδομένων. Με αυτόν τον τρόπο η έκφραση

```
"27"^^<http://www.w3.org/2001/XMLSchema#integer>
```

φανερώνει έναν ακέραιο. Βάσει των κυριολεκτικών με τύπο του RDF, αρχικά στοχεύουμε να υποστηρίξουμε περιορισμούς στους ακόλουθους τύπους δεδομένων από το XML Schema:

- ακέραιος (`xsd:integer`)
- μεγάλος ακέραιος (`xsd:float`)
- συμβολοσειρά (`xsd:string`)

Για τους δύο πρώτους τύπους δεδομένων, τους ακεραίους, θα υποστηρίξουμε επερωτήσεις εύρους για να υπάρχει δυνατότητα περιορισμού του εύρους των μεταβλητών σε μια επερώτηση σύζευξης. Γενικά, μια επερώτηση σύζευξης με περιορισμούς έχει την ακόλουθη μορφή:

$$\phi \wedge (?x_i \geq l_i \wedge ?x_i \leq u_i) \wedge \dots \wedge (?x_m \geq l_m \wedge ?x_m \leq u_m)$$

όπου ϕ είναι μια επερώτηση σύζευξης, x_i, \dots, x_m είναι μεταβλητές και $l_i, \dots, l_m, u_i, \dots, u_m$ είναι τα κάτω και άνω όρια τους εύρους της επερώτησης.

Χρησιμοποιώντας το σχήμα του Σχήματος 4.2, θεωρούμε το παρακάτω παράδειγμα επερώτησης εύρους:

- Βρες υπηρεσίες που προσφέρονται από τον οργανισμό Fiat και κοστίζουν από 50 έως 100 ευρώ.

$$?x : (?x, hasServiceOrganization, "Fiat") \wedge$$
$$(?x, hasServiceCost, ?cost) \wedge (?cost \geq 50 \wedge ?cost \leq 100)$$

Για μεταβλητές τύπου συμβολοσειράς, η πρώτη απαίτηση είναι να υποστηριχθούν περιορισμοί ισότητας. Μια τέτοια επερώτηση έχει την ακόλουθη μορφή:

$$q \wedge (?x_i = c_i) \wedge \dots \wedge (?x_m = c_m)$$

όπου c_i, \dots, c_m είναι σταθερές συμβολοσειρές που περιορίζουν το εύρος των μεταβλητών x_i, \dots, x_m που εμφανίζονται στην q . Ακολουθεί ένα απλό παράδειγμα:

- Βρες τις υπηρεσίες που παρέχονται από τον οργανισμό Fiat και έχουν ως συγγραφέα τον John Smith.

$$?x : (?x, hasServiceOrganization, "Fiat") \wedge (?x, hasServiceAuthor, ?author) \wedge (?author = "JohnSmith")$$

Είναι επίσης σημαντικό να υποστηρίζονται οι περιορισμοί σε συμβολοσειρές με χρήση του τελεστή *LIKE* της RQL για ταίριασμα συμβολοσειρών χρησιμοποιώντας wildcards. Για παράδειγμα, ας θεωρήσουμε την ακόλουθη επερώτηση:

- Βρες τις υπηρεσίες που έχουν όνομα που περιλαμβάνει την λέξη Travel.

$$?x : (?x, hasServiceName, ?name) \wedge (?name \text{ LIKE } "*Travel*")$$

Η παρουσία των τύπων δεδομένων και των περιορισμών στην γλώσσα επερωτήσεων κάθε κεντροποιημένου ή κατακερματισμένου συστήματος βάσεων δεδομένων, όπως στο σύστημα Atlas, απαιτεί την χρήση κατάλληλων *ευρητηρίων*. Ο μόνος τύπος ευρητηρίων στο Atlas είναι αυτό που προκύπτει μέσω του κατακερματισμένου κατακερματισμού που προσφέρεται από το Bamboo. Ο κατακερματισμός είναι γνωστός για τις εξαιρετικές επιδόσεις του για επερωτήσεις με περιορισμούς ισότητας. Το πρόβλημα με τις υπόλοιπες επερωτήσεις (π.χ. περιορισμοί εύρους, ταίριασμα συμβολοσειρών) δεν έχει μελετηθεί με λεπτομέρεια στο Atlas v0.5. Για να ενημερωθούμε για το τι μπορεί να γίνει, πρώτο βήμα αποτελεί η ανασκόπηση των λίγων σχετικών εργασιών στην περιοχή (βλέπε [19, 72, 30])

4.5.2 RDFS

Ο χειρισμός πληροφορίας RDFS στο σύστημα Atlas αποτελεί μια από τις άμεσες προτεραιότητες μας έτσι να ώστε να υποστηρίξουμε την συλλογιστική RDFS για τάξεις και στιγμιότυπα με αποδοτικό τρόπο. Αυτό το πρόβλημα παραμένει ανοικτό στην περιοχή των ΚΠΚ.

4.5.3 Υποστήριξη για RQL και RUL

Μία από τις βασικές προτεραιότητες μας στο σύστημα Atlas είναι η υποστήριξη των γλωσσών RQL και RUL. Αυτό το πρόβλημα περιλαμβάνει (στην περίπτωση της RQL) την ανάπτυξη κατάλληλων μεταφρασιών στην τρέχουσα γλώσσα επερωτήσεων του Atlas, η οποία βασίζεται στις τριπλέτες.

4.5.4 Δημοσίευση / Συνδρομή

Πρόσφατα, στο άρθρο [61], οι Λιάρου κ.ά. ασχοληθήκανε με το πρόβλημα της δημοσίευσης / συνδρομής σε δίκτυα ομότιμων κόμβων βασισμένα σε σχήματα RDF για μια υποκλάση των συζευκτικών επερωτήσεων (όπως ορίζεται στην Ενότητα 4.2.1). Τα αποτελέσματα της πειραματικής ανάλυσης που παρουσιάζονται στο [61] θα βοηθήσουν στην υλοποίηση των λειτουργιών δημοσίευσης / συνδρομής στο Atlas.

4.6 Συμπεράσματα

Σε αυτό το κεφάλαιο παρουσιάσαμε και περιγράψαμε το σύστημα Atlas. Το Atlas είναι ένα σύστημα ομότιμων κόμβων βασισμένο στο ΚΠΚ Bamboo με δυνατότητα αποθήκευσης και επεξεργασίας επερωτήσεων δεδομένων RDF. Αρχικά, παρουσιάσαμε την αρχιτεκτονική του Atlas και έπειτα ορίσαμε το μοντέλο δεδομένων και την γλώσσα επερωτήσεων που χρησιμοποιεί έτσι ώστε να ορίσουμε τυπικά τους τύπους επερωτήσεων που μπορεί να επεξεργαστεί το σύστημα. Επιπλέον, περιγράψαμε λεπτομερώς τα πρωτόκολλα επεξεργασίας επερωτήσεων της τρέχουσας υλοποίησης του Atlas v0.5. Τέλος, παρουσιάσαμε την κατάσταση της τρέχουσας υλοποίησης, τις πιθανές βελτιστοποιήσεις και τα σχέδια μας για μελλοντική εργασία.

