Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής

Διδακτορική Διατριβή

Ευφυές, προσαρμοστικό σύστημα πολλών πρακτόρων για τη συλλογή πληροφοριών από το διαδίκτυο.

Νικολάου Ζ. Ζάχαρη

Πειραιάς, Ιούνιος 2002

ΝΟ : 13412
ΝΕΕ : 13282

Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής

Διδακτορική Διατριβή

2 8 ΝΟΕ. 2002

Ευφυές, προσαρμοστικό σύστημα πολλών πρακτόρων για τη συλλογή πληροφοριών από το διαδίκτυο.

Νικολάου Ζ. Ζάχαρη

Πειραιάς, Ιούνιος 2002

Πανεπιστήμιο Πειραιώς
Τμήμα Πληροφορικής

Διδακτορική Διατριβή

Ευφυές, προσαρμοστικό σύστημα πολλών πρακτόρων για τη συλλογή πληροφοριών από το διαδίκτυο.

Νικολάου Ζ. Ζάχαρη

Συμβουλευτική Επιτροπή

Επιβλέπων

Θεμιστοκλής Παναγιωτόπουλος, Επίκουρος Καθηγητής Πανεπιστημίου Πειραιώς

Μέλη

Νικόλαος Αλεξανδρής, Καθηγητής ΠανεπιστημίουΠειραιώς

Δημήτριος Δεσπότης, Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιώς

Εξεταστική Επιτροπή

Νικόλαος Αλεξανδρής, Καθηγητής Πανεπιστημίου Πειραιώς

Γεώργιος Παπακωνσταντίνου, Καθηγητής Εθνικού Μετσοβίου Πολυτεχνείου

Εμμανουήλ Σκορδαλάκης, Καθηγητής Εθνικού Μετσοβίου Πολυτεχνείου

Δημήτριος Δεσπότης, Αναπληρωτής Καθηγητής Πανεπιστημίου Πειραιώς

Θεμιστοκλής Παναγιωτόπουλος, Επίκουρος Καθηγητής Πανεπιστημίου Πειραιώς

Χρήστος Δουληγέρης, Επίκουρος Καθηγητής Πανεπιστημίου Πειραιώς

Δημήτρης Γκιζόπουλος, Λέκτορας Πανεπιστημίου Πειραιώς

Περιεχόμενα

Κεφάλαιο 1 ΕΙΣΑΓΩΓΗ

1.1 Εισαγωγή 1

1.2 Αντικείμενο της Διδακτορικής Διατριβής 3

1.3 Δομή της Διδακτορικής Διατριβής 5

1.4 Συνεισφορά στην Ερευνητική Περιοχή 7

1.5 Ευχαριστίες 8

Κεφάλαιο 2 Η ΑΝΑΠΤΥΞΗ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ

2.1 Εισαγωγή 9

2.2 Το Πρωτόκολλο Επικοινωνίας TCP/IP 11

2.3 Το Πρωτόκολλο Επικοινωνίας HTTP 13

2.4 Η Γλώσσα HTML και το Πρόγραμμα Πλοήγησης 15

2.5 Προβλήματα στην Οργάνωση του Διαδικτύου 18

2.6 Συμπεράσματα 19

Κεφάλαιο 3 ΑΝΑΖΗΤΗΣΗ ΠΛΗΡΟΦΟΡΙΩΝ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

3.1 Εισαγωγή 20

3.2 Διαδικτυακοί Κόμβοι Αναζήτησης Πληροφοριών 22

3.2.1 Η Δημιουργία της Βάσης Δεδομένων των Διαδικτυακών Κόμβων Αναζήτησης Πληροφοριών 26

3.2.2 Η Καταλογογράφηση και Ταξινόμηση της Βάσης Δεδομένων των Διαδικτυακών Κόμβων Αναζήτησης Πληροφοριών 28

3.2.3 Θεματικοί Κατάλογοι ή Μηχανές Αναζήτησης; 31

3.3 Προβλήματα απο τη Χρήση των Διαδικτυακών Κόμβων Αναζήτησης Πληροφοριών 33

3.4 Συμπεράσματα 34

Κεφάλαιο 4 ΠΡΑΚΤΟΡΕΣ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΤΗΝ ΣΥΛΛΟΓΗ ΠΛΗΡΟΦΟΡΙΩΝ ΑΠΟ ΤΟ ΔΙΑΔΙΚΤΥΟ

4.1 Εισαγωγή 36

4.2 Τι είναι ένας Πράκτορας Λογισμικού; 38

4.3 Κατηγοριοποίηση των Πρακτόρων Λογισμικού 40

4.3.1 Κινητοί Πράκτορες Λογισμικού 41

4.3.2 Στατικοί Πράκτορες Λογισμικού 42

4.4 Η Τεχνολογία των Πρακτόρων Λογισμικού για την Συλλογή Πληροφοριών από το Διαδίκτυο 43

4.4.1 Φιλτράρισμα και Συλλογή Πληροφοριών στο Διαδίκτυο 43

4.4.2 Στόχοι των Πρακτόρων Λογισμικού για Συλλογή Πληροφοριών από το Διαδίκτυο 44

4.4.3 Τα Στάδια Λειτουργίας ενός Πρακτόρα Λογισμικού για Συλλογή Πληροφοριών από το Διαδίκτυο 46

4.4.4 Τεχνικές Ανάπτυξης των Πρακτόρων Λογισμικού για Συλλογή Πληροφοριών από το Διαδίκτυο 49

4.5 Συμπεράσματα 54

Κεφάλαιο 5 ΤΟ ΣΥΣΤΗΜΑ WEBNAUT

5.1 Εισαγωγή 55

5.2 Η Αρχιτεκτονική του Συστήματος Webnaut 57

5.3 Σχεδίαση του Πράκτορα Διακομιστή Διαμεσολάβησης 58

5.3.1 Λειτουργικότητα του Πράκτορα Διακομιστή Διαμεσολάβησης 61

5.3.2 Η Πινακίδα Κεντρικού Ελέγχου 64

5.3.3 Λεπτομερής Σχεδιασμός του Πράκτορα Διακομιστή Διαμεσολάβησης 67

5.4 Σχεδίαση του Πράκτορα Φιλοξενούμενων Δικτυακών Τόπων .. 70

5.4.1 Λεπτομερής Σχεδιασμός του Πράκτορα Φιλοξενούμενων Δικτυακών Τόπων 71

5.4.2 Λειτουργικότητα του Πράκτορα Φιλοξενούμενων Δικτυακών Τόπων 72

5.4.3 Σχεδίαση Διεπαφής του Πράκτορα Φιλοξενούμενων

Δικτυακών Τόπων 75
5.5 Σχεδίαση του Πράκτορα Μαυροπίνακα 76
5.5.1 Λεπτομερής Σχεδιασμός του Πράκτορα Μαυροπίνακα 77
5.6 Σχεδίαση του Πράκτορα Εξυπηρετητή Διαδικτυακού Κόμβου . 78
5.6.1 Λεπτομερής Σχεδιασμός του Πράκτορα Εξυπηρετητή Διαδικτυακού Κόμβου 79
5.6.2 Λειτουργικότητα του Πράκτορα Εξυπηρετητή Διαδικτυακού Κόμβου 79
5.7 Σχεδίαση του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών 80
5.7.1 Λειτουργικότητα του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών 81
5.7.2 Λεπτομερής Σχεδιασμός του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών 84
5.7.3 Σχεδίαση Διεπαφής του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών 90
5.7.4 Ερευνητικές Προεκτάσεις και Εφαρμογές της Λειτουργίας του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών 91
5.8 Σχεδίαση του Πράκτορα Πύλη Εικονικής Πραγματικότητας ... 94
5.8.1 Λεπτομερής Σχεδιασμός του Εξυπηρετητή της Υπηρεσίας Εικονικής Πραγματικότητας 96
5.8.2 Λειτουργικότητα του Εξυπηρετητή της Υπηρεσίας Εικονικής Πραγματικότητας 97
5.8.3 Υλοποίηση του Κόσμου Εικονικής Πραγματικότητας 99
5.8.4 Η Διεπαφή του Πράκτορα Πύλη Εικονικής Πραγματικότητας 101
5.8.5 Λεπτομερής Σχεδιασμός του Πράκτορα Πύλη Εικονικής Πραγματικότητας 104
5.9 Σχεδίαση του Πράκτορα Εκμάθησης 107
5.9.1 Ο Αναλυτής Κειμένων 107
5.9.2 Λεπτομερής Σχεδιασμός του Αναλυτή Κειμένων 109
5.9.3 Το Ιστορικό Αρχείο 110
5.9.4 Λεπτομερής Σχεδιασμός του Ιστορικού Αρχείου 110
5.9.5 Ο Πράκτορας Γενετικός Αλγόριθμος 111
5.10 Συμπεράσματα 111

Κεφάλαιο 6 Η ΛΕΙΤΟΥΡΓΙΑ ΤΗΣ ΠΡΟΣΑΡΜΟΖΟΜΕΝΗΣ ΕΚΜΑΘΗΣΗΣ ΤΩΝ ΕΝΔΙΑΦΕΡΟΝΤΩΝ ΤΟΥ ΧΡΗΣΤΗ

6.1 Εισαγωγή 112

6.2 Η Λειτουργία του Πράκτορα Εκμάθησης 114

6.3 Το Προφίλ των Ενδιαφερόντων του Χρήστη 115

6.4 Η Δημιουργία του Προφίλ του Χρήστη 117

6.5 Λεπτομερής Σχεδιασμός του Προφίλ του Χρήστη 120

6.6 Σχεδίαση του Πράκτορα Γενετικός Αλγόριθμος 121

6.6.1 Οι Γενετικοί Αλγόριθμοι 122

6.6.2 Λειτουργικότητα του Πράκτορα Γενετικός Αλγόριθμος 124

6.6.3 Λεπτομερής Σχεδιασμός του Πράκτορα Γενετικός Αλγόριθμος 128

6.6.4 Λεπτομερής Σχεδιασμός των Γενετικών Μηχανισμών 131

6.7 Η Ανάδραση του Χρήστη 135

6.8 Συμπεράσματα 140

Κεφάλαιο 7 ΑΞΙΟΛΟΓΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

7.1 Εισαγωγή 141

7.2 Πρώτο Πείραμα-Αναζήτηση των Ενδιαφερόντων του Χρήστη . 141

7.2.1 Προετοιμασία του Πρώτου Πειράματος 142

7.2.2 Εκτέλεση του Πρώτου Πειράματος 145

7.3 Δεύτερο Πείραμα-Προσαρμογή στα Ενδιαφέροντα του Χρήστη 148

7.3.1 Προετοιμασία του Δεύτερου Πειράματος 148

7.3.2 Εκτέλεση του Δεύτερου Πειράματος 149

7.4 Συμπεράσματα 150

Κεφάλαιο 8 ΣΥΜΠΕΡΑΣΜΑΤΑ - ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

8.1 Συμπεράσματα 151

8.2 Μελλοντικές Επεκτάσεις 154

Συντομογραφίες 155

Παράρτημα Α 156
Παράρτημα Β 157
Παράρτημα Γ 168
Βιβλιογραφία 172

Κεφάλαιο 1

ΕΙΣΑΓΩΓΗ

1.1 Εισαγωγή

Τα τελευταία χρόνια παρατηρούμε τη γιγάντωση του διαδικτύου αφού πλέον είναι παγκόσμια η αποδοχή του ως πλατφόρμας ανάκτησης και διάδοσης πληροφοριών, ηλεκτρονικού εμπορίου και διασκέδασης. Το πλήθος των αναγνωστών, σε συνδυασμό με την ευκολία της συγγραφής και ανάρτησης μιας ιστοσελίδας, είναι οι κύριοι λόγοι που ωθούν τους χρήστες να χρησιμοποιούν το διαδίκτυο ως μέσο προβολής και διάδοσης των πληροφοριών.

Η χρήση του παγκόσμιου ιστού για την παρουσίαση πληροφοριών δεν είναι χωρίς προβλήματα, τα οποία σχετίζονται κυρίως με την οργάνωση, την διαχείριση και την συντήρηση της πληροφορίας. Αυτά τα προβλήματα σε συνδυασμό με την απουσία μιας αρχής-οργανισμού, που σκοπό θα έχει τον έλεγχο των πληροφοριών του διαδικτύου για τη δημιουργία ενός κεντρικού καταλόγου περιεχομένων, δημιουργούν μεγάλα προβλήματα στην έρευνα ενός χρήστη προς αναζήτηση πληροφοριών, οι οποίες είναι σχετικές με τα ενδιαφέροντα του.

Οι μηχανές αναζήτησης και οι θεματικοί κατάλογοι, είναι δύο προσεγγίσεις για την δημιουργία ευρετηρίου στα περιεχόμενα του διαδικτύου. Τα πλεονεκτήματα και τα μειονεκτήματα μοιράζονται εξίσου και στις δύο προσεγγίσεις και καμία δεν παρέχει τον καλύτερο κατάλογο αναζήτησης, για όλα τα θέματα. Ο κάθε χρήστης για να αποκτήσει χρήσιμες πληροφορίες, θα πρέπει να πλοηγηθεί από μια δικτυακή τοποθεσία σε μια άλλη, να μάθει τις διαφορετικές διεπαφές, τις διαθέσιμες επιλογές και τα χαρακτηριστικά του κάθε καταλόγου, να αποστείλει τις ερωτήσεις, να συλλέξει τις απαντήσεις και να ομαδοποιήσει τα αποτελέσματα.

Επιπλέον, ο χρήστης θα πρέπει αρκετές φορές να αλλάξει τη στρατηγική έρευνας, να κάνει χρήση διαφορετικής ορολογίας, να χρησιμοποιήσει διαφορετικούς συνδυασμούς λέξεων με λογικούς τελεστές, και πάντα θα πρέπει να είναι προετοιμασμένος να επισκεφτεί από δεκάδες μέχρι μερικές εκατοντάδες δικτυακούς κόμβους, χωρίς κανείς να του εξασφαλίζει με σιγουριά, ότι θα εντοπίσει την

επιθυμητή πληροφορία.

Ερευνητές από τον χώρο της τεχνητής νοημοσύνης έχουν υποδείξει μεθόδους για την ανάπτυξη λογισμικού με σκοπό τη συλλογή και το φιλτράρισμα της πληροφορίας από μεγάλους, σύνθετους, αδόμητους και ετερογενείς χώρους πληροφοριών, όπως ο παγκόσμιος ιστός. Τα τελευταία χρόνια, η ιδέα ενός πράκτορα λογισμικού έχει κερδίσει έδαφος, ως το πλαίσιο υλοποίησης τέτοιων συστημάτων που θα υποδεικνύουν και θα συμβουλεύουν τους χρήστες στην εύρεση και συλλογή χρήσιμης πληροφορίας.

Ο όρος *πράκτορας* έχει μεταφορική έννοια στην επιστήμη των υπολογιστών, αφού αποδίδεται στο λογισμικό το οποίο εκτελεί εργασίες για λογαριασμό ενός χρήστη ή ενός άλλου προγράμματος. Ειδικότερα στο χώρο της τεχνητής νοημοσύνης, ο πράκτορας λογισμικού θα πρέπει να εκτελεί τις εργασίες έχοντας γνώση ή αναπαράσταση των στόχων του χρήστη ή των επιθυμιών του, καθώς επίσης να λειτουργεί με κάποιο βαθμό ανεξαρτησίας ή αυτονομίας.

Σύμφωνα με τον Matthias Klusch [Klu01], οι περισσότεροι πράκτορες λογισμικού που κατασκευάζονται σήμερα, είτε στα ακαδημαϊκά ιδρύματα είτε στα εργαστήρια των εμπορικών εταιριών, και δραστηριοποιούνται στο χώρο του διαδικτύου με σκοπό την μείωση του φόρτου εργασίας λόγω υπερβολικής ενημέρωσης, δεν είναι ικανοί να υποστηρίξουν τους χρήστες σε πραγματικές συνθήκες. Ο ίδιος ερευνητής, προβλέπει ότι όλοι οι φιλόδοξοι στόχοι των ερευνητών καθώς και οι υπερβολικές απαιτήσεις των χρηστών από τους πράκτορες λογισμικού, θα έχουν ικανοποιηθεί μέσα στην επόμενη δεκαετία.