Κεφάλαιο 5

Ο ρόλος του Atlas στο πρόγραμμα OntoGrid

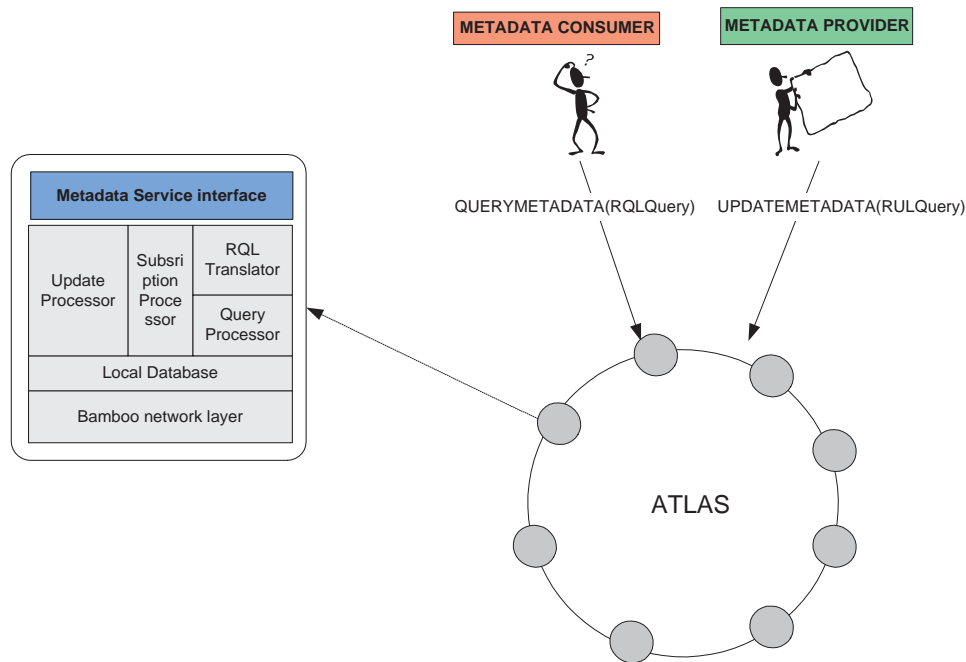
Σε αυτό το κεφάλαιο θα συζητήσουμε τον ρόλο του Atlas στο πρόγραμμα OntoGrid¹. Το σύστημα Atlas θα χρησιμοποιηθεί για να υλοποιήσει την *υπηρεσία μεταδεδομένων* (metadata service) όπως ορίζεται στο Παραδοτέο 1.2 του OntoGrid [58]. Άλλες αντίστοιχες υπηρεσίες είναι η *υπηρεσία επιστημείωσης* (annotation service) και η *υπηρεσία διαπραγματεύσεων* (negotiation service). Η υπηρεσία μεταδεδομένων χρησιμοποιεί το Atlas σαν μια *κατανεμημένη βάση δεδομένων RDF* για την αποθήκευση, την υποβολή επερωτήσεων και την ανανέωση μεταδεδομένων που δημιουργούνται από άλλες σημασιολογικές υπηρεσίες του Πλέγματος.

Στις ενότητες που ακολουθούν, αρχικά θα συζητήσουμε πως μπορεί η υπηρεσία μεταδεδομένων να χρησιμοποιήσει το Atlas για να αποθηκεύσει, να θέσει επερωτήσεις και να ανανεώσει μεταδεδομένα εκφρασμένα σε RDF. Έπειτα θα παρουσιάσουμε ένα συγκεκριμένο σενάριο όπου το σύστημα Atlas μπορεί να χρησιμοποιηθεί για την ανακάλυψη υπηρεσιών. Σε αυτή την περίπτωση το Atlas χρησιμοποιείται για την αποθήκευση και την υποβολή επερωτήσεων περιγραφών υπηρεσιών εκφρασμένες σε μια συγκεκριμένη οντολογία υπηρεσιών. Όπως θα δείξουμε οι γλώσσες επερωτήσεων και ανανέωσης (RQL και RUL) που θα χρησιμοποιηθούν στο Atlas καθιστούν πολύ εύκολη την υλοποίηση της υπηρεσίας μεταδεδομένων. Τα σενάρια προκύπτουν από τις ιδέες που έχουν παρουσιαστεί στο Παραδοτέο 1.2 του OntoGrid [58].

5.1 Το Atlas και η Υπηρεσία Μεταδεδομένων

Η υπηρεσία μεταδεδομένων σχετίζεται με το Atlas με τον τρόπο που απεικονίζεται στο Σχήμα 5.1. Όπως φαίνεται στο σχήμα, η υπηρεσία μεταδεδομένων είναι υπεύθυνη για την χρήση των λειτουργιών που παρέχονται από το σύστημα Atlas για να παρέχει την επιθυμητή λειτουργικότητα. Δύο είδη χρηστών χρησιμοποιούν

¹OntoGrid Project: <http://www.ontogrid.net>



Σχήμα 5.1: Υπηρεσία μεταδεδομένων

την υπηρεσία μεταδεδομένων: οι *καταναλωτές μεταδεδομένων* (metadata consumers) και οι *παροχείς μεταδεδομένων* (metadata providers). Οι παροχείς μεταδεδομένων παράγουν μεταδεδομένα και τα αποθηκεύουν χρησιμοποιώντας την υπηρεσία μεταδεδομένων και οι καταναλωτές μεταδεδομένων χρησιμοποιούν αυτά τα μεταδεδομένα για να συνεχίσουν τις δραστηριότητες τους. Στην αρχιτεκτονική του OntoGrid, για παράδειγμα η υπηρεσία επισημείωσης θα παίζει και τους δύο ρόλους.

5.2 Αποθήκευση, Αποτίμηση Επερωτήσεων και Ενημέρωση Μεταδεδομένων

Για να μπορέσει ένας παροχέας μεταδεδομένων να αποθηκεύσει μεταδεδομένα για έναν πόρο στο σύστημα Atlas πρέπει πρώτα να γίνει γνωστή η οντολογία που θα χρησιμοποιήσει. Υπάρχουν δύο εναλλακτικοί τρόποι να το επιτύχουμε αυτό, είτε ο παροχέας μεταδεδομένων παρέχει την οντολογία ή η υπηρεσία μεταδεδομένων μπορεί να την αποκτήσει επικοινωνώντας με κάποιον που μπορεί να την παρέχει. Εμείς θα θεωρήσουμε τον πρώτο και απλούστερο τρόπο. Η υπηρεσία μεταδεδομένων λοιπόν, παρέχει την λειτουργία *StoreOntology*, που παίρνει ως είσοδο δύο παραμέτρους: το RDFS σχήμα, που αποτελεί την οντολογία και τον

χώρο ονομάτων (namespace), που αποτελεί τον χώρο ονομάτων που ο παροχέας των μεταδεδομένων θέλει να ορίσει στην οντολογία έτσι ώστε να μπορεί να την χρησιμοποιήσει για να θέσει επερωτήσεις. Αντίστοιχη είναι και λειτουργικότητα που παρέχεται από κεντριοποιημένες βάσεις δεδομένων RDF, π.χ. το RSSDB API από το RDFSuite².

Για πόρους, όπως οι υπηρεσίες, υπάρχουν κάποιες γνωστές οντολογίες που μπορούν να χρησιμοποιηθούν. Θεωρούμε ότι αυτές θα γίνονται γνωστές στο σύστημα Atlas κατά την εκκίνηση του.

Το γεγονός ότι το σύστημα Atlas χρειάζεται να γνωρίζει την ακριβή οντολογία των μεταδεδομένων που αποθηκεύει είναι βασικό με τον ίδιο τρόπο που είναι και στα συστήματα σχεσιακών βάσεων δεδομένων [3]. Η γνώση του σχήματος στο Atlas θα χρησιμοποιηθεί για την επιλογή κατάλληλων αποθηκευτικών δομών, για την δημιουργία πλάνων επεξεργασίας επερωτήσεων, για τον έλεγχο της εγκυρότητας των ανανεώσεων κ.ο.κ. Μετά την αποθήκευση των μεταδεδομένων είναι απαραίτητο να μπορούμε να τα ανανεώσουμε ή να τα διαγράψουμε. Όλες αυτές οι λειτουργίες μπορούν να εκφραστούν χρησιμοποιώντας την γλώσσα ανανεώσεων RUL [63]. Η αντίστοιχη λειτουργία που παρέχεται από την υπηρεσία μεταδεδομένων για αυτόν τον σκοπό είναι η UpdateMetadata.

Μια άλλη βασική λειτουργικότητα που παρέχεται από την υπηρεσία μεταδεδομένων είναι η επεξεργασία επερωτήσεων. Ένας χρήστης μπορεί να θέσει μια επερώτηση εκφρασμένη στην γλώσσα RQL [52] χρησιμοποιώντας την λειτουργία QueryMetadata.

Όλες οι λειτουργίες που αναφέρθηκαν παραπάνω περιλαμβάνονται στο ακόλουθο API:

```
package org.ontogrid.ontokit.MetadataService;
```

```
StoreOntology(RDFSschema, namespace);
```

```
UpdateMetadata(RULquery);
```

```
QueryMetadata(RQLquery);
```

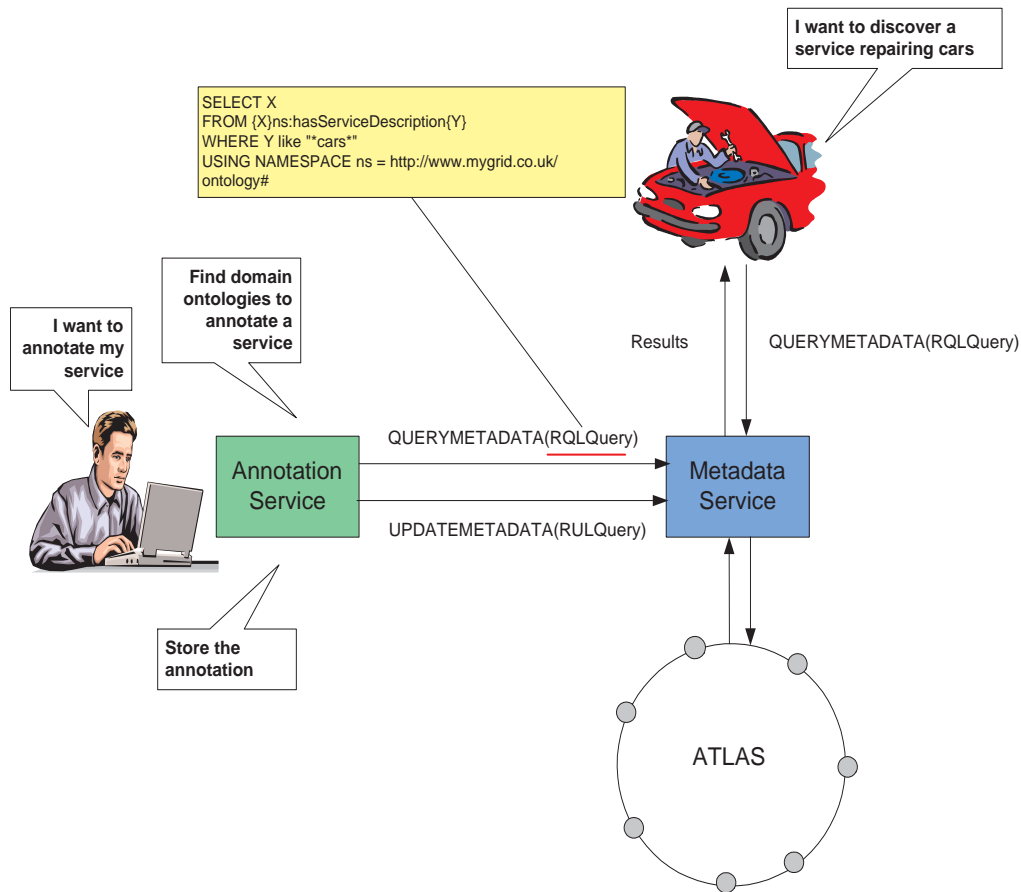
Το παραπάνω API μπορεί να υλοποιηθεί πάνω από το API του Atlas που παρουσιάστηκε στην Ενότητα 4.4.2.

Το API της υπηρεσίας μεταδεδομένων θα παρέχεται από κάθε κόμβο στο δίκτυο του Atlas.

Με αυτόν τον τρόπο θα έχουμε μια καταμεμημένη υλοποίηση της υπηρεσίας μεταδεδομένων με όλα τα αντίστοιχα πλεονεκτήματα που παρέχονται από τα ΚΠΚ: υψηλή απόδοση, ευρωστία και προσαρμοστικότητα.

Στην αρχική έκδοση της υπηρεσίας μεταδεδομένων κάθε κόμβος του συστήματος Atlas θα υλοποιεί το παραπάνω API με μια απλή μετάφραση στο API του Atlas που παρουσιάστηκε στην Ενότητα 4.4.2. Αργότερα, η υλοποίηση του API