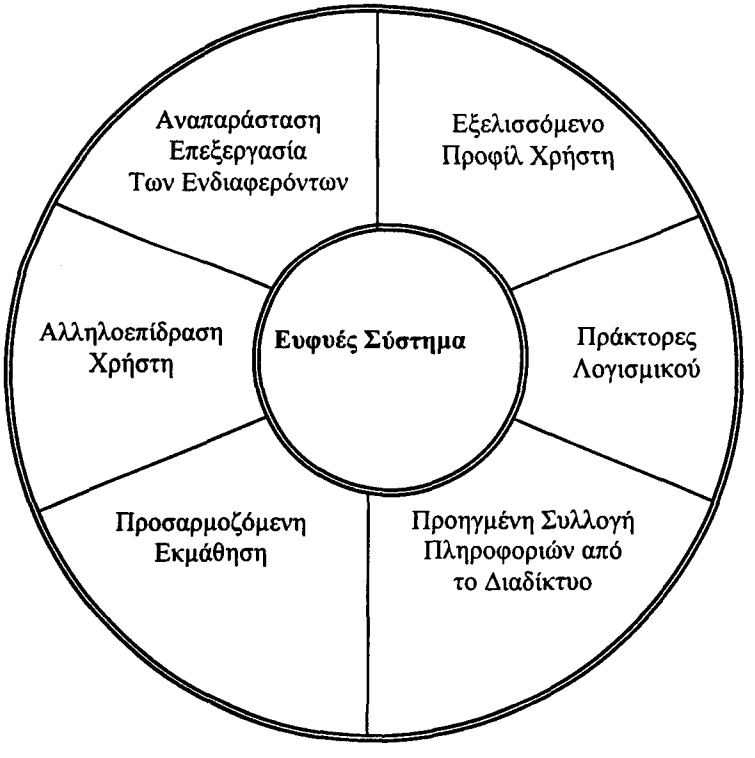
Στα πλαίσια αυτής της ερευνητικής προσπάθειας τοποθετείται και η παρούσα Διδακτορική Διατριβή η οποία κινείται στην περιοχή της ανάδειξης μεθοδολογίας για την αναπαράσταση και την διαχείριση των ενδιαφερόντων ενός χρήστη του διαδικτύου με την χρήση ευφυών συστημάτων, που αποτελούνται από πολλούς πράκτορες λογισμικού, οι οποίοι συλλέγουν πληροφορίες από το διαδίκτυο και προσαρμόζονται στα ενδιαφέροντα του χρήστη, καθώς αυτά εξελίσσονται δυναμικά μέσα στο χρόνο.

1.2 Αντικείμενο της Διδακτορικής Διατριβής

Αντικείμενο της παρούσας Διδακτορικής Διατριβής αποτελεί η παρουσίαση μεθοδολογίας :

- για την αναπαράσταση και την επεξεργασία των ενδιαφερόντων ενός χρήστη του διαδικτύου.
- για την εκμάθηση και την προσαρμογή στα ενδιαφέροντα του χρήστη
- για την διαχείριση του προφίλ του χρήστη καθώς τα ενδιαφέροντα του εξελίσσονται μέσα στο χρόνο
- για την δημιουργία ευφυών συστημάτων, τα οποία αποτελούνται από πολλούς πράκτορες λογισμικού, οι οποίοι συνεργάζονται μεταξύ τους και δραστηριοποιούνται στο διαδίκτυο.
- για την αλληλοεπίδραση ενός χρήστη με συστήματα πολλών πρακτόρων
- για την δημιουργία ευέλικτων μεταμηχανών αναζήτησης με σκοπό την συλλογή πληροφοριών από πολλές και διαφορετικές πηγές

Επίσης στην Διδακτορική Διατριβή παρουσιάζεται η σχεδίαση και η λειτουργία του συστήματος Webnaut, το οποίο έχει αναπτυχθεί με τις παραπάνω μεθοδολογίες και στόχο έχει να υποστηρίξει τις πληροφοριακές ανάγκες ενός χρήστη του διαδικτύου.



Αντικείμενα της Διατριβής

Η μεθοδολογία που προτείνουμε για την αναπαράσταση των ενδιαφερόντων του χρήστη, δεν απευθύνεται στους χρήστες ενός συγκεκριμένου γνωστικού αντικειμένου αλλά είναι γενική, δηλαδή, έχει εφαρμογή σε όλα τα γνωστικά πεδία, και επίσης απευθύνεται σε όλους τους χρήστες, ανεξαρτήτως εμπειρίας ή γνώσεως.

Η μεθοδολογία για την εκμάθηση των ενδιαφερόντων του χρήστη, επιτρέπει την δημιουργία συστημάτων τα οποία προσαρμόζονται στα ενδιαφέροντα του χρήστη. Επίσης, το προφίλ του χρήστη δεν είναι στατικό αλλά εξελίσσεται καθώς τα ενδιαφέροντα, η εμπειρία και η γνώση του χρήστη, μεταβάλλεται με το πέρασμα του χρόνου.

Η μεθοδολογία ανάπτυξης του συστήματος επιτρέπει την υλοποίηση συστημάτων, τα οποία αποτελούνται από πολλούς πράκτορες λογισμικού, οι οποίοι είναι εξειδικευμένοι σε μια συγκεκριμένη εργασία αλλά όλοι μαζί συνεργάζονται αρμονικά για την δημιουργία ενός ομοιογενούς συστήματος.

Η μεθοδολογία ανάπτυξης του συστήματος επιτρέπει την προσθήκη νέων λειτουργιών ή υπηρεσιών, την υποστήριξη του χρήστη καθ'όλη την διάρκεια της πλοήγησης του στο διαδίκτυο, την συλλογή πληροφοριών για τα ενδιαφέροντα του χρήστη από την αλληλοεπίδραση του με διαφορετικές υπηρεσίες του συστήματος, την συνεργασία με όλους τους φυλλομετρητές, την αύξηση της ασφάλειας του συστήματος.

Επίσης, με την σχεδίαση μιας ευέλικτης μεταμηχανής αναζήτησης ενισχύονται οι δυνατότητες του συστήματος αφού κατά αυτό το τρόπο δίνεται η δυνατότητα στο χρήστη να προσθέσει νέες πηγές πληροφοριών.

ΕΘΝΙΚΟ ΚΕΝΤΡΟ ΤΕΚΜΗΡΙΩΣΗΣ
ΒΙΒΛΙΟΘΗΚΗ

1.3 Δομή της Διδακτορικής Διατριβής

Η παρούσα Διδακτορική Διατριβή αποτελείται από τα παρακάτω κεφάλαια :

Στο **Κεφάλαιο 1**, *Εισαγωγή*, παρουσιάζουμε το αντικείμενο, τους στόχους καθώς και την συνεισφορά στην ερευνητική περιοχή, της Διδακτορικής Διατριβής.

Στο **Κεφάλαιο 2**, *Η Ανάπτυξη του Διαδικτύου*, γίνεται σύντομη παρουσίαση του διαδικτύου, ο οποίος είναι και το περιβάλλον λειτουργίας του συστήματος Webnaut. Οι σκοποί αυτού του κεφαλαίου είναι : α) να παρουσιάσουμε, τα πρωτόκολλα επικοινωνιών τόσο σε επίπεδο δικτύου όσο και σε επίπεδο εφαρμογών που θα πρέπει να χρησιμοποιήσει ένα σύστημα που θα δραστηριοποιηθεί στο διαδίκτυο, όπως το σύστημα Webnaut, β) να δείξουμε ότι δεν είναι εύκολη υπόθεση η αναζήτηση πληροφοριών αφού υπάρχουν αρκετά προβλήματα που σχετίζονται με την οργάνωση, την διαχείριση και την συντήρηση των πληροφοριών στο διαδίκτυο. Τέλος, η απουσία ενός οργανισμού, που σκοπό θα έχει τη δημιουργία ενός κεντρικού καταλόγου περιεχομένων, δημιουργεί ακόμα μεγαλύτερα προβλήματα στην αναζήτηση πληροφοριών.

Στο **Κεφάλαιο 3**, *Αναζήτηση Πληροφοριών στο Διαδίκτυο*, παρουσιάζουμε όλες τις δυνατότητες που έχει στην διάθεση του ένας χρήστης με σκοπό την αναζήτηση και συλλογή πληροφοριών από το διαδίκτυο. Πιο συγκεκριμένα, παρουσιάζουμε τους διαδικτυακούς κόμβους αναζήτησης πληροφοριών, οι οποίοι χωρίζονται σε δύο κατηγορίες, στις μηχανές αναζήτησης και στους θεματικούς καταλόγους. Ειδικότερα αναλύουμε, θέματα που σχετίζονται με τον τρόπο λειτουργίας, τα πλεονεκτήματα και τα μειονεκτήματα της κάθε κατηγορίας, καθώς και θέματα που δεν επιλύουν οι διαδικτυακοί κόμβοι αναζήτησης πληροφοριών. Τέλος, παρουσιάζεται η ανάγκη για την δημιουργία ευφυών συστημάτων, τα οποία θα συλλέγουν πληροφορίες από το διαδίκτυο και θα προσαρμόζονται στα ενδιαφέροντα ενός χρήστη.

Στο **Κεφάλαιο 4**, *Πράκτορες Λογισμικού για την Συλλογή Πληροφοριών από το Διαδίκτυο*, παρουσιάζουμε τις ιδιότητες και τα χαρακτηριστικά που διαθέτουν οι πράκτορες λογισμικού. Επίσης, κάνουμε μια κατηγοριοποίηση και παρουσίαση των πεδίων εφαρμογής των πρακτόρων λογισμικού και επικεντρωνόμαστε στους πράκτορες που έχουν στόχο το φιλτράρισμα και την συλλογή πληροφοριών από το διαδίκτυο.

Στο **Κεφάλαιο 5**, *Το Σύστημα Webnaut*, παρουσιάζουμε την αρχιτεκτονική του συστήματος *Webnaut* και περιγράφουμε αναλυτικά τις τεχνικές λεπτομέρειες που διέπουν την σχεδίαση, το πλαίσιο λειτουργίας, καθώς και την υλοποίηση των επιμέρους συστημάτων. Όλοι οι πράκτορες του συστήματος συνεργάζονται αρμονικά μεταξύ τους με σκοπό την ευχάριστη πλοήγηση του χρήστη στο διαδίκτυο και την υποστήριξη του σε μια σειρά από χρονοβόρες και κουραστικές εργασίες.

Στο **Κεφάλαιο 6**, *Η Λειτουργία της Προσαρμοζόμενης Εκμάθησης των Ενδιαφερόντων του Χρήστη*, παρουσιάζουμε τον πράκτορα εκμάθησης, ο οποίος οργανώνει και διευθύνει όλη την διαδικασία για την εκμάθηση των ενδιαφερόντων του χρήστη, καθώς αυτά εξελίσσονται στο χρόνο. Επίσης, περιγράφουμε αναλυτικά την μεθοδολογία της αναπαράστασης, επεξεργασίας και εκμάθησης των ενδιαφερόντων του χρήστη, με την χρήση ενός γενετικού αλγορίθμου δύο επιπέδων. Τέλος, παρουσιάζουμε τη λειτουργία της ανάδρασης του χρήστη στις υποδείξεις του συστήματος.

Στο **Κεφάλαιο 7**, *Αξιολόγηση του Συστήματος*, γίνεται παρουσίαση και ανάλυση των πειραματικών αποτελεσμάτων τα οποία αποκομίσαμε από την χρήση του συστήματος *Webnaut*. Τα συγκεκριμένα στοιχεία προέκυψαν κατά τη διάρκεια πειραμάτων τα οποία είχαν σαν στόχο την συλλογή και το φιλτράρισμα πληροφοριών από το διαδίκτυο, σύμφωνα με προκαθορισμένα προφίλ χρηστών.

Στο **Κεφάλαιο 8**, *Συμπεράσματα – Μελλοντικές Επεκτάσεις*, παραθέτουμε τα συμπεράσματα από την εργασία μας και περιγράφουμε την ερευνητική κατεύθυνση των μελλοντικών εργασιών.

Στο **Παράρτημα Α** αναφέρουμε όλες τις κλάσεις και τα πακέτα της επίσημης γλώσσας Java που χρησιμοποιήσαμε για την ανάπτυξη του συστήματος Webnaut.
Στο **Παράρτημα Β** αναφέρουμε όλες τις κλάσεις και τις μεθόδους που έχουμε σχεδιάσει και αποτελούν το σύστημα *Webnaut*.
Στο **Παράρτημα Γ** παραθέτουμε όλες τις ηλεκτρονικές διευθύνσεις που αναφέρονται στην διατριβή.

Τέλος, περιλαμβάνεται πλήρης βιβλιογραφική αναφορά των δημοσιεύσεων και βιβλίων που χρησιμοποιήθηκαν ως υπόβαθρο στην παρούσα Διατριβή.

1.4 Συνεισφορά στην Ερευνητική Περιοχή

Οι πρωτότυπες λύσεις που παρουσιάζονται στην Διδακτορική Διατριβή και οι οποίες αποτελούν την συνεισφορά της στην επιστήμη, είναι :

- Η ανάπτυξη μεθοδολογίας για την αναπαράσταση και την εκμάθηση των ενδιαφερόντων ενός χρήστη, με την βοήθεια ενός γενετικού αλγορίθμου δύο επίπεδων.

- Η ανάπτυξη μεθοδολογίας για την σχεδίαση ευφυών συστημάτων τα οποία αποτελούνται από πράκτορες λογισμικού, οι οποίοι συνεργάζονται μεταξύ τους για την συλλογή των πληροφοριών από το διαδίκτυο.

- Η ανάπτυξη ενός μοντέλου αλληλεπίδρασης του χρήστη, κατά την διάρκεια της πλοήγησης του στο διαδίκτυο, με τους πράκτορες λογισμικού.

Το ευφυές σύστημα Webnaut, υλοποιεί την μεθοδολογία για την αναπαράσταση και την εκμάθηση των ενδιαφερόντων ενός χρήστη, με την βοήθεια ενός γενετικού αλγορίθμου δύο επίπεδων. Η συγκεκριμένη μεθοδολογία είναι αρκετά γενική ώστε να μπορεί να χρησιμοποιηθεί σε διαφορετικά γνωστικά πεδία, επιτρέποντας μας ταυτόχρονα να δημιουργούμε σύνθετες δομές αναπαράστασης των μακροχρόνιων ενδιαφερόντων του χρήστη. Το σύστημα πραγματοποιεί την συλλογή πληροφοριών, σύμφωνα με ένα διαρκώς προσαρμοζόμενο προφίλ του χρήστη, λαμβάνοντας υπόψη του διάφορες παραμέτρους, όπως :

α) Τα παραδείγματα που παραθέτει ο χρήστης κατά την διάρκεια της πλοήγησης του στο διαδίκτυο.

β) Τα ενδιαφέροντα του χρήστη καθώς αυτά εξελίσσονται μέσα στο χρόνο.

γ) Την απόκριση του χρήστη στις υποδείξεις του συστήματος.

Επίσης, το σύστημα αποτελείται από πολλούς πράκτορες λογισμικού, οι οποίοι προσφέρουν διαφορετικές υπηρεσίες και όλοι μαζί συνεργάζονται για την εκμάθηση των ενδιαφερόντων του χρήστη αλλά και για την συλλογή των πληροφοριών από το διαδίκτυο.

1.5 Ευχαριστίες

Από την θέση αυτή θα ήθελα να εκφράσω τις ευχαριστίες μου στο Πανεπιστήμιο του Πειραιά και στο Τμήμα Πληροφορικής για την ευκαιρία που μου έδωσαν να εκπονήσω την παρούσα Διατριβή.

Ειδικότερα, θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα Επ. Καθηγητή κ. Θ. Παναγιωτόπουλο για την ευκαιρία που μου έδωσε να ασχοληθώ ερευνητικά με το θέμα των πρακτόρων λογισμικού στο διαδίκτυο, την καθοδήγηση καθώς και για την συμπαράσταση κατά την διάρκεια της διατριβής. Επίσης θα ήθελα να ευχαριστήσω τα μέλη της τριμελούς επιτροπής, Καθηγητή κ. Ν. Αλεξανδρή και Αν. Καθηγητή κ. Δ. Δεσπότη για την πολύπλευρη υποστήριξη τους καθώς και για τις εύστοχες παρατηρήσεις τους. Τέλος θα ήθελα να ευχαριστήσω όλα τα μέλη, την γραμματεία καθώς και όλους τους μεταπτυχιακούς φοιτητές, του Τμήματος Πληροφορικής του Πανεπιστημίου Πειραιώς για όλα τα ευχάριστα και δημιουργικά χρόνια που περάσαμε μαζί και τους εύχομαι κάθε επιτυχία στην ζωή τους.

Νικόλαος Ζ. Ζάχαρης

Πειραιάς, Ιούνιος 2002

Κεφάλαιο 2

Η ΑΝΑΠΤΥΞΗ ΤΟΥ ΔΙΑΔΙΚΤΥΟΥ

2.1 Εισαγωγή

Τα τελευταία χρόνια προκαλεί μεγάλη εντύπωση η ανάπτυξη του διαδικτύου καθώς επίσης και η διεισδυτικότητα του σε όλα τα επίπεδα της καθημερινής μας ζωής. Ολοένα και περισσότεροι είναι οι άνθρωποι που χρησιμοποιούν το διαδίκτυο είτε για την εύρεση του προγράμματος ενός θεάτρου είτε για τις τραπεζικές τους συναλλαγές είτε ακόμα και για τις καθημερινές τους αγορές που μπορεί να περιλαμβάνουν βιβλία, Cd's, τις αγορές τους σε πολυκαταστήματα μέχρι αεροπορικά εισιτήρια. Σύμφωνα με διεθνείς οργανισμούς [Γ22, Γ28] ο αριθμός των χρηστών της Ευρωπαϊκής Ένωσης που έκαναν χρήση των υπηρεσιών του διαδικτύου το έτος 2000 ήταν 119 εκατομμύρια και αναμένεται αυτός ο αριθμός να αυξηθεί κατά 55% έως το έτος 2003.