²<http://139.91.183.30:9090/RDF/RSSDB/>

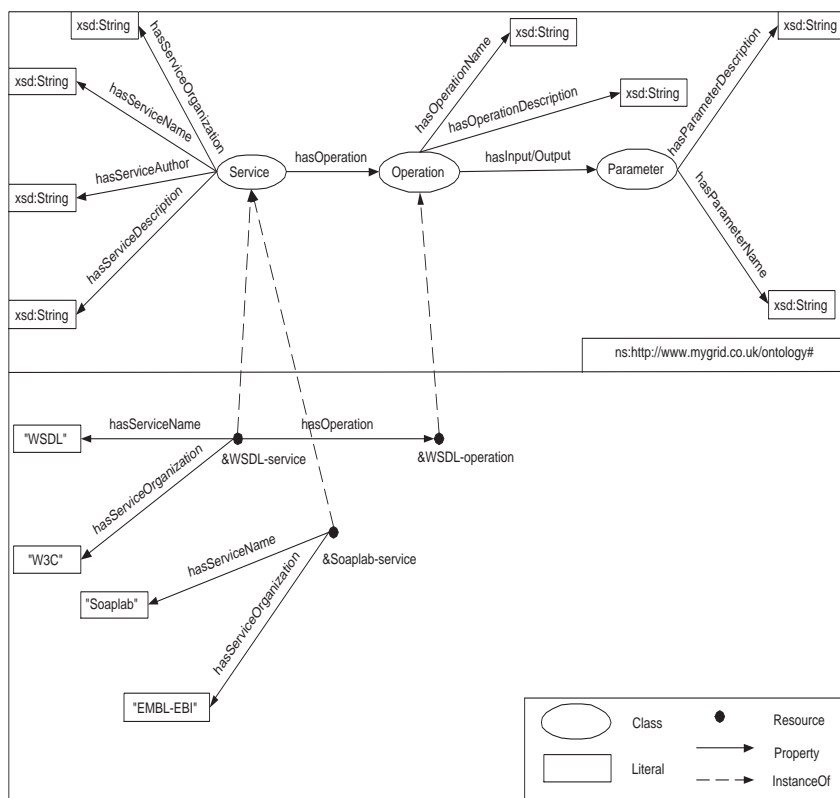


Σχήμα 5.2: Χρησιμοποιώντας το Atlas για ανακάλυψη υπηρεσιών

της υπηρεσίας μεταδεδομένων μπορεί να γίνει περισσότερο πολύπλοκη αναλόγως της λειτουργικότητας που μπορεί να ζητηθεί από αυτήν. Για παράδειγμα, εάν χρησιμοποιηθούν πιο εξελιγμένες γλώσσες οντολογιών (OWL ή άλλες πιο εκφραστικές λογικές σε σχέση με τις RDF/RDFS), η υπηρεσία μεταδεδομένων μπορεί να χρειάζεται να επικοινωνεί με άλλους ώστε να μπορέσει να ολοκληρώσει την εργασία της.

5.3 Σενάρια

Στην ακόλουθη ενότητα θα συζητήσουμε πως μπορεί να χρησιμοποιηθεί το σύστημα Atlas για την ανακάλυψη υπηρεσιών (service discovery).



Σχήμα 5.3: Σχήμα RDFS για υπηρεσίες παγκόσμιου ιστού

5.3.1 Χρησιμοποιώντας το Atlas για ανακάλυψη υπηρεσιών

Σε αυτήν την ενότητα θα δείξουμε πως μπορεί να χρησιμοποιηθεί το σύστημα Atlas κατά την διαδικασία της επισημείωσης υπηρεσιών και της ανακάλυψης. Ολόκληρο το σενάριο απεικονίζεται στο Σχήμα 5.2. Μια οντολογία για υπηρεσίες και ένα σύνολο οντολογιών που αναφέρονται σε συγκεκριμένα πεδία (domain ontologies) είναι απαραίτητα για την δημιουργία της επισημείωσης.

Ας υποθέσουμε ότι η υπηρεσία επισημείωσης θέλει να βρει μια οντολογία για αυτοκίνητα έτσι ώστε να επισημειώσει την υπηρεσία. Μπορεί να θέσει μια RQL επερώτηση στην υπηρεσία μεταδεδομένων και να του επιστραφεί μια κατάλληλη οντολογία για την επισημείωση της υπηρεσίας, π.χ. μια οντολογία για αυτοκίνητα.

Το αποτέλεσμα της επισημείωσης της υπηρεσίας θα συμπεριληφθεί σε ένα έγγραφο RDF το οποίο θα αποθηκευτεί στο σύστημα Atlas με την κλήση της λειτουργίας StoreMetadata.

Η οντολογία που περιγράφει την υπηρεσία πρέπει να έχει προηγουμένως αποθηκευτεί στο σύστημα Atlas με την κλήση της λειτουργίας StoreOntology.

Αφού αποθηκευτούν τα μεταδεδομένα της υπηρεσίας είναι απαραίτητο να μπορεί κάποιος να τα ανανεώσει ή να τα διαγράψει. Αυτές οι λειτουργίες θα εκφραστούν με την γλώσσα ανανεώσεων RUL. Για παράδειγμα, μια υπηρεσία

μπορεί να αλλάξει όνομα. Στο ακόλουθο παράδειγμα μια υπηρεσία με όνομα "car-service" μετονομάζεται σε "car-repair-service".

```
MODIFY ns:hasServiceName(X, Y <- "car-repair-service") FROM  
{X}ns:hasServiceName{Y} WHERE Y = "car-service" USING NAMESPACE ns  
= http://www.mygrid.co.uk/ontology#
```

Ας υποθέσουμε ότι κάποιος χρήστης ή μια υπηρεσία θέλει να ανακαλύψει μια υπηρεσία επισκευής αυτοκινήτων. Αυτό επιτυγχάνεται με το να θέσει επερωτήσεις χρησιμοποιώντας την οντολογία που περιγράφει υπηρεσίες. Δείχνουμε παραδείγματα τέτοιων επερωτήσεων χρησιμοποιώντας την γλώσσα RQL. Η ακόλουθη επερώτηση RQL περιγράφει μια επερώτηση όπου η ζητούμενη υπηρεσία έχει περιγραφή που περιέχει την συμβολοσειρά "cars".

```
SELECT X FROM {X}ns:hasServiceDescription{Y} WHERE Y like "*cars*" USING NAMESPACE ns = http://www.mygrid.co.uk/ontology#
```

Μια άλλη πιθανή επερώτηση είναι να αναζητήσει μια υπηρεσία σύμφωνα με κάποια παράμετρο εισόδου, π.χ. βρες μια υπηρεσία της οποίας κάποια παράμετρο εισόδου περιέχει την συμβολοσειρά "cars".

```
SELECT X  
FROM{X;ns:Service}ns:hasOperation.ns:hasInput/Output{Y},  
{Y}ns:hasParameterDescription{Z}  
WHERE Z like "*cars*" USING NAMESPACE ns =  
http://www.mygrid.co.uk/ontology#
```

5.4 Άλλα θέματα

Δεν έχουμε ασχοληθεί με θέματα όπως θέματα ιδιοκτησίας και εξουσιοδότησης στο σύστημα Atlas ή στην υπηρεσία μεταδεδομένων.

5.5 Συμπεράσματα

Σε αυτό το κεφάλαιο, συζητήσαμε τον ρόλο του Atlas σαν μέρος του προγράμματος OntoGrid. Αρχικά, παρουσιάσαμε πως η υπηρεσία μεταδεδομένων σχετίζεται με το σύστημα Atlas. Έπειτα, αναλύσαμε πως ένας χρήστης μπορεί να αποθηκεύσει, να θέσει επερωτήσεις και να ανανεώσει μεταδεδομένα στο Atlas χρησιμοποιώντας την υπηρεσία μεταδεδομένων. Τέλος, παρουσιάσαμε ένα σενάριο για το πως η υπηρεσία μεταδεδομένων μπορεί να χρησιμοποιηθεί για την αποθήκευση, την υποβολή επερωτήσεων και την ανανέωση μεταδεδομένων υπηρεσιών.

Κεφάλαιο 6

Επίλογος

6.1 Σύνοψη και Συμπεράσματα

Ξεκινήσαμε την παρουσίαση μας, περιγράφοντας το απαραίτητο θεωρητικό υπόβαθρο για την κατανόηση του αντικειμένου αυτής της διπλωματικής. Συγκεκριμένα, αναφερθήκαμε στις τεχνολογίες των δικτύων ομότιμων κόμβων και στο πλαίσιο RDF. Έπειτα, παρουσιάσαμε τις σχετικές εργασίες έτσι ώστε να δημιουργηθεί το ανάλογο υπόβαθρο και να μπορέσει ο αναγνώστης να κατανοήσει την συμβολή του Atlas. Παρουσιάσαμε το σύστημα Atlas, το οποίο χρησιμοποιείται για αποθήκευση RDF δεδομένων και την αποτίμηση επερωτήσεων σε αυτά. Τέλος, αναφερθήκαμε στο πως χρησιμοποιείται αυτό το σύστημα στα πλαίσια του προγράμματος OntoGrid για την υλοποίηση της υπηρεσίας μεταδεδομένων.