Τα ακαδημαϊκά και ερευνητικά ιδρύματα δεν μονοπωλούν πλέον την χρήση του διαδικτύου αλλά όλες σχεδόν οι επιχειρήσεις και οργανισμοί έχουν συμμετοχή σε αυτό που ονομάζουμε παγκόσμιο ιστό πληροφοριών (World Wide Web). Καθημερινά νέοι εξυπηρετητές συνδέονται σε αυτό το παγκόσμιο δίκτυο προσφέροντας είτε νέες υπηρεσίες είτε απλά κάνουν αισθητή την παρουσία τους στο *παγκόσμιο χωριό* παρέχοντας ενημέρωση και πληροφορίες σχετικές με τις ίδιες τις εταιρίες και τα προϊόντα τους.

Σήμερα ο οποιοσδήποτε μπορεί να έχει πρόσβαση στο διαδίκτυο ακόμα και τις ώρες που δεν βρίσκεται στην εργασία του, αφού η σύνδεση μέσω του τηλεφωνικού δικτύου είναι μια αρκετά εύκολη υπόθεση λόγω του ότι τα πρωτόκολλα επικοινωνίας θεωρούνται πλέον *ώριμη* τεχνολογία. Όλα τα λειτουργικά συστήματα παρέχουν υποστήριξη για οποιαδήποτε μέθοδο σύνδεσης του προσωπικού μας υπολογιστή με το διαδίκτυο, από απλές τηλεφωνικές συνδέσεις μέχρι σύνδεση μέσω δορυφόρου.

Επίσης σε αρκετές χώρες που θεωρούν την επέκταση του διαδικτύου ως στοιχείο ανάπτυξης της οικονομίας τους, το κόστος της σύνδεσης στο διαδίκτυο μέσω τηλεφώνου είτε δεν χρεώνεται είτε είναι αρκετά χαμηλό έτσι ώστε η επιλογή του παροχέα σύνδεσης να εξαρτάται μόνο από τις υπηρεσίες που προσφέρει καθώς και

από την ποιότητα τους. Σήμερα σχεδόν όλες οι εταιρίες που παρέχουν πρόσβαση στο διαδίκτυο εκτός των άλλων υπηρεσιών τους όπως το ηλεκτρονικό ταχυδρομείο, παρέχουν ως *δώρο* στους πελάτες τους και ικανό αποθηκευτικό χώρο για να αναπτύξουν την *προσωπική* τους σελίδα.

Η συγγραφή μιας προσωπικής σελίδας για το διαδίκτυο και η τοποθέτηση της στον εξυπηρετητή του παροχέα μας είτε σε άλλον εξυπηρετητή που έχουμε πρόσβαση μπορεί να γίνει είτε μέσω ενός από τα εξειδικευμένα εργαλεία με τον εποπτικό τρόπο είτε με τον *παραδοσιακό* τρόπο δηλαδή, να κάνουμε χρήση ενός οποιουδήποτε κειμενογράφου για την συγγραφή της και εν συνεχεία μέσω του πρωτοκόλλου μεταφοράς αρχείων FTP [Γ3] να *ανεβάσουμε* τις σελίδες από τον δίσκο του προσωπικού μας υπολογιστή στον δίσκο του εξυπηρετητή για το διαδίκτυο. Η ανάπτυξη και γιγάντωση του παγκοσμίου ιστού θα μπορούσαμε συνοπτικά να αναφέρουμε ότι οφείλεται κυρίως :

1. Στο βασικό χαρακτηριστικό του διαδικτύου που είναι η απρόσκοπτη και επιτυχής λειτουργία του μέσω ενός μεγάλου αριθμού διαφορετικών δικτύων, υπολογιστών και εφαρμογών.

2. Στη χρήση του υπερκειμένου ως βασικού στοιχείου περιγραφής και παρουσίασης των πληροφοριών.

Δεν είναι μόνο οι απλοί χρήστες που συνδέονται στο διαδίκτυο και αναπτύσσουν σελίδες που περιγράφουν τα προσωπικά ενδιαφέροντα τους αλλά είναι και οι εταιρίες που έχουν επιδοθεί σε έναν αγώνα προβολής στον παγκόσμιο ιστό. Αυτοί οι εξυπηρετητές, εκτός των πληροφοριών που προσφέρουν με ιστορικά και οικονομικά στοιχεία για την εταιρία, προσφέρουν τεχνικές πληροφορίες αναφορικά με τα προϊόντα τους, υποστήριξη για τυχόν προβλήματα, αναβαθμίσεις και αρκετές φορές άμεσες παραγγελίες ή/και αγορές με την χρήση της πιστωτικής κάρτας του πελάτη.

Σχεδόν όλες οι επιχειρήσεις - οργανισμοί μεταφέρουν τις παλαιές τους εφαρμογές σε νέες πλατφόρμες υλικού-λογισμικού ή αναπτύσσουν διεπαφές ώστε να έχουν σύνδεση με το διαδίκτυο. Επίσης όλες οι εταιρίες αναπτύσσουν με εκθετικούς ρυθμούς και το εσωτερικό τους δίκτυο με σκοπό να υποστηρίξουν τόσο την οργανωτική τους δομή όσο και τους ίδιους τους εργαζομένους τους. Εταιρίες με παγκόσμια παρουσία, για παράδειγμα του μεγέθους της Xerox [Rei97], έχουν πάνω

από 200 εξυπηρετητές για την υποστήριξη τόσο της οργάνωσης της επιχείρησης όσο και των 20000 εργαζομένων της. Το πρόγραμμα πλοήγησης για το διαδίκτυο χρησιμοποιείται ως βασική διεπαφή και στο εσωτερικό δίκτυο αφού η λειτουργία των εφαρμογών της επιχείρησης αλλά και η πρόσβαση σε αυτές βασίζεται στα ίδια πρωτόκολλα τόσο σε επίπεδο επικοινωνίας όσο και σε επίπεδο εφαρμογών.

2.2 Το Πρωτόκολλο Επικοινωνίας TCP/IP

Το διαδίκτυο ξεκίνησε ως ένα στρατιωτικό ερευνητικό πρόγραμμα Advanced Research Projects Agency Network [Γ19] στη δεκαετία του 1960 και εξελίχθηκε στο τεράστιο δίκτυο που όλοι σήμερα γνωρίζουμε και έχουμε σε αυτό ελεύθερη πρόσβαση. Ο αρχικός στόχος ήταν να δημιουργηθεί ένα κατανεμημένο και αποκεντρωμένο δίκτυο συλλογής και αποστολής στρατιωτικών πληροφοριών, που θα ήταν ικανό να μεταφέρει πληροφορίες ακόμη και στην περίπτωση που ένας ή περισσότεροι σταθμοί επικοινωνίας παρουσιάσουν βλάβη ή δεχτούν επίθεση από τον εχθρό.

Το TCP/IP [Stev94,Wri95,Stev96] είναι ένα σύνολο από πρωτόκολλα που αναπτύχθηκαν με σκοπό να υποστηρίξουν την μεταφορά και διευθυνσιοδότηση αυτού του είδους των επικοινωνιών. Το ίδιο πρωτόκολλο χρησιμοποιείται και σήμερα στο διαδίκτυο για την επίτευξη της αποστολής και λήψης των δεδομένων, ανεξαρτήτως του τρόπου σύνδεσης μας με το διαδίκτυο. Η ευελιξία του πρωτοκόλλου οφείλεται στο ότι έχει χτιστεί σε πολλαπλά επίπεδα και το καθένα από τα επίπεδα ελέγχει ένα μέρος της επικοινωνίας.

Τα δεδομένα ξεκινούν από το πρώτο επίπεδο, της *Εφαρμογής*, δηλαδή από την εφαρμογή που τα αποστέλλει και πρέπει να περάσουν από τα υπόλοιπα επίπεδα μέχρι να φτάσουν στη φυσική σύνδεση. Στα επόμενα στάδια, το επίπεδο *Παρουσίασης* μετατρέπει τα δεδομένα ώστε να υπάρχει μια καθιερωμένη διεπαφή επικοινωνίας μεταξύ των εφαρμογών - η συμπίεση και η κρυπτογράφηση είναι οι συνήθεις εργασίες σε αυτό το επίπεδο, ενώ η δημιουργία, ο έλεγχος και ο τερματισμός της επικοινωνίας μεταξύ δύο ή περισσοτέρων εφαρμογών που συνεργάζονται για ένα κοινό σκοπό, ελέγχονται από το επίπεδο της *Συνόδου*.

Το επίπεδο *Μεταφοράς* φροντίζει για την ροή των δεδομένων ανάμεσα στους υπολογιστές των δύο δικτύων, ενώ το επίπεδο *Δικτύου* ασχολείται με το δρομολόγιο που πρέπει να ακολουθήσει ένα πακέτο δεδομένων ώστε να φτάσει από τον

υπολογιστή Α στον υπολογιστή Β. Επίσης παρέχει μια μέθοδο αναγνώρισης των υπολογιστών με μοναδιαίο τρόπο. Το προτελευταίο επίπεδο της *Σύνδεσης των Δεδομένων* στέλνει και παραλαμβάνει τα δεδομένα από το φυσικό μέσο που έχει επιλεγεί για την μεταφορά ενώ το τελευταίο επίπεδο της *Φυσικής Σύνδεσης* ελέγχει ηλεκτρικές και μηχανικές λειτουργίες για τον έλεγχο των φυσικών συνδέσεων.

Σε αυτή την αλυσίδα των επιπέδων, η πληροφορία ρέει με τέτοιο τρόπο ώστε κατά την μεταφορά, το κάθε επίπεδο προσθέτει πληροφορίες που είναι απαραίτητες για την μετακίνηση των δεδομένων και κατά την λήψη, το κάθε επίπεδο αφαιρεί πληροφορίες που δεν χρειάζονται στο ανώτερο επίπεδο, ώστε στο τελικό επίπεδο της εφαρμογής να φτάνουν "*καθαρά*" δεδομένα.

Το TCP/IP χρησιμοποιεί δύο πρωτόκολλα μεταφοράς : το TCP (Transmission Control Protocol) [Γ20] και το UDP (User Datagram Protocol) [Γ21]. Το TCP είναι αξιόπιστο κανάλι μεταφοράς, αφού τα μεγάλα πακέτα δεδομένων αναλαμβάνει να τα χωρίσει σε μικρότερα πακέτα, τα οποία είναι κατάλληλα για το φυσικό μέσο μεταφοράς που θα χρησιμοποιηθεί και επιπλέον εγγυάται ότι η πληροφορία που αποστέλλεται από το ένα άκρο θα παραληφθεί από το άλλο άκρο της σύνδεσης.

Το πρωτόκολλο επικοινωνίας TCP χρησιμοποιείται από πάρα πολλά προγράμματα δικτύου και είναι η βάση σε πολλά πρωτόκολλα επιπέδου εφαρμογής, όπως για παράδειγμα το HTTP (HyperText Transfer Protocol) [Γ1], το FTP (File Transfer Protocol) [Γ3] κ.λπ. Από την άλλη πλευρά το UDP δεν εγγυάται την λήψη της πληροφορίας και αφήνει όλο τον έλεγχο και τον εντοπισμό σφαλμάτων στην εφαρμογή. Χρησιμοποιείται κυρίως για την μεταφορά δεδομένων μικρής αξίας, όπως για παράδειγμα στην υπηρεσία DNS (Domain Name Service) [Γ23] ή μερικές εφαρμογές συνεχούς ροής οπτικοακουστικού υλικού.

Αλλαγές, όσο αφορά το πρωτόκολλο έχουν γίνει στα υψηλά επίπεδα της δομής του με σκοπό να ενσωματώσουν νέους τρόπους επικοινωνίας στο επίπεδο των εφαρμογών. Οι εφαρμογές στο διαδίκτυο χωρίζονται σε πελάτες και εξυπηρετητές και η διάκριση μεταξύ τους είναι ποίος στέλνει μια αίτηση και ποίος θα εξυπηρετήσει την αίτηση. Η αρχικοποίηση της σύνδεσης, η μεθοδολογία της αίτησης και της ανταλλαγής των δεδομένων, καθώς και ο τερματισμός της σύνδεσης μεταξύ δύο εφαρμογών, καθορίζεται πάλι από πρωτόκολλα μεταφοράς δεδομένων αλλά πλέον σε επίπεδο εφαρμογής.

2.3 Το Πρωτόκολλο Επικοινωνίας HTTP

Στο διαδίκτυο υπάρχουν αρκετά πρωτόκολλα επικοινωνίας στο επίπεδο των εφαρμογών, τα οποία μπορούμε να τα θεωρήσουμε ως ένα σύνολο από κανόνες που πρέπει να ακολουθήσουν ο πελάτης και ο εξυπηρετητής, ώστε να μπορέσουν με επιτυχία να επικοινωνήσουν μεταξύ τους. Το σχήμα της επικοινωνίας μεταξύ ενός πελάτη και ενός εξυπηρετητή, μπορεί να είναι σχετικά απλό, δηλαδή μια αίτηση που συνοδεύεται από μια απόκριση ή μπορεί να είναι ένα αρκετά σύνθετο σχήμα, όπου περιλαμβάνει θέματα ασφαλείας και πιστοποίησης του πελάτη και του εξυπηρετητή.

Το HTTP θεωρείται η ραχοκοκαλιά του παγκοσμίου ιστού, παρ' ότι, σε σχέση με τα άλλα πρωτόκολλα του διαδικτύου, είναι αρκετά καινούργιο. Το πρωτόκολλο αναπτύχθηκε το 1990-91 από το Tim Berners-Lee, στα εργαστήρια του Ευρωπαϊκού Κέντρου Φυσικής, CERN, στη Γενεύη και σύμφωνα με το δημιουργό του, είναι αρκετά ελαφρύ και γρήγορο ώστε να μπορεί να χρησιμοποιηθεί σε κατανεμημένες, συνεργαζόμενες εφαρμογές υπερκειμένων. Το HTTP ακολουθεί το γενικό σχήμα επικοινωνίας αίτησης/απόκρισης πάνω από το πρωτόκολλο TCP, ώστε να είναι εγγυημένη η μεταφορά και λήψη των δεδομένων. Έτσι λοιπόν, σύμφωνα με το πρωτόκολλο, ο πελάτης - το πρόγραμμα που απευθύνει την αίτηση, αφού συνδεθεί με τον εξυπηρετητή του απευθύνει μια αίτηση που έχει την παρακάτω μορφή :

[Μέθοδος Αίτησης] [Πόρος] [Έκδοση του Πρωτοκόλλου]

Η αίτηση μπορεί να συνοδεύεται από διάφορα στοιχεία και πληροφορίες για τον πελάτη, όπως για παράδειγμα, τι είδους αποκρίσεις μπορεί να χειριστεί ο πελάτης, καθώς επίσης μπορεί να περιέχει και άλλα στοιχεία που πρέπει να χρησιμοποιήσει ο εξυπηρετητής ώστε να ανταποκριθεί στην αίτηση. Ο χειρισμός της αίτησης από την πλευρά του εξυπηρετητή διαφοροποιείται ανάλογα με την μέθοδο αίτησης του πελάτη. Για παράδειγμα, μια αίτηση με την μέθοδο GET [Γ1], σημαίνει την αποστολή του πόρου, ενώ η ίδια αίτηση με την μέθοδο HEAD [Γ1], σημαίνει την αποστολή μόνο πληροφοριών σχετικών με τον πόρο, όπως για παράδειγμα, την ημερομηνία της τελευταίας τροποποίησης του πόρου, ώστε να αποφεύγεται η αποστολή των δεδομένων τη στιγμή που δεν έχουν τροποποιηθεί από την προηγούμενη αίτηση.

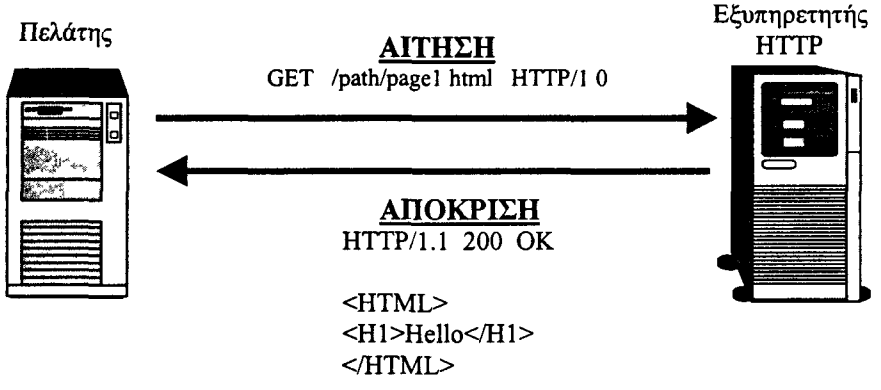
Ο πόρος [Γ8] δείχνει το πλήρες μονοπάτι του αντικειμένου που εμπεριέχεται στην αίτηση σε σχέση πάντα με τη ρίζα των πόρων του εξυπηρετητή. Γενικά, ένας πόρος στο διαδίκτυο αναπαριστάται αρκετά εύκολα, αφού περιγράφεται με τον τρόπο

με τον οποίο μπορούμε να τον προσεγγίσουμε, δηλαδή έχει την παρακάτω μορφή :

Πρωτόκολλο://Εξυπηρετητής.Δικτυακός_Τόπος:Πόρτα_Επικοινωνίας/Μονοπάτι_Πόρου

π.χ. http://www.microsoft.com:80/windows2000/index.html

Η γενική περιγραφή του πόρου αρχίζει με το πρωτόκολλο που πρέπει να χρησιμοποιήσουμε (http), εν συνεχεία περιγράφεται το μηχάνημα εξυπηρέτησης (www), ο δικτυακός τόπος που ανήκει το μηχάνημα εξυπηρέτησης (microsoft.com), η πόρτα επικοινωνίας που περιμένει ο εξυπηρετητής για να ικανοποιήσει τις συνδέσεις των πελατών (80) και τέλος το μονοπάτι του πόρου (windows2000/index.html).



Σχήμα 2.1 Τυπική συναλλαγή πελάτη - εξυπηρετητή μέσω του πρωτοκόλλου HTTP

Ο εξυπηρετητής, αφού παραλάβει την αίτηση του χρήστη, απαντά αρχίζοντας με μια γραμμή κατάστασης, η οποία περιλαμβάνει την έκδοση του πρωτοκόλλου και ένα κωδικό, που δείχνει είτε επιτυχία, είτε αποτυχία/σφάλμα, είτε ανακατεύθυνση σε άλλο πόρο. Η έκδοση του πρωτοκόλλου είναι χρήσιμη ως αναφορά ώστε να αποφεύγονται και από τις δύο πλευρές (πελάτης - εξυπηρετητής) οι εντολές που δεν είναι κοινά αποδεκτές. Το σχήμα 2.1 δείχνει μια τυπική συναλλαγή μεταξύ ενός πελάτη και ενός εξυπηρετητή μέσω του πρωτοκόλλου HTTP.

Η απόκριση του εξυπηρετητή, μετά την γραμμή κατάστασης, συνεχίζει με άλλες πληροφορίες, που θα βοηθήσουν τον πελάτη να χειριστεί την απάντηση του εξυπηρετητή. Για παράδειγμα, το μέγεθος των πληροφοριών που αποστέλλονται, τον τύπο της απόκρισης, δηλαδή, αν πρόκειται για ένα κείμενο ή μια φωτογραφία, την ημερομηνία της τελευταίας τροποποίησης κ.λπ.

Το πρωτόκολλο HTTP είναι αρκετά γρήγορο αφού δεν διαθέτει "*μνήμη*", δηλαδή, δεν λαμβάνει υπόψη του την κατάσταση της σύνδεσης του πελάτη. Με την

ολοκλήρωση της αποστολής της απάντησης του εξυπηρετητή, διακόπτεται και η σύνδεση με τον πελάτη. Κατ' αυτόν τον τρόπο όμως δημιουργείται μεγάλος φόρτος στο δίκτυο, γιατί απαιτείται μια νέα σύνδεση, για κάθε πόρο που αιτείται ο ίδιος πελάτης.

Ένας εξυπηρετητής στο διαδίκτυο, εξυπηρετεί στατικά αρχεία, αφήνοντας τις χρονοβόρες και αργές διαδικασίες στις εφαρμογές που υλοποιούν το πρωτόκολλο CGI (Common Gateway Interface) [Γ7]. Όλοι οι εξυπηρετητές υποστηρίζουν τα CGI προγράμματα, τα οποία τα τρέχουν ως ανεξάρτητες λειτουργίες και μπορούν να υλοποιηθούν σε οποιαδήποτε γλώσσα προγραμματισμού. Το CGI πρωτόκολλο καθορίζει, για ένα οποιοδήποτε πρόγραμμα, τον τρόπο συλλογής όλων των πληροφοριών που είναι σχετικές με την αίτηση του πελάτη και κατόπιν την αποστολή της απόκρισης του προγράμματος, σύμφωνα με το πρωτόκολλο επικοινωνίας.

Τα CGI προγράμματα, παρ' ότι θεωρούνται απαρχαιωμένα σε σχέση με τις γλώσσες προγραμματισμού που υποστηρίζουν και εκτελούν εγγενώς οι εξυπηρετητές π.χ. VbScript [Γ25], JavaScript [Γ10] JSP [Γ27] κ.λπ., είναι ο μοναδικός τρόπος διασύνδεσης παλαιών εφαρμογών με το διαδίκτυο.

2.4 Η Γλώσσα HTML και το Πρόγραμμα Πλοήγησης

Το πρωτόκολλο HTTP δημιουργήθηκε για την υποστήριξη υπερκειμένων, τα οποία, πολύ απλά, μπορούμε να ορίσουμε ως κείμενα που περιέχουν συνδέσμους που οδηγούν σε άλλα κείμενα. Δηλαδή, πρόκειται για ένα σύνολο από κόμβους, που περιέχουν κείμενα, και από δείκτες, που είναι οι σύνδεσμοι που ενώνουν τους κόμβους. Έτσι λοιπόν, όταν ο αναγνώστης του κειμένου επιλέξει ένα σύνδεσμο τότε εμφανίζεται ένα νέο κείμενο το οποίο μπορεί να βρίσκεται στον ίδιο ή σε διαφορετικό εξυπηρετητή.

Συνήθως ο συγγραφέας ενός υπερκειμένου συνδέει κάποιες λέξεις με συνδέσμους, ώστε να παρέχει την δυνατότητα στον αναγνώστη του κειμένου να συλλέξει περισσότερες πληροφορίες για το θέμα που περιγράφει η συγκεκριμένη λέξη. Η μετακίνηση των αναγνωστών μέσα στο δίκτυο των κόμβων ονομάζεται πλοήγηση ή πιο απλά "*σερφάρισμα*".

Τα τελευταία χρόνια η έννοια του υπερκειμένου έχει επεκταθεί και σε άλλα μέσα, όπως για παράδειγμα σε οπτικοακουστικό υλικό ώστε πλέον να μιλάμε για εφαρμογές υπερμέσων. Η ανάλυση και σχεδίαση μιας εφαρμογής υπερμέσων για το διαδίκτυο

είναι μια αρκετά σύνθετη υπόθεση [Gar95, Kah95] αφού θα πρέπει να λάβουμε υπόψη τα παρακάτω :

Περιεχόμενα : Είναι τα στοιχεία που περιλαμβάνει η εφαρμογή, δηλαδή, το κείμενο, οι φωτογραφίες και τα υπόλοιπα μέσα, που χρησιμοποιούνται για την περιγραφή της συγκεκριμένης γνωστικής περιοχής.

Δομή : Ο τρόπος με τον οποίο ομαδοποιείται και οργανώνεται η πληροφορία.

Παρουσίαση : Πως παρουσιάζεται το περιεχόμενο της εφαρμογής, τι δυναμική έχει, καθώς και τις δυνατότητες αλληλεπίδρασης που έχει στην διάθεση του ο χρήστης.

Ένας από τους λόγους της γιγάντωσης του παγκοσμίου ιστού, εκτός από την χρήση του υπερκειμένου ως βασικό μέσο παρουσίασης των πληροφοριών, είναι και η ευκολία συγγραφής μιας σελίδας υπερκειμένου, με την βοήθεια της HTML [Γ12]. Η συγκεκριμένη λέξη, αναφέρεται τόσο στη γλώσσα που χρησιμοποιείται για την συγγραφή των υπερκειμένων και η οποία είναι υποσύνολο της γλώσσας SGML [Γ29], όσο και στην περιγραφή των κειμένων που χρησιμοποιούν την συγκεκριμένη μορφοποίηση.

Στην HTML χρησιμοποιούνται ετικέτες οι οποίες περικλείονται από τους χαρακτήρες < και >, όπως για παράδειγμα <ετικέτα>, και θεωρούνται οδηγίες σύνθεσης του κειμένου σε επικεφαλίδες, παραγράφους, λίστες, συνδέσμους κ.λπ. Τα HTML κείμενα είναι ASCII χαρακτήρες έτσι ώστε για την συγγραφή τους να επιτρέπεται η χρήση ενός απλού κειμενογράφου ή η χρήση ενός εξειδικευμένου εποπτικού εργαλείου. Επιπλέον, η συγκεκριμένη μορφή καθιστά αρκετά εύκολη υπόθεση την μετάφραση κειμένων από άλλες μορφοποιήσεις, για αυτό το λόγο η πλειοψηφία των κειμενογράφων αλλά και άλλων εφαρμογών π.χ. λογιστικά φύλλα, εφαρμογές παρουσιάσεων, υποστηρίζουν την αποθήκευση σε μορφή HTML.

Οι ετικέτες, εκτός ελαχίστων εξαιρέσεων, εμφανίζονται σε ζευγάρια, δηλαδή, μία ετικέτα έναρξης δηλώνει την αρχικοποίηση της διαμόρφωσης και εν συνεχεία η ίδια ετικέτα, με την προσθήκη του χαρακτήρα / σημαίνει το τέλος της διαμόρφωσης. Το ίδιο ισχύει και για ολόκληρο το κείμενο, το οποίο περικλείεται μέσα στις ετικέτες <HTML> και </HTML>. Με την χρήση των ετικετών, ο συγγραφέας μπορεί να διαμορφώσει ένα ολόκληρο κείμενο και να εισαγάγει, αν φυσικά το επιθυμεί, τους

απαραίτητους συνδέσμους, εικόνες κ.λπ.

Η HTML δεν είναι μόνο μια γλώσσα περιγραφής κειμένων αλλά επιπλέον επιτρέπει την δημιουργία ποικίλων δυναμικών διεπαφών, σε αρκετές διαδικτυακές-εσωτερικού δικτύου εφαρμογές [Lev98]. Η αλληλοεπίδραση του χρήστη με την εφαρμογή υποστηρίζεται από Java Applets [Γ9], HTML ετικέτες φόρμας [Γ12.b], εφαρμογές ActiveX [Γ30], γλώσσες σεναρίων για το πρόγραμμα πλοήγησης [Γ10, Γ25] ή συνδυασμό των προηγουμένων.

Η γλώσσα HTML δεν περιλαμβάνει οδηγίες επεξεργασίας για την ακριβή εμφάνιση των περιεχομένων αλλά αφήνει αυτό το θέμα στο πρόγραμμα που θα χρησιμοποιηθεί για την εμφάνιση του υπερκειμένου. Έτσι λοιπόν, τα ίδια περιεχόμενα σε διαφορετικά προγράμματα απεικόνισης υπερκειμένου θα εμφανίζονται διαφορετικά. Αυτή η υλοποίηση δημιουργεί ορισμένα προβλήματα, ως προς την τελική απεικόνιση του υπερκειμένου, αλλά από την άλλη πλευρά, αποφεύγει προβλήματα, σχετικά με τις ετικέτες που δεν υποστηρίζονται, με τα διαφορετικά μεγέθη χαρτιού καθώς και με τις γραμματοσειρές που μπορεί να απουσιάζουν από τον υπολογιστή του αναγνώστη.

Το πρόγραμμα πλοήγησης για το διαδίκτυο (Web Browser) ή φυλλομετρητής είναι ένα λογισμικό, το οποίο γνωρίζει πως, να επικοινωνήσει με ένα εξυπηρετητή μέσω του πρωτοκόλλου HTTP, να ζητήσει ένα αρχείο και εν συνεχεία, είτε να το εμφανίσει στην οθόνη του υπολογιστή, είτε να το αποθηκεύσει στο τοπικό δίσκο, είτε τέλος να καλέσει την εφαρμογή που μπορεί να χειριστεί το συγκεκριμένο τύπο αρχείου. Το πρόγραμμα πλοήγησης γνωρίζει από την απόκριση του HTTP εξυπηρετητή αν πρόκειται για ένα κείμενο σε μορφή HTML, οπότε θα αναλύσει τις ετικέτες του κειμένου και θα συλλέξει όλες τις εικόνες ή άλλα μέσα που εμπεριέχονται σε αυτό, ώστε να το εμφανίσει σε μια ολοκληρωμένη μορφή στην οθόνη του υπολογιστή.

Η πλειοψηφία των χρηστών χρησιμοποιεί για την πλοήγηση στο διαδίκτυο τα προγράμματα Microsoft Internet Explorer [Γ26] ή/και Netscape Communicator [Γ24]. Και τα δύο προγράμματα διατίθενται δωρεάν και ενσωματώνουν επιπλέον δυνατότητες όπως για παράδειγμα διαχείριση ηλεκτρονικού ταχυδρομείου, διευθυνσιογράφο κ.λπ. Σε κάθε νέα έκδοση, και τα δύο προγράμματα, προσθέτουν νέες εντολές και δυνατότητες στην διαχείριση των περιεχομένων του διαδικτύου, με αποτέλεσμα οι προδιαγραφές της γλώσσας να επιδέχονται συνεχώς βελτιώσεις.

2.5 Προβλήματα στην Οργάνωση του Διαδικτύου

Η χρήση του παγκοσμίου ιστού ως κύριο μέσο παρουσίασης πληροφοριών δεν είναι χωρίς προβλήματα. Αυτά σχετίζονται κυρίως με την οργάνωση, διαχείριση και συντήρηση της πληροφορίας [The98]. Όλοι σχεδόν οι χρήστες του διαδικτύου γνωρίζουν τα προβλήματα τα οποία μπορούμε να συνοψίσουμε στα παρακάτω :

α) Το περιεχόμενο στις σελίδες του διαδικτύου αλλάζει χωρίς καμία οργάνωση, σχεδόν αυθαίρετα. Για παράδειγμα, κανείς δεν μας εγγυάται ότι η σελίδα που έχουμε μπροστά μας στο πρόγραμμα πλοήγησης κατά την επίσκεψη μας σε ένα κόμβο του διαδικτύου, θα είναι διαθέσιμη και την αμέσως επόμενη χρονική στιγμή. Αυτή η πρακτική δεν θα πρέπει να εκπλήσσει κανέναν, αφού τις περισσότερες φορές αν όχι όλες, ακόμα και ειδικού ενδιαφέροντος δικτυακοί κόμβοι που ζητούν την διεύθυνση του χρήστη για ηλεκτρονική αλληλογραφία, ώστε να τον ενημερώσουν σε περίπτωση αλλαγής των περιεχομένων του κόμβου, δεν παρέχουν καμία ενημέρωση για την εξαφάνιση, αλλαγή ή αντικατάσταση σε μία ή περισσότερες από τις σελίδες τους.

β) Λίγη ή καθόλου βοήθεια παρέχεται για την διαχείριση μιας σελίδας. Για παράδειγμα, σε αρκετές περιπτώσεις δεν αναφέρεται πότε άρχισε η υποστήριξη της σελίδας, πότε ενημερώθηκε για τελευταία φορά και φυσικά αν έχει σταματήσει και πότε η υποστήριξη μιας σελίδας. Επιπλέον, σε αρκετούς δικτυακούς τόπους, που είτε βρίσκονται στο στάδιο της κατασκευής ή του επανασχεδιασμού, είναι αρκετά κοινή πρακτική να εμφανίζεται μια μόνο σελίδα με την ένδειξη "Υπό Κατασκευή", χωρίς καμία επισήμανση για το πότε θα έχουν ολοκληρωθεί οι εργασίες.

Συνέπεια όλων των προηγουμένων είναι, να επισκέπτεται ο χρήστης πάλι και πάλι ένα δικτυακό κόμβο, ώστε να αντλήσει πληροφορίες και μάλιστα χωρίς καμία ένδειξη για την κατάσταση τους ή την ηλικία τους.

γ) Είναι αρκετά δύσκολη και χρονοβόρα η διαδικασία της συντήρησης των συνδέσμων που αναφέρονται σε μία σελίδα και κατά συνέπεια, μπορεί να δείχνουν σε ανύπαρκτες ή σε λάθος σελίδες. Λόγω της δυναμικής φύσης του διαδικτύου, είναι αρκετά συνηθισμένο φαινόμενο, σύνδεσμοι, που αναφέρονται ακόμα σε διαφορετικά τμήματα της ίδιας εταιρίας, να μην είναι ενημερωμένοι.

δ) Η διαθέσιμη πληροφορία είναι ετερογενής, δηλαδή εμφανίζεται σε διαφορετικές

μορφές και τύπους αρχείων π.χ. κείμενα, εικόνες, προγράμματα κ.λπ. Συνεχώς εμφανίζονται νέοι τύποι αρχείων, οι οποίοι συνοδεύονται από ειδικό λογισμικό, το οποίο μπορεί να ενσωματωθεί στο πρόγραμμα πλοήγησης για το διαδίκτυο.

Όλα τα παραπάνω προβλήματα, μαζί με την απουσία μιας αρχής-οργανισμού, που σκοπό θα έχει τον έλεγχο των περιεχομένων του διαδικτύου, με στόχο την δημιουργία ενός κεντρικού καταλόγου περιεχομένων, δημιουργούν μεγάλα προβλήματα στην έρευνα ενός χρήστη προς αναζήτηση πληροφοριών.

2.6 Συμπεράσματα

Στο κεφάλαιο αυτό πραγματοποιήθηκε μια σύντομη επισκόπηση του παγκοσμίου ιστού, ο οποίος αποτελεί και το περιβάλλον δραστηριοποίησης και συλλογής πληροφοριών του συστήματος Webnaut. Πιο συγκεκριμένα, έγινε παρουσίαση των πρωτοκόλλων επικοινωνίας τόσο σε επίπεδο δικτύου όσο και σε επίπεδο εφαρμογών, της γλώσσας HTML, του προγράμματος πλοήγησης για το διαδίκτυο.