6.2 Μελλοντική Εργασία

Αυτή η διπλωματική εργασία μπορεί να επεκταθεί με διάφορους τρόπους. Συγκεκριμένα, όσον αφορά το σύστημα Atlas, μπορούν να γίνουν οι βελτιστοποιήσεις που αναφέρθηκαν καθώς και μια μελέτη της απόδοσης του συστήματος. Οι επεκτάσεις που ξεχωρίζουν είναι η υποστήριξη πληροφορίας σχήματος RDF(S) καθώς και των γλωσσών RQL και RUL.

Ορολογία

Ξενογλωσσος όρος	:	Ελληνικός όρος
Grid	:	Πλέγμα
peer-to-peer network	:	δίκτυο ομότιμων κόμβων
Distributed hash tables	:	Κατανεμημένοι Πίνακες Κατακερματισμού
peer	:	κόμβος
super-peer network	:	Ιεραρχικό δίκτυο ομότιμων κόμβων
peer-to-peer network	:	δίκτυο ομότιμων κόμβων
consistent hashing	:	συνεπής κατακερματισμός
successor node	:	διάδοχος κόμβος
predecessor node	:	προκάτοχος κόμβος
finger table	:	πίνακας δεικτών
leaf set	:	σύνολο φύλλων
routing table	:	πίνακας δρομολόγησης
churn	:	συνεχής διαδικασία αφίξεων-αναχωρήσεων
Semantic Web	:	Σημασιολογικός Ιστός
Web Services	:	υπηρεσίες του παγκόσμιου ιστού
publish / subscribe	:	δημοσίευση / συνδρομή
range queries	:	επερωτήσεις εύρους

Συντμήσεις - Αρκτικόλεξα

Σύντμηση	:	Πλήρης Ανάπτυξη
KPIK	:	Κατακερματισμένοι Πίνακες Κατακερματισμού
RDF	:	Resource Description Framework
RDFS	:	Resource Description Framework Schema
XML	:	Extensible Markup Language
W3C	:	WorldWide Web Consortium
SQL	:	Structured Query Language
WSDL	:	Web Services Description Language
UDDI	:	Universal Description, Discovery and Integration
WSMO	:	Web Services Modelling Ontology
DAML	:	DARPA Agent Markup Language
OWL	:	Web Ontology Language

Bibliograf'ia

Βιβλιογραφία

- [1] Karl Aberer. P-Grid: A Self-Organizing Access Structure for P2P Information Systems. In *Proceedings of 9th International Conference on Cooperative Information Systems (CoopIS)*, pages 179–194, 2001.
- [2] Karl Aberer, Philippe Cudre-Mauroux, Manfred Hauswirth, and Tim Van Pelt. GridVine: Building Internet-Scale Semantic Overlay Networks. In *Proceedings of the Thirteenth International World Wide Web Conference (WWW2004)*, New York, May 2004.
- [3] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
- [4] Luc Onana Alima, Ali Ghodsi, Sameh El-Ansary, Seif Haridi, and Per Brand. Multicast in DKS(N, k, f) Overlay Networks. In *Proceedings of the 3rd International Conference on Peer-to-Peer Computing (P2P'2003)*, pages 196–197, 2002.
- [5] Luc Onana Alima, Ali Ghodsi, and Seif Haridi. A Framework for Structured Peer-to-Peer Overlay Networks. In Springer, editor, *Global Computing 2004*, volume 3267 of LNCS, pages 223–250, 2004.
- [6] Anupriya Ankolekar, Mark Burstein, Jerry R. Hobbs, Ora Lassila, David L. Martin, Sheila A. McIlraith, Srin Narayanan, Massimo Paolucci, Terence Payne, Katia Sycara, and Honglei Zeng. DAML-S: Semantic Markup For Web Services. In *Proceedings of the International Semantic Web Workshop*, 2001.
- [7] Arnold deVos. *An RDF query language based on DAML*. Available at <http://www.langdale.com.au/RDF/DAML-Query.html>.
- [8] Lerina Aversano, Gerardo Canfora, and Anna Ciampi. An algorithm for Web service discovery through their composition. In *Proceedings of the 2nd IEEE International Conference on Web Services (ICWS)*, pages 332–, San Diego, California, June 2004.

- [9] Hari Balakrishnan, M. Frans Kaashoek, David R. Karger, Robert Morris, and Ion Stoica. Looking up data in P2P systems. *Communications of the ACM*, 46(2):43–48, 2003.
- [10] Boualem Benatallah, Mohand-Said Hacid, Christopher Rey, and Farouk Toumani. Request Rewriting-Based Web Service Discovery. In *Proceedings of the 2nd International Semantic Web Conference (ISWC) 2003*, September 2003.
- [11] T. Berners-Lee, R. Fielding, and L. Masinter. RFC 2396 - Uniform Resource Identifiers (URI): Generic Syntax. <http://www.isi.edu/in-notes/rfc2396.txt>, August 1998.
- [12] Tim Berners-Lee. Notation 3 - An RDF Language for the Semantic Web. <http://www.w3.org/DesignIssues/Notation3>, 1998.
- [13] Abraham Bernstein and Mark Klein. Discovering services: Towards High-Precision Service Retrieval. In *Proceedings of the CaiSE workshop on Web Services, e-Business, and the Semantic Web: Foundations, Models, Architecture, Engineering and Applications*, Toronto, Canada, 2002.
- [14] A. Soydan Bilgin and Munindar P. Singh. A DAML-Based Repository for QoS-Aware Semantic Web Service Selection. In *Proceedings of the 2nd IEEE International Conference on Web Services (ICWS)*, pages 368–375, San Diego, California, June 2004.
- [15] Dan Brickley and R.V. Guha. Resource Description Framework (RDF) Schema Specification 1.0. Technical report, W3C Recommendation, 2000.
- [16] Jeen Broekstra and Arjohn Kampman. SeRQL: An RDF Query and Transformation Language. <http://www.cs.vu.nl/~jbroeks/papers/SeRQL.pdf>, 2004.
- [17] John Brooke, Donal Fellows, Kevin Garwood, and Carole Goble. Semantic matching of Grid Resource Descriptions. In *Proceedings of the European Across Grids Conference*, 2004.
- [18] M. Cai, M. Frank, and P. Szekely. MAAN: A Multi-Attribute Addressable Network for Grid Information Services. In *Proceedings of the 4th International Workshop on Grid Computing (Grid2003)*, 2003.
- [19] Min Cai, Martin R. Frank, Baoshi Yan, and Robert M. MacGregor. A Subscribable Peer-to-Peer RDF Repository for Distributed Metadata Management. *Journal of Web Semantics: Science, Services and Agents on the World Wide Web*, 2(2):109–130, December 2004.