Ο όγκος των πληροφοριών που παρουσιάζονται στο διαδίκτυο συνεχώς αυξάνει λόγω της παγκόσμιας αποδοχής του ως πλατφόρμας ανάκτησης και διάδοσης των πληροφοριών, ηλεκτρονικού εμπορίου και διασκέδασης. Στο φαινόμενο αυτό συμβάλλει η χρήση του υπερκειμένου ως βασικό μέσο παρουσίασης των πληροφοριών αλλά και η ευκολία στην συγγραφή και ανάρτηση των πληροφοριών στο διαδίκτυο.

Η χρήση του παγκοσμίου ιστού ως κύριο μέσο παρουσίασης πληροφοριών δεν είναι χωρίς προβλήματα που σχετίζονται κυρίως με την οργάνωση, την διαχείριση καθώς και με την συντήρηση των πληροφοριών. Αυτά τα προβλήματα, σε συνδυασμό με την απουσία μιας αρχής-οργανισμού, που σκοπό θα έχει τον έλεγχο των πληροφοριών του διαδικτύου για τη δημιουργία ενός κεντρικού καταλόγου περιεχομένων, δημιουργούν μεγάλα προβλήματα στην αναζήτηση πληροφοριών.

Στο επόμενο κεφάλαιο, θα παρουσιάσαμε αναλυτικά τις δυνατότητες που έχει στην διάθεση του ένας χρήστης του διαδικτύου για την αναζήτηση και συλλογή πληροφοριών.

Κεφάλαιο 3

ΑΝΑΖΗΤΗΣΗ ΠΛΗΡΟΦΟΡΙΩΝ ΣΤΟ ΔΙΑΔΙΚΤΥΟ

3.1 Εισαγωγή

Πριν από μερικά χρόνια, ελάχιστοι επιστήμονες προέβλεψαν την ανάπτυξη του παγκοσμίου ιστού που ενώ ξεκίνησε από τα εργαστήρια του CERN ως ένα μέσο ανταλλαγής επιστημονικών εργασιών, σήμερα είναι ένας τεράστιος κατανεμημένος πληροφοριακός χώρος ο οποίος περιλαμβάνει : α) σελίδες χρηστών που περιγράφουν τα ενδιαφέροντα τους και τις απόψεις τους β) σελίδες ακαδημαϊκών και ερευνητικών ιδρυμάτων που παρέχουν γενικές πληροφορίες, δημοσιευμένες ερευνητικές εργασίες, προγράμματα κ.λπ. γ) κυβερνητικές πληροφορίες και ανακοινώσεις προς τους πολίτες σχετικά με τις υπηρεσίες του κράτους, διαγωνισμούς, προσλήψεις κ.λπ. δ) σελίδες οργανισμών και εταιριών σχετικά με τα προϊόντα ή/και υπηρεσίες που παρέχουν ε) πολιτιστικά κέντρα και εκθέσεις, μουσεία, βιβλιοθήκες κ.λπ.

Πριν από τη δημιουργία του πρωτοκόλλου HTTP και βασικά την γέννηση του παγκοσμίου ιστού, στο χώρο του διαδικτύου κυριαρχούσε το πρωτόκολλο FTP [Γ3]. Ο μόνος τρόπος για να δώσει κάποιος πρόσβαση στα δεδομένα του, ήταν να τα τοποθετήσει σε ένα κατάλογο ενός FTP εξυπηρετητή, ο οποίος έδινε την δυνατότητα της "*ανώνυμης σύνδεσης*" - δηλαδή την δυνατότητα σύνδεσης και τοποθέτησης αρχείων σε συγκεκριμένους καταλόγους στον εξυπηρετητή χωρίς να έχει πιστοποιηθεί ο επισκέπτης ότι είναι και χρήστης του συστήματος.

Η εύρεση ενός αρχείου, ειδικά σε μικρούς και όχι αρκετά δημοφιλείς εξυπηρετητές, ήταν μια αρκετά δύσκολη υπόθεση αν δεν είχε ανακοινωθεί η διάθεση του σε διάφορες λίστες μηνυμάτων μέσω του ηλεκτρονικού ταχυδρομείου. Το πρόγραμμα Archie [Γ35], το οποίο αναπτύχθηκε το 1990 από τον Alan Emtage στο Πανεπιστήμιο McGill του Μόντρεαλ, ήταν μια βάση δεδομένων η οποία ανανεωνόταν αρκετά συχνά με τα περιεχόμενα των *ανώνυμων* FTP εξυπηρετητών και επέτρεπε την έρευνα βάση του ονόματος για τον εντοπισμό ενός ή περισσοτέρων αρχείων.

Η λειτουργία του Archie επεκτάθηκε αργότερα και σε "*καθαρά*" αρχεία κειμένου

(χωρίς εικόνες και συνδέσμους) τα οποία βρίσκονταν σε GOPHER [Γ2] εξυπηρετητές. Το πρόγραμμα Veronica [Γ36], το οποίο αναπτύχθηκε στο τμήμα υπολογιστών του Πανεπιστημίου της Νεβάδα, έδινε την δυνατότητα της έρευνας σε αυτού του είδους τα αρχεία. Η έρευνα γινόταν είτε βάση του ονόματος του αρχείου είτε με την χρήση μερικών λέξεων με λογικούς τελεστές όπως το AND, OR και NOT οι οποίες ταίριαζαν όχι με τα περιεχόμενα του κειμένου αλλά με τις λέξεις που συνόδευαν το κάθε κείμενο.

Σήμερα τα παραπάνω προγράμματα δεν υποστηρίζονται γιατί η συγγραφή και η διάχυση των πληροφοριών στον παγκόσμιο ιστό έχει αλλάξει μορφή και μάλιστα είναι μια αρκετά πιο εύκολη και απλή διαδικασία. Η ανακοίνωση και ενημέρωση των χρηστών για τις νέες πληροφορίες δεν είναι εξίσου εύκολη και κατά συνέπεια και η αναζήτηση πληροφοριών στον παγκόσμιο ιστό υποστηρίζεται από άλλου είδους εφαρμογές που λειτουργούν σε συγκεκριμένους δικτυακούς κόμβους.

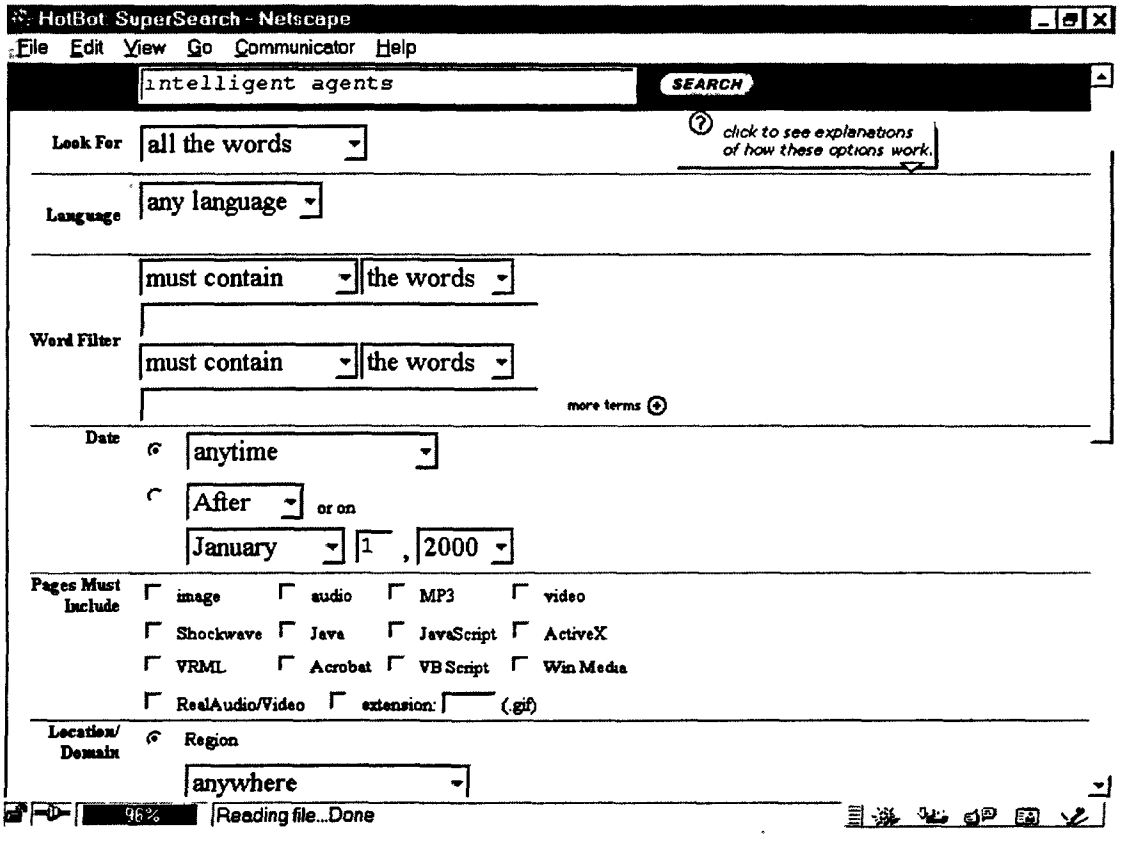
Είναι αρκετά δύσκολο να υπολογίσει κάποιος τον ακριβή αριθμό των σελίδων του διαδικτύου ακόμα και σε προσέγγιση της τάξης του εκατομμυρίου σελίδων. Η πρόσφατη έρευνα του οργανισμού Internet Software Consortium [Γ43.a] υπολόγισε ότι μέχρι τον Ιούλιο του 2001 οι DNS, σε ολόκληρο το κόσμο, περιείχαν 125.888.197 διαφορετικά ονόματα κόμβων. Πρόκειται για μια εκθετική αύξηση διότι ο ίδιος αριθμός μόλις το 1993 ήταν 1.313.000 ενώ το 1999 ήταν 56.218.000. Επίσης, τον Οκτώβριο του 2001 στον παγκόσμιο ιστό λειτουργούσαν τουλάχιστον 33 εκατομμύρια HTTP εξυπηρετητές [Γ43.d]. Με μέτριους υπολογισμούς, αν θεωρήσουμε ότι κατά μέσο όρο κάθε εξυπηρετητής διαθέτει τουλάχιστον 100 στατικές σελίδες, τότε αμέσως έχουμε μια εκτίμηση για το κάτω όριο του αριθμού των σελίδων του παγκοσμίου ιστού.

Όλα τα παραπάνω στοιχεία σε συνδυασμό με την έλλειψη ενός κεντρικού καταλόγου περιεχομένων και με την δυνατότητα του οποιουδήποτε να δημοσιεύει οτιδήποτε, και μάλιστα οποιαδήποτε στιγμή, τότε μιλάμε για ένα τεράστιο μη καταλογογραφημένο όγκο πληροφοριών, ο οποίος δημιουργεί προβλήματα στην αναζήτηση ενός χρήστη για σχετική πληροφορία σε ένα συγκεκριμένο θέμα. Ο όγκος των πληροφοριών κάνει *απαγορευτική* σε τοπικό επίπεδο - από ένα και μόνο χρήστη - την αρχικοποίηση της έρευνας και στην συνέχεια την άντληση των πληροφοριών από τον παγκόσμιο ιστό. Η επεξεργαστική ισχύς αλλά και ο χρόνος που απαιτείται για την ολοκλήρωση μιας και μόνο ερώτησης, είναι ικανό επιχείρημα για την αποτροπή ενός τέτοιου εγχειρήματος.

3.2 Διαδικτυακοί Κόμβοι Αναζήτησης Πληροφοριών

Η τεράστια ποσότητα πληροφορίας που είναι διαθέσιμη στο διαδίκτυο, είναι αναξιοποίητη από τους περισσότερους χρήστες, εκτός και εάν έχει δεικτοδοτηθεί σε έναν η περισσότερους Διαδικτυακούς Κόμβους Αναζήτησης Πληροφοριών - ΔΚΑΠ, οι οποίοι προσπαθούν να δώσουν λύση στο πρόβλημα της έλλειψης ενός κεντρικού καταλόγου περιεχομένων για το διαδίκτυο.

Οι ΔΚΑΠ παρέχουν την δυνατότητα στους επισκέπτες τους, μέσω μιας διεπαφής, να απευθύνουν ερωτήσεις για ένα οποιοδήποτε θέμα. Ο εξυπηρετητής του ΔΚΑΠ αφού παραλάβει την αίτηση του επισκέπτη, εν συνεχεία ανακτά από την βάση δεδομένων του όλες τις ηλεκτρονικές διευθύνσεις, που σχετίζονται με το συγκεκριμένο θέμα.



Σχήμα 3.1 Όλες οι επιλογές έρευνας του δικτυακού κόμβου Hotbot

Οι πληροφορίες και τα θέματα που είναι αποθηκευμένα στις βάσεις δεδομένων των ΔΚΑΠ ποικίλουν, από γενικά μέχρι πολύ ειδικά, για παράδειγμα μπορεί να αναφέρονται σε εξειδικευμένους κυβερνοχώρους (π.χ. ειδικά επιστημονικά θέματα [Γ41], αναζήτηση εργασίας [Γ40] κ.λπ) ή να ενδιαφέρουν χρήστες με συγκεκριμένα

θεματικά ενδιαφέροντα όπως για τη μουσική [Γ38], τον κινηματογράφο [Γ37], τις τέχνες [Γ39] κ.λπ.

Ο χρήστης μέσω μιας διεπαφής, που συνήθως έχει την μορφή μιας HTML φόρμας [Γ12.b], απευθύνει στον εξυπηρετητή του κόμβου μια ερώτηση που αποτελείται από λέξεις κλειδιά με ή χωρίς λογικούς τελεστές ή/και φράσεις που πρέπει ή δεν πρέπει να εμπεριέχονται στις απαντήσεις. Όλοι σχεδόν οι ΔΚΑΠ προσφέρουν δύο εκδόσεις των διεπαφών τους, μια απλή και μια λεπτομερειακή, με σκοπό να διευκολύνουν το χρήστη κατά την έρευνα του, ανάλογα με το επίπεδο γνώσης του στο χειρισμό της διεπαφής.

Το σχήμα 3.1 εμφανίζει την "*πλούσια*" έκδοση που μπορεί να χρησιμοποιήσει στην έρευνα του ο επισκέπτης του δικτυακού κόμβου HotBot [Γ16]. Στην συγκεκριμένη φόρμα, ο επισκέπτης θα πρέπει να πληκτρολογήσει τις λέξεις ή την φράση που θεωρεί ότι περιγράφει/ουν καλύτερα το θέμα που τον ενδιαφέρει π.χ. intelligent agents. Εν συνεχεία θα πρέπει να επιλέξει τον τρόπο με τον οποίο η μηχανή του HotBot θα ερμηνεύσει τις συγκεκριμένες λέξεις.

Η προκαθορισμένη τιμή "όλες οι λέξεις", για την επιλογή *Εξέτασε Για* (*Look For*) - δες σχήμα 3.1, θα εμφανίσει όλα τα αποτελέσματα στα οποία εμπεριέχονται όλες οι λέξεις, δηλαδή είναι αντίστοιχο του λογικού τελεστή AND. Οι εναλλακτικές τιμές της συγκεκριμένης επιλογής *Εξέτασε Για,* είναι :

Οποιαδήποτε από τις λέξεις : είναι αντίστοιχο του λογικού τελεστή OR
Επακριβώς την φράση : οι λέξεις θα πρέπει να έχουν τη συγκεκριμένη σειρά.
Τίτλο σελίδας : οι λέξεις θα πρέπει να εμφανίζονται στον τίτλο της σελίδας.
Το πρόσωπο : αναφορές σε συγκεκριμένο όνομα
Αναφορές σε μία διεύθυνση : σελίδες που περιέχουν σύνδεσμο σε μια διεύθυνση
Λογική Εκφραση : Οι λέξεις AND, OR, και NOT θεωρούνται λογικοί τελεστές

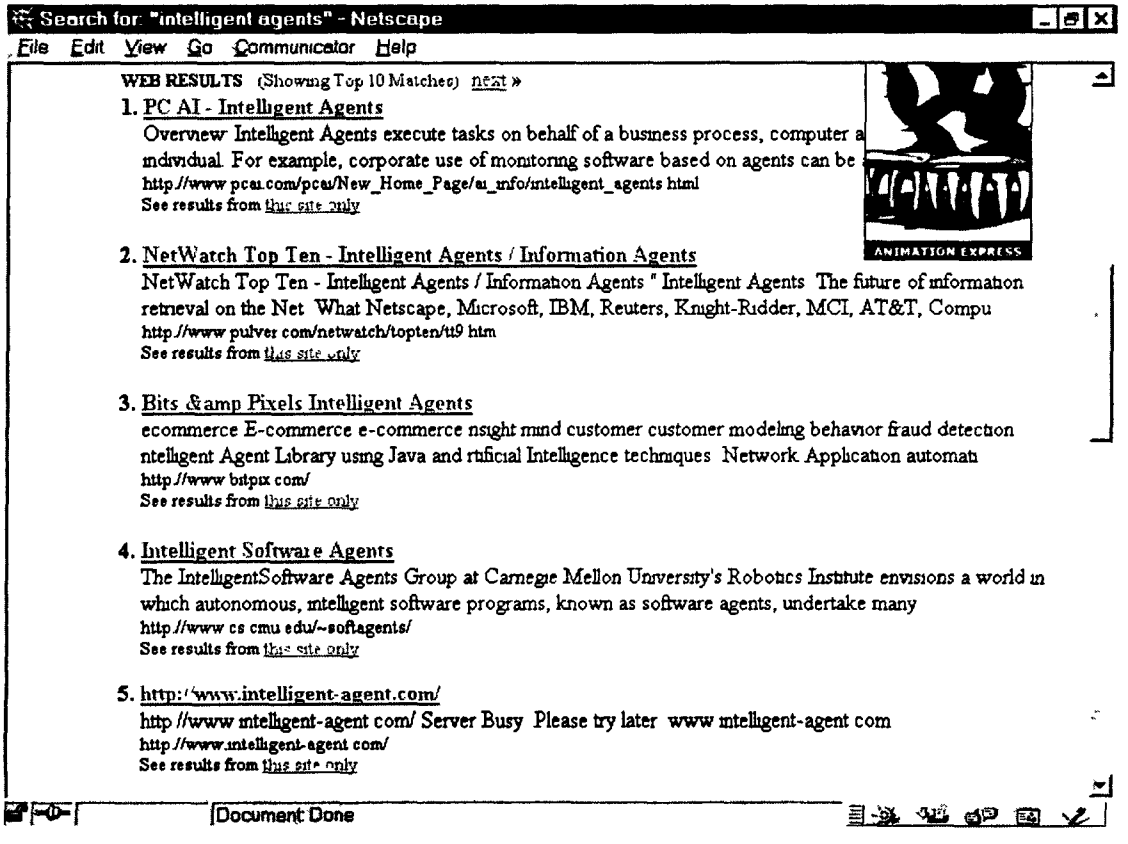
Η επιλογή *Γλώσσα (Language)* θα εμφανίσει εκείνα τα αποτελέσματα που περιέχουν κωδικοποίηση σελίδας στη συγκεκριμένη τιμή π.χ. Γαλλικά, Ιταλικά κ.λπ. Εν συνεχεία ο χρήστης με την βοήθεια των δυο άλλων κουτιών κειμένου στην επιλογή *Φίλτρο Λέξεων (Word Filter)* μπορεί να δημιουργήσει σύνθετες ερωτήσεις δηλώνοντας επιπλέον τις λέξεις ή τις εκφράσεις που *πρέπει* ή *δεν πρέπει* ή *μπορεί* να περιέχονται στα αποτελέσματα.

Αν ο χρήστης ενδιαφέρεται για σελίδες που δημοσιεύτηκαν ή τροποποιήθηκαν

για τελευταία φορά πριν ή μετά από μια συγκεκριμένη ημερομηνία, θα πρέπει να το δηλώσει στην επιλογή *Ημερομηνία (Date),* η οποία έχει σαν προκαθορισμένη τιμή *οποτεδήποτε* δηλαδή χωρίς χρονικό περιορισμό.

Η επιλογή *Οι Σελίδες Πρέπει Να Περιέχουν* (*Pages Must Include*) μας επιτρέπει να δηλώσουμε τους τύπους αρχείων π.χ. εικόνες, μουσική κ.λπ., που θα πρέπει να περιέχονται στα αποτελέσματα της έρευνας, εκτός φυσικά από τις λέξεις ή φράσεις που περιγράψαμε στις προηγούμενες επιλογές.

Ο περιορισμός της έρευνας σε μια συγκεκριμένη γεωγραφική περιοχή ή κόμβο επιτυγχάνεται με την επιλογή *Τοποθεσία / Κόμβο* (*Location / Domain*). Στις τιμές της συγκεκριμένης επιλογής περιλαμβάνονται οι ήπειροι π.χ. Ευρώπη, Αυστραλία κ.λπ. αλλά και συγκεκριμένοι τύποι κόμβων (π.χ. .com, .edu κ.λπ.)



Σχήμα 3.2 Η λίστα των αποτελεσμάτων του κόμβου Hotbot

Η μηχανή του HotBot έχει σαν προκαθορισμένη επιλογή να επιστρέφει μια σελίδα ανά κόμβο αυτή που θεωρεί σαν καλύτερη, δηλαδή πιο κοντά στην ερώτηση. Στην επιλογή *Καλύτερη Σελίδα Μόνο* (*Best Page Only*) δίνεται η δυνατότητα στον επισκέπτη να αδρανοποιήσει τη συγκεκριμένη λειτουργία ώστε να επιστρέψει η μηχανή περισσότερες σελίδες ανά κόμβο. Ο περιορισμός της έρευνας σε ένα κόμβο

μόνο στην κεντρική σελίδα ή μέχρι ενός συγκεκριμένου βάθους, ενεργοποιείται από την επιλογή *Βάθος Σελίδας* (*Page Depth*). Η προκαθορισμένη τιμή για την συγκεκριμένη επιλογή είναι *οπουδήποτε*.

Η προτελευταία επιλογή ενεργοποιεί την λειτουργία της *Λεξικής Καταγωγής* (*Word Stemming*). Χάρη στην συγκεκριμένη λειτουργία, η μηχανή του HotBot εμφανίζει αποτελέσματα τα οποία έχουν γραμματικές διαφοροποιήσεις με τις λέξεις της έρευνας. Για παράδειγμα, αν η λέξη της έρευνας είναι "*thought*" θα εμφανίσει αποτελέσματα που θα περιέχουν "think" και "thinking". Τέλος ο αριθμός των αποτελεσμάτων που θα επιδεικνύονται ανά σελίδα καθορίζεται στην επιλογή *Επιστροφή Αποτελεσμάτων* (*Return Results*). Οι τρεις τελευταίες επιλογές βρίσκονται στο κάτω μέρος του σχήματος 3.1.

Από όλες τις παραπάνω διαθέσιμες λειτουργίες, καταλαβαίνουμε ότι ο επισκέπτης του κόμβου έχει στην διάθεση του πληθώρα επιλογών με σκοπό την περιγραφή του αντικειμένου της έρευνας του με τον πλέον περιγραφικό τρόπο. Δεν είναι απαραίτητο ο χρήστης να δώσει μια τιμή σε όλες τις επιλογές, αλλά μπορεί να διατηρήσει τις προεπιλεγμένες τιμές και εν συνεχεία να πατήσει το πλήκτρο *SEARCH*, με σκοπό να αποστείλει τις παραμέτρους της ερώτησης στον εξυπηρετητή του ΔΚΑΠ.

Η συναλλαγή ανάμεσα στο πρόγραμμα πλοήγησης ενός χρήστη και στον εξυπηρετητή βασίζεται στο καθιερωμένο μοντέλο αίτησης/απόκρισης του πρωτοκόλλου HTTP. Ο πελάτης, αφού αποστείλει την αίτηση, εν συνεχεία περιμένει την απόκριση του εξυπηρετητή.

Οι εργασίες που θα κάνει ο εξυπηρετητής με την παραλαβή της αίτησης είναι: α) να αναλύσει την αίτηση του χρήστη και εν συνεχεία να ελέγξει αν είναι αποδεκτή β) να επεξεργαστεί την αίτηση, αυτό το βήμα μπορεί να είναι αρκετά πολύπλοκο και να περιλαμβάνει επικοινωνία με άλλους εξυπηρετητές, με βάσεις δεδομένων κ.λπ., γ) να μορφοποιήσει και να αποστείλει τα αποτελέσματα στο πελάτη. Για λόγους απλούστευσης και χωρίς να χάνουμε την ουσία, θεωρούμε ότι ο εξυπηρετητής εκτελεί όλες τις παραπάνω εργασίες, ενώ στην πραγματικότητα ενεργοποιεί μια εφαρμογή (π.χ. CGI ή πρόγραμμα εξυπηρετητή) η οποία εκτελείται σαν ανεξάρτητη διεργασία και θα κάνει την απαραίτητη επεξεργασία.

Ο εξυπηρετητής του Hotbot, μετά από την επεξεργασία της αίτησης, απαντά με μια λίστα αποτελεσμάτων, όπως αυτή εμφανίζεται στο σχήμα 3.2. Τα αποτελέσματα ομαδοποιούνται και εμφανίζονται σε μορφή σελίδας. Το κάθε ένα από αυτά

περιλαμβάνει έναν τίτλο, μια διεύθυνση και μια σύντομη περιγραφή. Όλα τα αποτελέσματα εμφανίζονται συνήθως ταξινομημένα ανάλογα με την αξία τους ως προς την ερώτηση του χρήστη. Ο χρήστης μπορεί να επιλέξει να δει την επόμενη σελίδα των αποτελεσμάτων ή να διαλέξει μια από τις ηλεκτρονικές διευθύνσεις για να συνεχίσει την περιήγηση του στο διαδίκτυο.

Σε αυτό το σημείο θα πρέπει να τονίσουμε ότι, παρά τις προσπάθειες [Γ44] για μια κοινά αποδεκτή ορολογία και προτυποποίηση των δυνατοτήτων-επιλογών που προσφέρουν οι ΔΚΑΠ, οι επιλογές δεν είναι ίδιες για όλες αλλά ποικίλουν από κόμβο σε κόμβο και αυτός είναι ένας από τους λόγους επίσκεψης ενός χρήστη. Ο δεύτερος λόγος επίσκεψης, και ίσως ο σημαντικότερος, είναι η ποιότητα και ο όγκος της βάσης δεδομένων του.

3.2.1 Η Δημιουργία της Βάσης Δεδομένων των Διαδικτυακών Κόμβων Αναζήτησης Πληροφοριών

Νέες πληροφορίες συλλέγονται και εμπλουτίζουν την βάση δεδομένων των ΔΚΑΠ είτε από ανθρώπους που είναι ειδικοί σε ένα συγκεκριμένο θέμα είτε αυτόματα από ειδικές εφαρμογές λογισμικού. Αυτή η διαφοροποίηση χωρίζει αντίστοιχα τους ΔΚΑΠ σε δύο κατηγορίες α) στους Θεματικούς Καταλόγους - ΘΕΚ και στις β) ΜηΧανές Αναζήτησης - ΜΧΑ.

Οι βάσεις δεδομένων των ΘΕΚ ενημερώνονται από ανθρώπους οι οποίοι έχουν δημιουργήσει ένα κεντρικό κατάλογο με θεματικές ενότητες και κατατάσσουν όλες τις νέες πληροφορίες βάση της δομής του καταλόγου ή δημιουργούν νέα κατηγορία στον κατάλογο. Για παράδειγμα, σε ένα δικτυακό κόμβο που ο κατάλογος του αποτελείται από θεματικές ενότητες και ο επισκέπτης του κόμβου ενδιαφέρεται για το πρόγραμμα σπουδών ενός τμήματος υπολογιστών στην Ελλάδα, τότε θα πρέπει να ξεκινήσει την ερευνά του από την θεματική ενότητα Εκπαίδευση, εν συνεχεία να επιλέξει γεωγραφική περιοχή π.χ. Ευρώπη, μετά να επιλέξει την χώρα π.χ. Ελλάδα, ύστερα να διαλέξει ανάμεσα στις βαθμίδες της Εκπαίδευσης όπως Ανωτάτη, Ανωτέρα, Μεταλυκειακή κ.λπ., έπειτα την σχολή και το τμήμα που τον ενδιαφέρει, και τέλος το πρόγραμμα σπουδών.

Τρεις πολύ γνωστοί δικτυακοί κόμβοι, οι κατάλογοι των οποίων χωρίζονται σε θεματικές ενότητες και ενημερώνονται από ανθρώπους είναι οι : Yahoo [Γ18], Open Directory [Γ31], και LookSmart [Γ32] . Βέβαια, και οι τρεις παρέχουν εναλλακτικά

μια διεπαφή, όπου ο χρήστης χρησιμοποιεί λέξεις κλειδιά για να ανακτήσει διευθύνσεις με σχετικές πληροφορίες αντί να διασχίζει τον κατάλογο. Αν για την ενημέρωση των βάσεων δεδομένων των ΘΕΚ είναι υπεύθυνος ένας άνθρωπος ή μια ομάδα ειδικών, για τις μηχανές αναζήτησης η διαδικασία είναι αυτόνομη και μηχανική μέσω προγραμμάτων λογισμικού που ονομάζονται *spiders*, *robots* ή απλά *bots* [Wil96][Che96].

Τα προγράμματα αυτά μπορούν να κάνουν μια αίτηση και εν συνεχεία να ανακτήσουν την απάντηση του εξυπηρετητή, όπως ακριβώς τα προγράμματα πλοήγησης, σε αντίθεση όμως με αυτά, ο σκοπός δεν είναι τα περιεχόμενα να τα δουν άνθρωποι, αλλά να τα καταλογογραφήσουν και να τα συμπεριλάβουν στην βάση δεδομένων της ΜΧΑ.

Τις σελίδες και τους συνδέσμους στον παγκόσμιο ιστό μπορούμε να τους θεωρήσουμε σαν ένα κατευθυνόμενο γράφο [Kle99]. Με αυτόν τον τρόπο μπορούμε, όχι μόνο να επισκεφτούμε κάθε κόμβο χρησιμοποιώντας ένα αλγόριθμο διάσχισης, αλλά και να αντλήσουμε πληροφορίες, τόσο από την δομή της σελίδας, τα χαρακτηριστικά μορφοποίησης, όσο και από το κείμενο που συνοδεύει τους συνδέσμους [Spe98]. Επίσης, το μοντέλο πελάτη-εξυπηρετητή του πρωτοκόλλου HTTP επιτρέπει την χρήση ενός και μόνο υπολογιστή για την διάσχιση ολόκληρου του παγκοσμίου ιστού. Από τα παραπάνω βγαίνει το συμπέρασμα ότι μπορούμε να δημιουργήσουμε αυτόματα έναν κατάλογο με την χρήση ενός *robot* το οποίο θα χρησιμοποιήσει μια από τις ακόλουθες μεθοδολογίες [Gud97][Cho98]

α) Το *robot* ξεκινάει την έρευνα από μια αρχική διεύθυνση και αφού καταλογογραφήσει το κείμενο της αρχική διεύθυνσης, εν συνεχεία συγκεντρώνει όλες τις ηλεκτρονικές διευθύνσεις που εμπεριέχονται σε αυτό. Στην συνέχεια, αναδρομικά συγκεντρώνονται και καταλογογραφούνται όλα τα κείμενα που δείχνουν οι διευθύνσεις είτε ακολουθώντας διάσχιση κατά πλάτος ή διάσχιση κατά μήκος.

β) Το *robot* ξεκινάει με ένα σύνολο από ηλεκτρονικές διευθύνσεις που περιέχουν σχετικά μεταξύ τους θέματα. Για παράδειγμα, μπορεί να είναι μερικές από τις δημοφιλέστερες σελίδες για ένα θέμα. Η υπόθεση είναι ότι σε αυτές τις σελίδες θα περιέχονται μόνο σχετικές με το θέμα διευθύνσεις.

γ) Η διαίρεση του διαδικτύου βάση ονομάτων ή κωδικών χωρών και η ανάθεση σε

ένα ή περισσότερα *robots* να ερευνήσουν το κάθε τμήμα. Η συγκεκριμένη μέθοδος είναι αυτή που χρησιμοποιείται από τις περισσότερες ΜΧΑ.

Ο κατάλογος με τις λέξεις και τις αναφορές τους στις ηλεκτρονικές διευθύνσεις των κειμένων στο διαδίκτυο εμπλουτίζεται συνεχώς, αλλά, και οι υπάρχουσες διευθύνσεις δέχονται επισκέψεις σε τακτά χρονικά διαστήματα, με σκοπό να συμπεριλάβουν τυχόν αλλαγές. Η αυτόματη ενημέρωση στις ΜΧΑ ή η ενημέρωση ενός ΘΕΚ από ειδικούς επί του θέματος, είναι ένας από τους τρόπους εμπλουτισμού με νέες πληροφορίες των καταλόγων των ΔΚΑΠ. Ο δεύτερος τρόπος είναι η ενημέρωση μέσω του διαχειριστή ενός κόμβου ή από τον συγγραφέα μιας σελίδας.

Όλοι οι ΔΚΑΠ παρέχουν, μέσω μιας HTML φόρμας [Γ12.b], την δυνατότητα της αποστολής μιας ηλεκτρονικής διεύθυνσης ενός κόμβου ή μιας σελίδας, με σκοπό την ενσωμάτωση της στη βάση δεδομένων του. Η επιλογή αυτή επιτρέπει στον διαχειριστή ενός κόμβου ή στον συγγραφέα μιας σελίδας, να ενημερώσει για το νέο ή ανανεωμένο υλικό που θα πρέπει κατά την άποψη του αποστολέα να συμπεριληφθεί στον κατάλογο. Συνήθως στην φόρμα που χρησιμοποιείται για την συγκεκριμένη εργασία υπάρχει α) χώρος για μια σύντομη περιγραφή της σελίδας η οποία και θα χρησιμοποιηθεί κατά την επίδειξη των αποτελεσμάτων β) χώρος για την χρήση μερικών λέξεων κλειδιών που θα σχετίζονται με το περιεχόμενο γ) επιλογή για ένταξη του περιεχομένου σε μία γενική θεματική κατηγορία.

Η ενημέρωση ενός ΔΚΑΠ για νέο υλικό, δεν προϋποθέτει και την αυτόματη εισαγωγή της διεύθυνσης στη βάση δεδομένων αλλά θα πρέπει να γίνει επίσκεψη του κόμβου, έλεγχος του υλικού και τέλος απόρριψη ή ένταξη της διεύθυνσης και των σχετικών πληροφοριών που την συνοδεύουν στη βάση δεδομένων.