- [20] Bruno Carton and Valentin Mesaros. Improving the Scalability of Logarithmic-Degree DHT-based Peer-to-Peer Networks. In *Proceedings of EuroPar 2004*, August-September 2004.
- [21] A. Carzaniga, D.S. Rosenblum, and A.L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, August 2001.
- [22] R.G.G. Cattell, Douglas K. Barry, Mark Berler, Jeff Eastman, David Jordan, Craig Russell, Olaf Shadow, Torsten Stanienda, and Fernando Velez. *The Object Database Standard ODMG 3.0*. Morgan Kaufmann, January 2000.
- [23] L. Chen, N.R. Shadbolt, F. Tao, C. Goble, C. Puleston, and S. Cox. Managing Semantic Metadata for the Semantic Grid. In *Proceedings of Knowledge Grid and Grid Intelligence (KGGI) Workshop*, Beijing, China, 2004.
- [24] P.-A. Chirita, S. Idreos, M. Koubarakis, and W. Nejdl. Publish/Subscribe for RDF-based P2P Networks. In *Proceedings of the 1st European Semantic Web Symposium (ESWS 2004)*, volume 3053 of *Lecture Notes in Computer Science*, pages 182–197, Heraklion, Crete, Greece, May 10-12 2004.
- [25] Ben Clifford. Globus Monitoring and Discovery. USC/ISI, The Globus Alliance. Presentation slides.
- [26] Ben Clifford. Monitoring and Discovery. USC Grid Computing course 2003. Presentation slides available at: www.isi.edu/anc/classes/fall2003/lecture3info.ppt.
- [27] Dan Connolly, Frank van Harmelen, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. DAML+OIL (March 2001) Reference Description. <http://www.w3.org/TR/daml+oil-reference>, 2001.
- [28] Karl Czajkowski, Steven Fitzgerald, Ian Foster, and Carl Kesselman. Grid Information Services for Distributed Resource Sharing. In *Proceedings of the 10th IEEE International Symposium on High-Performance Distributed Computing (HPDC-10)*. IEEE Press, 2001.
- [29] N. Daswani, H. Garcia-Molina, and B. Yang. Open Problems in Data Sharing Peer-to-Peer Systems. In *Proceedings of the 9th International Conference on Database Theory (ICDT 2003)*, volume 2572 of *Lecture Notes in Computer Science*, pages 1–15. Springer, January 2003.

- [30] Anwitaman Datta, Manfred Hauswirth, Roman Schmidt, Renault John, and Karl Aberer. Use of Computers at the Edge of Networks (P2P, Grid, Clusters). In *Fifth IEEE International Conference on Peer-to-Peer Computing*, Konstanz, Germany, August 31 - September 2 2005. To be published.
- [31] Seema Degwekar, Stanley Y. W. Su, and Herman Lam. Constraint Specification and Processing in Web Services Publication and Discovery. In *Proceedings of the 2nd IEEE International Conference on Web Services (ICWS)*, pages 210–217, San Diego, California, June 2004.
- [32] D. DeWitt and R. Gerber. Multiprocessor Hash-Based Join Algorithms. In *Proceedings of 11th International Conference on Very Large Data Bases (VLDB '85)*, Stockholm, Sweden, August 1985.
- [33] D. DeWitt and J. Gray. Parallel database systems: the future of high performance database systems. *Communications of the ACM*, 35(6), 1992.
- [34] Marc Ehrig, Peter Haase, F. van Harmelen, Ronny Siebes, Steffen Staab, Heiner Stuckenschmidt, Rudi Studer, and Christoph Tempich. The SWAP data and metadata model for semantics-based peer-to-peer systems. In Michael Schillo, Matthias Klusch, Jörg P. Müller, and Huaglory Tianfield, editors, *Proceedings of MATES-2003. First German Conference on Multiagent Technologies*, volume 2831 of *LNAI*, pages 144–155, Erfurt, Germany, SEP 2003. Springer.
- [35] Fatih Emekci, Ozgur D. Sahin, Divyakant Agrawal, and Amr El Abbadi. A Peer-to-Peer Framework for Web Service Discovery with Ranking. In *Proceedings of the 2nd IEEE International Conference on Web Services (ICWS)*, pages 192–199, San Diego, California, June 2004.
- [36] Dietmar W. Erwin and David F. Snelling. UNICORE: A Grid Computing Environment. In *Euro-Par '01: Proceedings of the 7th International Euro-Par Conference Manchester on Parallel Processing*, pages 825–834, London, UK, 2001. Springer-Verlag.
- [37] R. V. Guha. Rdfdb ql. <http://www.guha.com/rdfdb/query.html>.
- [38] K. Gummadi, R. Gummadi, S. Gribble, S. Ratnasamy, S. Shenker, and I. Stoica. The impact of DHT routing geometry on resilience and proximity. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 381–394, New York, NY, USA, 2003. ACM Press.
- [39] Krishna P. Gummadi, Richard J. Dunn, Stefan Saroiu, Steven D. Gribble, Henry M. Levy, and John Zahorjan. Measurement, modeling, and

- analysis of a peer-to-peer file-sharing workload. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, pages 314–329. ACM Press, 2003.
- [40] Volker Haarslev and Ralf Moller. Description of the RACER System and its Applications. In *Proceedings of the International Workshop in Description Logics (DL2001)*, Stanford, USA, 2001.
- [41] Peter Haase, Jeen Broekstra, Andreas Eberhart, and Raphael Volz. A comparison of RDF query languages. In *Proceedings of the Third International Semantic Web Conference (ISWC2004)*, Hiroshima, Japan, NOV 2004.
- [42] Peter Haase, Jeen Broekstra, Marc Ehrig, Maarten Menken, Peter Mika, Michal Plechawski, Pawel Pyszlak, Björn Schnizler, Ronny Siebes, Steffen Staab, and Christoph Tempich. Bibster - a semantics-based bibliographic peer-to-peer system. In Sheila A. McIlraith, Dimitris Plexousakis, and Frank van Harmelen, editors, *Proceedings of the Third International Semantic Web Conference, Hiroshima, Japan, 2004*, volume 3298 of LNCS, pages 122–136. Springer, November 2004.
- [43] Andreas Harth, Stefan Decker, Yu He, Hongsuda Tangmunarunkit, and Carl Kesselman. A Semantic Matchmaker Service on the Grid. In *WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, pages 326–327, New York, NY, USA, 2004. ACM Press.
- [44] Patrick Hayes. RDF Model Theory. <http://www.w3.org/TR/2002/WD-rdf-mt-20020429/>.
- [45] Felix Heine, Matthias Hovestadt, and Odej Kao. Towards Ontology-Driven P2P Grid Resource Discovery. In *GRID '04: Proceedings of the Fifth IEEE/ACM International Workshop on Grid Computing (GRID'04)*, pages 76–83, Washington, DC, USA, 2004. IEEE Computer Society.
- [46] Kirsten Hildrum, John Kubiawicz, Satish Rao, and Ben Y. Zhao. Distributed object location in a dynamic network. In *Proceedings of the Fourteenth Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 41–52, 2002.
- [47] T.A. Howes and M. Smith. A Scalable, Deployable Directory Service Framework for the Internet. Technical report, Center for Information Technology Integration, University of Michigan, 1995.
- [48] BerkeleyDB: <http://sleepycat.com/>.