3.2.2 Η Καταλογογράφηση και Ταξινόμηση της Βάσης Δεδομένων των Διαδικτυακών Κόμβων Αναζήτησης Πληροφοριών

Σε ένα ΘΕΚ, οι ειδικοί, μετά την παραλαβή των στοιχείων της φόρμας, θα επισκεφθούν τον προτεινόμενο κόμβο ή την σελίδα και θα ελέγξουν την συνολική ποιότητα του υλικού, πριν αποφασίσουν την κατάταξη του στον θεματικό κατάλογο. Οι ειδικοί θα εκτιμήσουν συνολικά α) τα περιεχόμενα της σελίδας, β) την λειτουργικότητα, και γ) το γενικότερο σχεδιασμό, πριν καταλήξουν να δεχθούν ή να απορρίψουν την ένταξη της πρότασης σε κάποια συγκεκριμένη κατηγορία.

Σε αυτή την περίπτωση, τα κριτήρια τελικής κατάταξης είναι λίγο υποκειμενικά αφού η τελική απόφαση ανήκει σε ανθρώπους. Το σημαντικό είναι ότι τα περιεχόμενα της σελίδας αναφέρονται στη σχετική κατηγορία οπότε μπορεί να μην είναι η πιο αντιπροσωπευτική διεύθυνση της κατηγορίας αλλά πάντως δεν είναι και άσχετη.

Από την άλλη πλευρά, στις ΜΧΑ η ίδια εργασία δεν είναι το ίδιο εύκολη και η κάθε μια χρησιμοποιεί τις δικές της μεθόδους και κριτήρια για το τι θα συμπεριλάβει και τι όχι στην βάση δεδομένων της. Για παράδειγμα, άλλες ΜΧΑ καταλογογραφούν όλες τις σελίδες σε ένα δικτυακό κόμβο ενώ άλλες μόνο την κεντρική σελίδα. Άλλες ΜΧΑ καταλογογραφούν όλες τις λέξεις ενώ άλλες μόνο τις λέξεις του τίτλου, ή τις λέξεις που βρίσκονται στις ετικέτες Meta [Γ12.d] με ονόματα *Description* και *Keywords* [Γ12.d], ή στις περιγραφές των διευθύνσεων ή ακόμα και τις λέξεις που εμφανίζονται έντονα, πλάγια ή σε διαφορετικό μέγεθος γραμματοσειράς.

Εν συνεχεία η ταξινόμηση των αποτελεσμάτων σε μια συγκεκριμένη ερώτηση έχει να κάνει με την αξία των προτεινομένων διευθύνσεων και εξαρτάται από διαφορετικά κριτήρια. Τέτοια κριτήρια μπορεί να είναι ο αριθμός των λέξεων της ερώτησης που περιέχονται σε ένα κείμενο, η θέση των λέξεων σε ένα κείμενο, η συχνότητα των λέξεων, το μήκος του κειμένου [Cou99].

Ένα από τα προαιρετικά πεδία που μπορεί να συνοδεύουν την αίτηση του πελάτη, σύμφωνα με τις προδιαγραφές του πρωτοκόλλου HTTP, είναι και το πεδίο *Referer*. Το πεδίο αυτό καθορίζει την διεύθυνση του κειμένου στην οποία περιέχεται η τρέχουσα διεύθυνση. Αυτό σημαίνει ότι αν ένας χρήστης επισκοπεί ένα κείμενο στην ηλεκτρονική διεύθυνση Α, στο οποίο περιέχεται ένας σύνδεσμος σε ένα άλλο κείμενο με ηλεκτρονική διεύθυνση Β, και ο χρήστης επιλέξει την διεύθυνση Β, τότε το πρόγραμμα πλοήγησης θα αναφέρει στην αίτηση του προς τον εξυπηρετητή Β, την ηλεκτρονική διεύθυνση Α στο πεδίο *Referer*.

Η χρήση του πεδίου αναφέρεται στο αρχείο λειτουργίας του εξυπηρετητή Β, με σκοπό την ενημέρωση του διαχειριστή του εξυπηρετητή Α για την διόρθωση των συνδέσμων του κειμένου σε μη διαθέσιμους πόρους του εξυπηρετητή Β. Φυσικά εκτός από την προηγούμενη χρήση όλοι σχεδόν οι διαχειριστές των κόμβων μελετούν το αρχείο λειτουργίας του εξυπηρετητή με σκοπό να μάθουν από ποίο δικτυακό τόπο κατευθύνονται οι επισκέπτες προς τον κόμβο τους. Το στοιχείο αυτό θα χρησιμοποιηθεί σε μια επιτυχημένη διαφημιστική καμπάνια.

Σε ένα μεγάλο ποσοστό οι διαχειριστές αναγνωρίζουν ότι οι περισσότεροι

επισκέπτες τους κατευθύνονται προς τον κόμβο τους, αφού πρώτα έχουν περάσει από μια μηχανή αναζήτησης. Οπότε, η όσο το δυνατόν καλύτερη σειρά εμφάνισης του κόμβου τους στην κατάταξη των αποτελεσμάτων της ΜΧΑ, θα έχει σαν αποτέλεσμα περισσότερους επισκέπτες, περισσότερες πωλήσεις, περισσότερες διαφημίσεις κ.λπ. Έτσι λοιπόν προσπαθούν με κάθε τρόπο να "*μπερδέψουν*" τον μηχανισμό ταξινόμησης της ΜΧΑ ώστε να κερδίσουν μια καλύτερη θέση εμφάνισης στην κατάταξη των αποτελεσμάτων.

Για παράδειγμα, επαναλαμβάνουν ορισμένες λέξεις συνέχεια μέσα στο κείμενο με σκοπό να αυξήσουν την συχνότητα τους, και μάλιστα, εκτός από το ορατό κείμενο γράφουν ατελείωτες σειρές από τις ίδιες λέξεις αλλάζοντας το χρώμα του κειμένου ώστε να μην εμφανίζονται στο πρόγραμμα πλοήγησης αλλά να είναι ορατές στο *robot* της ΜΧΑ. Αυτές οι αόρατες σειρές λέξεων μπορεί να συνοδεύονται από ετικέτες διαμόρφωσης κειμένου, π.χ. πλάγια, έντονα, μεγαλύτερο μέγεθος γραμματοσειράς, με σκοπό την αύξηση της βαρύτητας των συγκεκριμένων λέξεων και την ταύτιση της ηλεκτρονικής διεύθυνσης του κειμένου με αυτές τις λέξεις. Η ίδια τακτική χρησιμοποιείται και στο πεδίο *keywords* της επικεφαλίδας του κειμένου, όπου αρκετοί κόμβοι με τυχερά παιχνίδια ή ερωτικό υλικό τοποθετούν άσχετες λέξεις με σκοπό να εμφανίζεται η διεύθυνση του κόμβου σχεδόν σε κάθε ερώτηση του χρήστη της ΜΧΑ.

Σε όλες τις παραπάνω περιπτώσεις ο σκοπός είναι προφανής, δηλαδή, να πετύχουν καλύτερη σειρά κατάταξης στην λίστα των αποτελεσμάτων, οπότε να προσελκύσουν όσο το δυνατόν περισσότερους επισκέπτες στο κόμβο τους. Από την άλλη πλευρά όμως για το χρήστη της ΜΧΑ όλες οι διευθύνσεις με άσχετο περιεχόμενο θεωρούνται "*σκουπίδια*" και όσο περισσότερο εμφανίζονται στις λίστες των αποτελεσμάτων τόσο περισσότερο μειώνεται η αξιοπιστία της ΜΧΑ. Η άμυνα των ΜΧΑ είναι οι ειδικοί μηχανισμοί ανίχνευσης μιας τέτοιας προσπάθειας παραπλάνησης από την πλευρά του διαχειριστή και εν συνεχεία η διαγραφή της διεύθυνσης του κόμβου από την βάση δεδομένων.

Ένας ακόμα παράγοντας, που επηρεάζει σημαντικά την κατάταξη των περιεχομένων της βάσης δεδομένων και δεν εξαρτάται άμεσα από το ίδιο το κείμενο, είναι και ο αριθμός των σελίδων από άλλους κόμβους που δείχνουν σε μια σελίδα. Όσο μεγαλύτερος είναι αυτός ο αριθμός τόσο πιο δημοφιλής θεωρείται η σελίδα ή ο κόμβος.

Οι ΜΧΑ χρησιμοποιούν και άλλες μεθόδους κατάταξης των κειμένων οι οποίες

έχουν να κάνουν περισσότερο με την κρίση και τον βαθμό ικανοποίησης του επισκέπτη στα αποτελέσματα που εμφανίζονται για μια συγκεκριμένη ερώτηση. Αυτές οι μέθοδοι είναι : α) *η πρώτη διεύθυνση*, έχει να κάνει με τη διεύθυνση που επισκέπτονται οι περισσότεροι χρήστες στην πρώτη σελίδα των αποτελεσμάτων. Η υπόθεση είναι : μετά την εμφάνιση των αποτελεσμάτων και αφού ο χρήστης διαβάσει την σελίδα με τις περιλήψεις εν συνεχεία αποφασίζει να επισκεφθεί μια διεύθυνση που νομίζει ότι είναι πιο κοντά στην ερευνά του. Άρα αυτή η διεύθυνση είναι η "*καλύτερη*" ανάμεσα στις άλλες διευθύνσεις. β) *παραμονή σε ένα κόμβο*, έχει να κάνει με το χρόνο που παραμένει ο χρήστης σε έναν κόμβο και τον μετράμε μεταξύ δύο διαδοχικών διευθύνσεων που εμφανίζονται στην σελίδα των αποτελεσμάτων. Αν ο χρόνος παραμονής είναι μικρός σημαίνει ότι ο χρήστης δεν ικανοποιήθηκε και συνεπώς θα πρέπει να μειωθεί η αξία των λέξεων που σχετίζονται με την συγκεκριμένη διεύθυνση.

3.2.3 Θεματικοί Κατάλογοι ή Μηχανές Αναζήτησης;

Ένα ερώτημα που διατυπώνουν συχνά οι χρήστες του διαδικτύου είναι "*αν θα πρέπει κατά την έρευνα τους να προτιμήσουν μια ΜΧΑ ή ένα ΘΕΚ*;" Μία σύντομη απάντηση είναι ότι δεν υπάρχει κάποιος ιδιαίτερος λόγος να προτιμήσει ένας χρήστης μια ΜΧΑ ή ένα ΘΕΚ αφού τα πλεονεκτήματα και τα μειονεκτήματα μοιράζονται εξίσου ανάμεσα στις δύο κατηγορίες των ΔΚΑΠ.

Από την μία πλευρά αρκετές μηχανές αναζήτησης [Γ14][Γ16][Γ17] προβάλουν σαν πλεονέκτημα τον όγκο των πληροφοριών που διαθέτουν και ανακοινώνουν ότι οι βάσεις δεδομένων τους περιέχουν πάνω από 2 εκατομμύρια διαφορετικές διευθύνσεις, οπότε έχουν ένα αρκετά πλούσιο κατάλογο, ενώ από την άλλη πλευρά οι ΘΕΚ συμφωνούν ότι δεν διαθέτουν μεγάλο πλήθος διευθύνσεων αλλά λίγες χρήσιμες και ποιοτικές διευθύνσεις.

Η κατηγοριοποίηση και ενημέρωση της βάσης δεδομένων στους ΘΕΚ είναι μια αρκετά δύσκολη και κουραστική εργασία, λόγω του μεγάλου όγκου πληροφοριών και του μικρού πλήθους των ανθρώπων που ασχολούνται με την ταξινόμηση [Fil98]. Επίσης είναι δύσκολη υπόθεση η εύρεση ειδικών σε κάποιο χώρο και η δημιουργία και συντήρηση μια νέας κατηγορίας στον κατάλογο.

Βασικό προτέρημα των ΘΕΚ είναι η ευκολία χρήσης, αφού ο επισκέπτης έχει την δυνατότητα να διασχίσει τον κατάλογο και να βρει την πληροφορία που τον

ενδιαφέρει αρκετά εύκολα, χωρίς να χρειάζεται να κάνει χρήση λέξεων κλειδιών και λογικών τελεστών.

Επίσης, άλλο πλεονέκτημα, είναι ότι ο επισκέπτης γνωρίζει πως κάποιος ειδικός έχει επιβλέψει την όλη ταξινόμηση και προτείνει την συγκεκριμένη διεύθυνση. Ακόμα, το σύνολο των προτάσεων είναι σχετικά μικρό και αρκετά ποιοτικό, οπότε αποτελεί ένα απαραίτητο αρχικό βήμα για την ολοκλήρωση μιας επιτυχημένης έρευνας. Βέβαια ένας ΘΕΚ δεν είναι χρήσιμος σε έναν επισκέπτη που έχει αρκετά εξειδικευμένα ενδιαφέροντα για τα οποία δεν υπάρχει σχετική κατηγορία στον κατάλογο.

Οι περισσότερες δημοφιλείς ΜΧΑ προσπαθούν να καταλογογραφήσουν όλο τον παγκόσμιο ιστό και μάλιστα κάθε λέξη που αναφέρεται στα κείμενα. Για το σκοπό αυτό χρησιμοποιείται ειδικό υλικό και λογισμικό ώστε να ανταποκριθούν σε αυτό το εγχείρημα καθώς οι ανάγκες για αποθηκευτικό χώρο είναι μεγάλες όπως επίσης και για γρήγορη αναζήτηση στον κατάλογο, ώστε η ανταπόκριση στην ερώτηση του χρήστη να είναι σχεδόν άμεση. Παράλληλες τεχνικές χρησιμοποιούνται ώστε να ερευνώνται οι τεράστιοι κατάλογοι με τις λέξεις σε χρόνο μερικών ms. Όμως η καταλογογράφηση όλου του παγκοσμίου ιστού με την χρήση ενός αυτόματου προγράμματος διάσχισης είναι αδύνατη [Dar98], όχι γιατί δεν είναι τεχνικά εφικτό αλλά λόγω του ότι :

α) Ο διαχειριστής του κόμβου ή ο συγγραφέας των σελίδων κάνει χρήση του Standard for Robot Exclusion [Γ45] - ένα σύνολο από κανόνες που θα πρέπει να σέβονται τα robots - με σκοπό την αποτροπή της καταλογογράφησης ορισμένων σελίδων, καταλόγων ή ολόκληρου του κόμβου.

β) απαιτείται η πιστοποίηση του επισκέπτη για την πρόσβαση του σε συγκεκριμένους πληροφοριακούς χώρους, π.χ. ειδησιογραφικά πρακτορεία, ηλεκτρονικές βιβλιοθήκες.

γ) οι σελίδες στον παγκόσμιο ιστό δεν είναι όλες στατικές αλλά ένα μεγάλο πλήθος και ίσως το πιο ενδιαφέρον, αποτελείται από δυναμικά παραγόμενες σελίδες οι οποίες μορφοποιούνται μετά την παραλαβή των πεδίων μιας φόρμας, ή οποία είτε συμπληρώνεται από τον επισκέπτη είτε ενεργοποιείται αυτόματα. Στην περίπτωση των δυναμικών σελίδων απαιτείται η επίβλεψη ενός ανθρώπου για την συμπλήρωση

των πεδίων της φόρμας.

δ) αρκετοί κόμβοι για την όσο το δυνατό καλύτερη και ευχάριστη περιήγηση των επισκεπτών τους : ανταλλάσσουν πληροφορίες (Cookies [Γ13]) με το πρόγραμμα πλοήγησης, χρησιμοποιούν γλώσσες προγραμματισμού και αντικείμενα για το πρόγραμμα πλοήγησης [Γ9,Γ10,Γ25,Γ30], χρησιμοποιούν εικόνες που συνδέονται με συγκεκριμένες ενέργειες [Γ12.e] κ.λπ. Όλα τα προηγούμενα δεν είναι κατανοητά και συνεπώς χρήσιμα σε ένα robot.

ε) αρκετές αν όχι όλες οι σελίδες περιέχουν αρχεία κειμένων και πολυμέσων στα οποία εκτός από το όνομα του αρχείου τα περιεχόμενα τους δεν είναι δυνατόν να καταλογογραφηθούν. Παρ' όλες τις ερευνητικές προσπάθειες [Smi97][Cha98] και τις προτάσεις για καταλογογράφηση τουλάχιστον των κειμένων [Bye98] οι ΜΧΑ συνεχίζουν να αγνοούν όλα τα περιεχόμενα εκτός HTML μορφοποίησης.

στ) αρκετοί συγγραφείς σελίδων χρησιμοποιούν την εθνική τους γλώσσα στον παγκόσμιο ιστό. Οι περισσότερες ΜΧΑ αγνοούν όλους τους μη αγγλικούς χαρακτήρες [Dar98], παρ' ότι υπάρχουν οι προδιαγραφές [Γ42] για την χρήση διαφορετικών συνόλων χαρακτήρων.