- [49] R. Huebsch, J. M. Hellerstein, N. Lanham, B. T. Loo, S. Shenker, and I. Stoica. Querying the Internet with PIER. In *Proceedings of 29th International Conference on Very Large Data Bases (VLDB 2003)*, Berlin, Germany, September 2003.
- [50] 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. In *Proceedings of the 1st International Workshop on Peer-to-Peer Computing and DataBases (P2P&DB 2004)*, volume 3268 of *Lecture Notes in Computer Science*, pages 496–505, Heraklion, Crete, Greece, March 2004.
- [51] D. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy. Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 654–663, El Paso, Texas, May 1997.
- [52] Greg Karvounarakis, Sofia Alexaki, Vassilis Christophides, Dimitris Plexousakis, and Michel Scholl. RQL: A Declarative Query Language for RDF. In *Proceedings of the 11th International World Wide Web Conference*, May 2002.
- [53] Uwe Keller, Ruben Lara, Axel Polleres, and Dieter Fensel. Automatic Location of services. In *Proceedings of The Semantic Web: Research and Applications: Second European Semantic Web Conference (ESWC 2005)*, Heraklion, Crete.
- [54] Uwe Keller, Ruben Lara, Axel Polleres, Ioan Toma, Michel Kifer, and Dieter Fensel. D5.1v0.1 WSMO Web Service Discovery WSML Working Draft. DERI, 2004. <http://www.wsmo.org/2004/d5/d5.1/v0.1/>.
- [55] Michael Kifer, Georg Lausen, and James Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42(4):741–843, 1995.
- [56] G. Kokkinidis and V. Christophides. Semantic Query Routing and Processing in P2P Database Systems: The ICS-FORTH SQPeer Middleware. In *EDBT Workshops*, Heraklion, Crete, Greece, March 2004.
- [57] D. Kossman. The State of the art in Distributed Query Processing. *ACM Computing Surveys*, 32(4):422–469, September 2000.
- [58] Ioannis Kotsiopoulos, Sean Bechhofer, Pinar Alper, Paolo Missier, Oscar Corcho, Dean Kuo, and Carole Goble. Specification of a Semantic Grid Architecture. Deliverable 1.2, OntoGrid project.

- [59] Ora Lassila and Ralph R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. Technical report, W3C Recommendation, 1999.
- [60] Lei Li and Ian Horrocks. A Software Framework for Matchmaking Based on Semantic Web Technology. In *Proceedings of the 12th international conference on World Wide Web (WWW'03)*, Budapest, Hungary, May 2003.
- [61] Erietta Liarou, Stratos Idreos, and Manolis Koubarakis. Publish-Subscribe with RDF Data over Large Structured Overlay Networks. In *Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2005) to be held at VLDB 2005 31st International Conference on Very Large Data Bases*, Trondheim, Norway, 28-29 August.
- [62] Phillip Lord, Pinar Alper, Chris Wroe, and Carole Goble. Feta: A light-weight architecture for user oriented semantic service discovery. In *Proceedings of The Semantic Web: Research and Applications: Second European Semantic Web Conference (ESWC 2005)*, Heraklion, Crete.
- [63] Matoula Magiridou, Stavros Sahtouris, Vassilis Christophides, and Manolis Koubarakis. RUL: A Declarative Update Language for RDF. In *Proceedings of the Fourth International Semantic Web Conference (ISWC2005)*, 2005.
- [64] Aimilia Magkanaraki, Val Tannen, Vassilis Christophides, and Dimitris Plexousakis. Viewing the Semantic Web Through RVL Lenses. In *Proceedings of the Second International Semantic Web Conference (ISWC2003)*, 2003.
- [65] Dahlia Malkhi, Moni Naor, and David Ratajczak. Viceroy: a scalable and dynamic emulation of the butterfly. In *Proceedings of the Twenty-First Annual ACM Symposium on Principles of Distributed Computing (PODC)*, pages 183–192, 2002.
- [66] M. Mehta and J. DeWitt. Data placement in shared-nothing parallel database systems. *VLDB Journal*, 6:53–72, 1997.
- [67] D. S. Milojisic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu. Peer-to-Peer Computing. Technical Report HP-2002-57, HP Laboratories, Palo Alto, 2001.
- [68] W. Neidl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch. EDUTELLA: A P2P networking infrastructure based on RDF. In *Proceedings of the 11th International World Wide Web Conference*, May 2002.

- [69] W. Nejdl, W. Siberski, U. Thaden, and W.-T. Balke. Top-k Query Evaluation for Schema-Based Peer-to-Peer Networks. In *Proceedings of the 3rd International Semantic Web Conference (ISWC)*, 2004.
- [70] W. Nejdl, B. Wolf, S. Staab, and J. Tane. EDUTELLA: Searching and Annotating Resources within an RDF-based P2P Network. In *Proceedings of the 11th WWW Conference*, Hawaii, USA, May 2002.
- [71] W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. Schlosser, I. Brunkhorst, and A. Loser. Super-Peer-Based Routing and Clustering Strategies for RDF-Based Peer-To-Peer Networks. In *Proceedings of the 12th WWW Conference*, Budapest, Hungary, May 2003.
- [72] Nikos Ntarmos, Theoni Pitoura, and Peter Triantafillou. Range query optimization leveraging peer heterogeneity. In *3rd International Workshop on Databases, Information Systems and Peer-to-Peer Computing (DBISP2P 2005)*, August 2005.
- [73] National Institute of Standards and Technology. Secure hash standard, 1995. Publication 180-1.
- [74] Massimo Paolucci, Takahiro Kawamura, Terry R. Payne, and Katia Sycara. Importing the semantic web in UDDI. In *Proceedings of E-Services and the Semantic Web Workshop*, 2002.
- [75] Massimo Paolucci, Katia Sycara, Takuya Nishimura, and Naveen Srinivasan. Using DAML-S for P2P Discovery. In *Proceedings of the First International Conference on Web Services (ICWS'03)*, pages 203-207, Las Vegas, Nevada, USA, June 2003.
- [76] Mike P. Papazoglou, Bernd J. Kramer, and Jian Yang. Leveraging Web-Services and Peer-to-Peer Networks. In J. Eder and M. Missikoff, editor, *Proceedings of CAiSE 2003*, volume LNCS 2681, pages 485-501. Springer-Verlag Berlin Heidelberg, 2003.
- [77] R. Raman, M. Livny, and M. Solomon. Matchmaking: Distributed Resource Management for High Throughput Computing. In *HPDC '98: Proceedings of the The Seventh IEEE International Symposium on High Performance Distributed Computing*, page 140, Washington, DC, USA, 1998. IEEE Computer Society.
- [78] Rajesh Raman, Miron Livny, and Marvin Solomon. Resource Management through Multilateral Matchmaking. In *HPDC '00: Proceedings of the Ninth IEEE International Symposium on High Performance Distributed Computing (HPDC'00)*, page 290, Washington, DC, USA, 2000. IEEE Computer Society.

- [79] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A Scalable Content-addressable Network. In *Proceedings of the ACM SIGCOMM '01 Conference*, San Diego, California, August 2001.
- [80] Sean Rhea, Dennis Geels, Timothy Roscoe, and John Kubiawicz. Handling Churn in a DHT. In *USENIX Annual Technical Conference*, 2004.
- [81] Dumitru Roman, Holger Lausen, and Uwe Keller. D2v1.2. Web Service Modeling Ontology (WSMO) WSMO Final Draft. DERI, 2005. <http://www.wsmo.org/TR/d2/v1.2/20050413/>.
- [82] A. Rowstron and P. Druschel. Pastry: Scalable, Distributed Object Location and Routing for Large-Scale- Peer-to-Peer Storage Utility. In *Proceedings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)*, November 2001.
- [83] B. Sapkota, L. Vasiliu, C. Bussler, D. Roman, and I. Toma. Peer-to-Peer Technology Usage in Web Service Discovery and Matchmaking. In *To appear in Proceedings of 6th International Conference on Web Information Systems Engineering (WISE 2005)*, New York, NY, USA, November 2005.
- [84] S. Saroiu, K. Gummadi, and S. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Conferencing and Networking*, January 2002.
- [85] M. Schlosser, M. Sintek, S. Decker, and W. Nejdl. A Scalable and Ontology-Based P2P Infrastructure for Semantic Web Services. In *Proceedings of the 2nd IEEE International Conference on Peer-to-Peer Computing*, 2002.
- [86] Mario Schlosser, Michael Sintek, Stefan Decker, and Wolfgang Nejdl. HyperCuP - hypercubes, ontologies and efficient search on peer-to-peer networks, May 2003.
- [87] Cristina Schmidt and Manish Parashar. A Peer-to-Peer Approach to Web Service Discovery. *World Wide Web: Internet and Web Information Systems*, Kluwer Academic Publishers. Manufactured in The Netherlands(7):211-229, 2004.
- [88] D. Schneider and D. DeWitt. Tradeoffs in Processing Complex Join Queries via Hashing in Multiprocessor Database Machines. In *Proceedings of 16th International Conference on Very Large Data Bases (VLDB'90)*, pages 469-480, Brisbane, Queensland, Australia.
- [89] Andy Seaborne. RDQL - A query Language for RDF, W3C Member Submission 9 January 2004.

- <http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/>,
Jan 2004.
- [90] <http://www.bamboo-dht.org/programmers-guide.html> Sean Rhea,
Bamboo: Programmer's guide, 2004.
- [91] Subhabrata Sen and Jia Wang. Analyzing peer-to-peer traffic across
large networks. In *Proceedings of the second ACM SIGCOMM Workshop
on Internet measurement*, pages 137–150. ACM Press, 2002.
- [92] Bernd Simon, Zoltan Miklos, Wolfgang Nejdl, Michael Sintek, and
Joaquin Salvachua. Elena: A Mediation Infrastructure for Educational
Services. In *Proceedings of Twelfth International World Wide Web Confer-
ence (WWW2003)*, Budapest, Hungary, May 2003.
- [93] Michael Sintek and Stefan Decker. TRIPLE – A Query, Inference and
Transformation Language for the Semantic Web. In *Proceedings of De-
ductive Databases and Knowledge Management (DDL'2001)*, 2001.
- [94] Kaarthik Sivashanmugam, Kunal Verma, and Amit Sheth. Discovery
of Web Services in a Federated Registry Environment. In *Proceedings
of the 2nd IEEE International Conference on Web Services (ICWS)*, pages
270–278, San Diego, California, June 2004.
- [95] Steffen Staab and Heiner Stuckenschmidt, editors. *Semantic Web and
Peer-to-Peer*. Springer-Verlag. Forthcoming book.
- [96] I. Stoica, D. Adkins, S. Ratnasamy, S. Shenker, S. Surana, and
S. Zhuang. Internet Indirection Architecture. In *Proceedings of ACM
SIGCOMM'02*, pages 73–86, August 2002.
- [97] Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari
Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet
applications. In *Proceedings of the ACM SIGCOMM '01 Conference*, San
Diego, California, August 2001.
- [98] Ion Stoica, Robert Morris, David Liben-Nowell, David Karger, M. Fran-
s Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: a scalable
peer-to-peer lookup protocol for internet applications. *IEEE/ACM Trans-
actions on Networking*, 11(1):17–32, 2003.
- [99] M. Stonebraker. The case for shared nothing. *Database Engineering*,
9(1), 1986.

- [100] Stanley Y. W. Su, Chunbo Huang, Joachim Hammer, Yihua Huang, Haifei Li, Liu Wang, Youzhong Liu, Charnyote Pluempitiwiriawej, Minsoo Lee, and Herman Lam. An internet-based negotiation server for e-commerce. *VLDB Journal*, 10(1):72–90, 2001.
- [101] Stanley Y.W. Su, Chunbo Huang, Joachim Hammer, Yihua Huang, Haifei Li, Liu Wang, Youzhong Liu, Charnyote Pluempitiwiriawej, Minsoo Lee, and Herman Lam. An Internet-based Negotiation Server for e-commerce. *The VLDB Journal*, 10(1):72–90, 2001.
- [102] Katia Sycara, Massimo Paolucci, Anupriya Ankolekar, and Naveen Srinivasan. Automated discovery, interaction and composition of Semantic Web services. *Journal of Web Semantics, Web Semantics: Science, Services and Agents on the World Wide Web* 1:27–46, 2003.
- [103] Valentina Tamma, Ian Blacoe, Ben Lithgow Smith, and Michael Wooldridge. SERSE: Searching for Semantic Web Content. In *Proceedings of the AAMAS 2004 workshop on Challenges in the coordination of large scale multi-agent systems*, New York, July 2004.
- [104] Hongyuda Tangmunarunkit, Stefan Decker, and Carl Kesselman. Ontology-Based Resource Matching in the Grid - The Grid Meets the Semantic Web. In *Proceedings of the 2nd International Semantic Web Conference*, pages 706–721, 2003.
- [105] P. Triantafyllou, C. Xiruhaki, M. Koubarakis, and N. Ntarmos. Towards high-performance peer-to-peer content and resource sharing systems. In *Proceedings of the First Biennial Conference on Innovative Data Systems Research (CIDR 2003)*, January 2003.
- [106] Dimitrios Tsoumakos and Nick Roussopoulos. A comparison of peer-to-peer search methods. In *Proceedings of International Workshop on Web and Databases (WebDB)*, pages 61–66, 2003.
- [107] UDDI. *The UDDI Technical White Paper*. Available at <http://www.uddi.org/>.
- [108] Kunal Verma, Kaarthik Sivashanmugam, Amit Sheth, Abhijit Patil, Swapna Oundhakar, and John Miller. METEOR-S WSDI: A Scalable Infrastructure of Registries for Semantic Publication and Discovery of Web Services. *Journal of Information Technology and Management, Special Issue on Universal Global Integration*, 6(1):17–39.
- [109] Yiqiao Wang and Eleni Stroulia. Flexible Interface Matching for Web-Service Discovery. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering*, Roma, Italy, December 2003.

- [110] Yiqiao Wang and Eleni Stroulia. Semantic Structure Matching for Assessing Web-Service Similarity. In *Proceedings of the First International Conference on Service Oriented Computing*, Trento, Italy, December 2003.
- [111] Matt Welsh, David Culler, and Eric Brewer. Seda: An architecture for well-conditioned, scalable internet services. In *Eighteenth Symposium on Operating Systems Principles (SOSP-18)*, October 2001.
- [112] C. Wroe, R. Stevens, C. Goble, A. Roberts, and M. Greenwood. A suite of DAML+OIL Ontologies to describe bioinformatics web services and data. *International Journal of Cooperative Information Systems*, 12(2):197-224, 2003.
- [113] B. Yang and H. Garcia-Molina. Designing a super-peer network. In *Proceedings of the 19th International Conference on Data Engineering (ICDE 2003)*, March 2003.