Οι παραπάνω αδυναμίες των ΜΧΑ θα κάνουν ίσως τους ΘΕΚ να εμφανίζονται σαν πιο ελκυστική λύση για την αναζήτηση πληροφοριών στο διαδίκτυο. Όμως παρά τις αδυναμίες τους, οι ΜΧΑ έχουν ιδιαίτερη αξία για την συλλογή πληροφοριών λόγω του πλήθους των διευθύνσεων που περιέχουν στην βάση δεδομένων τους.

3.3 Προβλήματα από τη Χρήση των Διαδικτυακών Κόμβων Αναζήτησης Πληροφοριών

Τα χαρακτηριστικά των ΔΚΑΠ καθώς και ο όγκος των πληροφοριών ποικίλει και κανένας τους δεν παρέχει τον καλύτερο κατάλογο για όλα τα θέματα. Επίσης, κανένας ΔΚΑΠ δεν καλύπτει περισσότερο από το ένα τρίτο του όγκου πληροφοριών του παγκοσμίου ιστού που μπορεί να καταλογογραφηθεί. Συνεπώς για την όσο το δυνατόν καλύτερη και πληρέστερη ενημέρωση, θα πρέπει να συνδυαστούν τα αποτελέσματα από διαφορετικούς ΔΚΑΠ [Law98] [Bha98]. Παράγοντας σημαντικός

που αυξάνει την χρηστική αξία ενός ΔΚΑΠ και συνεπώς την ικανοποίηση ενός χρήστη, δεν είναι μόνο το μέγεθος της βάσης δεδομένων αλλά και η ποιότητα, ειδικά με τόσα "*σκουπίδια*" που βρίσκονται στον παγκόσμιο ιστό [Bri98].

Έτσι λοιπόν ο κάθε επισκέπτης για να ανακτήσει χρήσιμη πληροφορία θα πρέπει να πλοηγηθεί από μία δικτυακή τοποθεσία σε μία άλλη, να μάθει τις διαφορετικές διεπαφές, τις διαθέσιμες επιλογές και τα χαρακτηριστικά της κάθε ΜΧΑ ή ΘΕΚ, να αποστείλει τις ερωτήσεις και τέλος να συλλέξει και να ομαδοποιήσει τα αποτελέσματα αφού πρώτα διαγράψει τις διπλές εγγραφές.

Ευτυχώς τα τελευταία χρόνια έχουν αναπτυχθεί οι ΜεταΜηΧανές Αναζήτησης - ΜΜΧΑ, όπως για παράδειγμα ο MetaCrawler [Γ33], DogPile [Γ34], οι οποίες δεν διαθέτουν δική τους βάση δεδομένων αλλά λειτουργούν κάνοντας χρήση των άλλων ΜΧΑ, μέσω μίας κοινής διεπαφής. Ο χρήστης απευθύνει μία ερώτηση στην ΜΜΧΑ και αυτή με την σειρά της προωθεί την ερώτηση στις άλλες μηχανές, είτε την μία μετά την άλλη ή σε όλες μαζί παράλληλα. Τα αποτελέσματα αφού συγκεντρωθούν και διαγραφούν οι διπλοεγγραφές, ομαδοποιούνται είτε σύμφωνα με την αξία τους είτε κατά ΜΧΑ.

Παρ' όλο ότι οι ΜΜΧΑ είναι ένα πολύτιμο εργαλείο για την επίλυση των παραπάνω χρονοβόρων θεμάτων, υπάρχουν αρκετά άλυτα προβλήματα όπως για παράδειγμα, η στρατηγική που θα πρέπει να ακολουθήσει ένας χρήστης για ανακαλύψει τις πληροφορίες που τον ενδιαφέρουν, ποίες λέξεις κλειδιά περιγράφουν μία έννοια, ή πως να συνδυάσει τις λέξεις κλειδιά με τους λογικούς τελεστές, ή τι να κάνει όταν τα αποτελέσματα είναι πάρα πολλά.

3.4 Συμπεράσματα

Στο κεφάλαιο αυτό πραγματοποιήθηκε μια λεπτομερής παρουσίαση των δυνατοτήτων που έχει ένας χρήστης του διαδικτύου με σκοπό την αναζήτηση και την συλλογή πληροφοριών, οι οποίες είναι σχετικές με τα ενδιαφέροντα του. Πιο συγκεκριμένα, παρουσιάσαμε τον τρόπο λειτουργίας, των Διαδικτυακών Κόμβων Αναζήτησης Πληροφοριών, οι οποίοι παρέχουν την δυνατότητα στους επισκέπτες τους, μέσω μιας διεπαφής να απευθύνουν ερωτήσεις για ένα οποιοδήποτε θέμα.

Ο τρόπος συλλογής νέων πληροφοριών και ενημέρωσης της βάσης δεδομένων των ΔΚΑΠ, τους χωρίζει σε δύο κατηγορίες, τους Θεματικούς Καταλόγους και τις Μηχανές Αναζήτησης. Οι διαφορές, τα πλεονεκτήματα καθώς και τα μειονεκτήματα

της κάθε κατηγορίας, παρουσιάζονται αναλυτικά σε σχέση με την έρευνα ενός χρήστη στο διαδίκτυο.

Ο όγκος των πληροφοριών που είναι διαθέσιμες σε διαφορετικές βάσεις δεδομένων, η απουσία ενός κεντρικού καταλόγου, η έλλειψη γνώσης του χρήστη στην χρήση των διαφορετικών διεπαφών, η επιλογή των κατάλληλων λέξεων-κλειδιών που περιγράφουν το θέμα της έρευνας, ο συνδυασμός των λέξεων με λογικούς τελεστές για την δημιουργία ερωτήσεων προς τους ΔΚΑΠ, η άντληση της λίστας των αποτελεσμάτων, η επίσκεψη του χρήστη σε κάθε ένα από τα αποτελέσματα, το φιλτράρισμα των πληροφοριών σύμφωνα με τα ενδιαφέροντα του, είναι τα προβλήματα που θα πρέπει να αντιμετωπίσει ο κάθε χρήστης κατά την διάρκεια της έρευνας του, στο διαδίκτυο.

Η δημιουργία ενός αυτόματου συστήματος, το οποίο θα αναλάμβανε το έργο της συλλογής των πληροφοριών από το διαδίκτυο και θα ήταν προσαρμοσμένο στις ιδιαιτερότητες και στα ενδιαφέροντα ενός χρήστη, θα ήταν ιδιαίτερα χρήσιμο αφού θα βοηθούσε το κάθε χρήστη ανεξαρτήτου εμπειρίας, στην επίλυση των παραπάνω χρονοβόρων και δύσκολων προβλημάτων.

Ερευνητές από τον χώρο της τεχνητής νοημοσύνης έχουν υποδείξει την ανάπτυξη έξυπνου λογισμικού με σκοπό την συλλογή πληροφοριών από μεγάλους, σύνθετους, αδόμητους και ετερογενείς πληροφοριακούς χώρους, όπως ο παγκόσμιος ιστός. Τα τελευταία χρόνια, η ιδέα ενός πράκτορα λογισμικού έχει κερδίσει έδαφος σαν ένα μέσο για την υλοποίηση τέτοιων συστημάτων που θα υποδεικνύουν και θα συμβουλεύουν τους χρήστες στην ανεύρεση και ανάκτηση χρήσιμης πληροφορίας.

Στο επόμενο κεφάλαιο θα παρουσιάσαμε την τεχνολογία των πρακτόρων λογισμικού, η οποία χρησιμοποιείται από το σύστημα Webnaut, και θα εστιάσουμε στην χρήση της σε θέματα αναζήτησης και συλλογής πληροφοριών από το διαδίκτυο.

Κεφάλαιο 4

ΠΡΑΚΤΟΡΕΣ ΛΟΓΙΣΜΙΚΟΥ ΓΙΑ ΤΗΝ ΣΥΛΛΟΓΗ ΠΛΗΡΟΦΟΡΙΩΝ ΑΠΟ ΤΟ ΔΙΑΔΙΚΤΥΟ

4.1 Εισαγωγή

Η εκθετική αύξηση των πληροφοριών που είναι διαθέσιμες στο διαδίκτυο έκανε επιτακτική την ανάγκη της ανάπτυξης και χρήσης αυτόματων εργαλείων, τα οποία θα βρίσκονται είτε σε κάποιον εξυπηρετητή είτε σε κάποιο προσωπικό υπολογιστή και σκοπό θα έχουν να ανακαλύπτουν και να φιλτράρουν πληροφορίες για λογαριασμό των χρηστών τους. Όλα αυτά τα αυτόματα προγράμματα ανήκουν στην ερευνητική περιοχή των *Πρακτόρων Λογισμικού* - ΠΛ, η οποία τα τελευταία χρόνια έλκει το ενδιαφέρον της επιστημονικής κοινότητας λόγω των προβλημάτων, προκλήσεων καθώς και των προσδοκιών που συνοδεύουν την παγκόσμια αποδοχή του διαδικτύου σαν την πλατφόρμα ενημέρωσης και διάδοσης των πληροφοριών, ηλεκτρονικού εμπορίου και διασκέδασης.

Η έννοια ενός πράκτορα λογισμικού [Bru91, Bro91, Mae89] δεν είναι νέα στο χώρο της τεχνητής νοημοσύνης, και παρ' ότι αρκετοί ερευνητές ασχολούνται με το συγκεκριμένο θέμα, η σύγχυση για το τι είναι ένας πράκτορας είναι αρκετά μεγάλη. Η σύγχυση φαίνεται και στους τίτλους των συνεδρίων που διαπραγματεύονται αντίστοιχα θέματα αλλά και στη διαφορετική ορολογία που χρησιμοποιούν αρκετοί επιστήμονες για να περιγράψουν όσο γίνεται καλύτερα αυτή την έννοια, όπως για παράδειγμα, πράκτορες ή πράκτορες δικτύου ή ευφυείς πράκτορες ή αυτόνομοι πράκτορες, ευφυείς ή προσαρμόσιμες διεπαφές, προσωπικοί βοηθοί, robot λογισμικού κ.λπ.

Όλη αυτή η σύγχυση πιστεύουμε ότι οφείλεται σε έναν κυρίως λόγο, στη ραγδαία τεχνολογική ανάπτυξη του υλικού και λογισμικού, η οποία είχε σαν αποτέλεσμα την χρήση πανίσχυρων υπολογιστικών συστημάτων, τα οποία είναι συνδεδεμένα στο διαδίκτυο και είναι σε θέση να διαχειρίζονται ταυτόχρονα διάφορες εφαρμογές πολυμέσων. Όλη αυτή η πρόοδος έθεσε ξανά από την αρχή το θέμα της διεπαφής μεταξύ ανθρώπου-μηχανής αφού οι εικόνες, το video, ο ήχος και σύντομα η

φωνή θα θεωρούνται δεδομένα σε κάθε οικιακό υπολογιστή. Ακόμα δεν θα πρέπει να ξεχνάμε και την τεράστια εξάπλωση του διαδικτύου, και κατά συνέπεια τον τεράστιο όγκο σχετικών και άσχετων πληροφοριών που δέχεται ο κάθε χρήστης από διαφορετικές πηγές π.χ. ηλεκτρονικό ταχυδρομείο, διακομιστές νέων, η οποία λειτούργησε σαν έναυσμα σε όλη αυτή την ερευνητική προσπάθεια.

Οι επιστήμονες πρέπει να προσαρμόσουν τις τεχνικές και τους αλγορίθμους που έχουν αναπτύξει στα εργαστήρια τους, όπου οι συνθήκες είναι ελεγχόμενες, και να χρησιμοποιήσουν τα εργαστηριακά αποτελέσματα ώστε να δημιουργήσουν πρακτικές και χρήσιμες εφαρμογές που θα λειτουργούν σε πραγματικές συνθήκες πάντα προς όφελος του τελικού χρήστη.

Η ανάπτυξη εφαρμογών και λειτουργικών συστημάτων, τα οποία θα κάνουν χρήση πιο εύχρηστων και φιλικών διεπαφών, έφερε κοντά, σε κοινά ερευνητικά προγράμματα, ανθρώπους οι οποίοι είχαν διαφορετικό θεωρητικό υπόβαθρο, από πολυμέσα και τρισδιάστατα γραφικά και κίνηση, μέχρι τεχνικές μηχανικής εκμάθησης, γνωστική ψυχολογία, γλωσσική επεξεργασία και κατανόηση φυσικής γλώσσας. Όλοι αυτοί οι επιστήμονες είναι φυσικό, από την δική του πλευρά ο καθένας, να χρησιμοποιούν ή να εφευρίσκουν ορολογίες οι οποίες μπορεί να περιγράψουν καλύτερα το ερευνητικό τους αντικείμενο.

Παρά την ανωτέρω σύγχυση στην ορολογία, κανένας δεν αμφισβητεί ότι τα υπολογιστικά συστήματα, τα οποία βασίζονται στην τεχνολογία των πρακτόρων, προσφέρουν μια νέα άποψη, μια νέα μεθοδολογία, με την οποία μπορούμε να προσεγγίσουμε δύσκολες εφαρμογές για τις οποίες οι υπάρχουσες τεχνικές της επιστήμης των υπολογιστών δεν προσφέρουν κάποια λύση ή αν προσφέρουν είναι αρκετά χρονοβόρα ή/και δαπανηρή [Woo97].

Επίσης η τεχνολογία των πρακτόρων αναμένεται να προσφέρει μια νέα διεπαφή σε κοινές συσκευές με τις οποίες αλληλεπιδρούμε καθημερινά με σκοπό όλες οι συσκευές να επικοινωνούν και να συνεργάζονται μεταξύ τους, πάντα προς όφελος του χρήστη τους. Μεγάλες εταιρίες όπως η Sun Microsystems , η IBM , η Microsoft αναπτύσσουν συστήματα για κατανεμημένη επεξεργασία ανάμεσα σε αλληλοσυνδεόμενες συσκευές [Γ53.a, Γ54.a, Γ55.a]. Παρ' ότι οι εταιρίες αποφεύγουν, για δικούς τους εμπορικούς λόγους, στις προδιαγραφές των συγκεκριμένων συστημάτων να αναφέρουν την έννοια πράκτορας λογισμικού, στις αρχιτεκτονικές των συστημάτων τους περιγράφονται τα πρωτόκολλα και όλη η απαραίτητη υποδομή ώστε να παρέχουν την δυνατότητα σε κοινές συσκευές να έχουν την συμπεριφορά

ενός πράκτορα [Huh99].

4.2 Τι είναι ένας Πράκτορας Λογισμικού;

Διάφορες επιστημονικές ομάδες προσπάθησαν κατά καιρούς να δώσουν έναν ορισμό για το τι είναι ένας ΠΛ και ποία τα επιθυμητά χαρακτηριστικά του. Αρκετοί ορισμοί καθώς και μια κατηγοριοποίηση των πρακτόρων και των χαρακτηριστικών τους υπάρχουν στο άρθρο των Franklin και Graesser [Fra96]. Οι ορισμοί ποικίλουν από πολύ γενικούς όπως για παράδειγμα ο ορισμός των Russell και Norvig [Rus95],

"Ένας πράκτορας είναι οτιδήποτε το οποίο δείχνει ότι αντιλαμβάνεται το περιβάλλον μέσω αισθητήρων και ενεργεί σε αυτό το περιβάλλον μέσω αντιδραστήρων"

ή πολύ ειδικούς όπως ο ορισμός των εργαστηρίων της IBM [Fra96]

"Έξυπνοι πράκτορες είναι οντότητες λογισμικού οι οποίες φέρνουν σε πέρας ορισμένες ενέργειες για λογαριασμό του χρήστη ή ενός άλλου προγράμματος με κάποιο βαθμό ανεξαρτησίας ή αυτονομίας, και το κάνουν ώστε να εμπεριέχουν γνώση ή αναπαράσταση των στόχων του χρήστη ή των επιθυμιών του."

Οι παραπάνω ορισμοί καθώς και αρκετοί άλλοι [Mae95, Smit94, Hay95] έδειξαν ότι δεν υπάρχει ένας σαφής ορισμός ο οποίος να διαχωρίζει τα προγράμματα σε πράκτορες και μη αλλά συνέβαλαν στο να συγκεντρώσουμε μερικά από τα χαρακτηριστικά και τις ιδιότητες που θα πρέπει να διαθέτει ένας ΠΛ, οι οποίες συνοψίζονται στα παρακάτω [Woo95b] :

Αυτονομία : οι πράκτορες λειτουργούν χωρίς την άμεση συμμετοχή του ανθρώπου ή άλλων, και έχουν κάποιο έλεγχο των ενεργειών τους καθώς και της εσωτερικής τους κατάστασης [Cast95].

Κοινωνική Δραστηριότητα - Ικανότητα Επικοινωνίας : οι πράκτορες μπορεί να επικοινωνούν με τους ανθρώπους ή με άλλους πράκτορες μέσω μιας γλώσσας επικοινωνίας [Gen94].

Επαναδραστηριοποίηση : οι πράκτορες δεν παρατηρούν παθητικά το περιβάλλον δράσης τους, το οποίο μπορεί να είναι, ο φυσικός κόσμος, το περιβάλλον λειτουργίας, η διεπαφή ενός προγράμματος λογισμικού, μια ομάδα από άλλους πράκτορες, το διαδίκτυο ή ένας συνδυασμός από τα προηγούμενα, αλλά ανταποκρίνονται στα ερεθίσματα-μηνύματα που συμβαίνουν σε αυτό λαμβάνοντας υπόψη τους την παράμετρο χρόνος.

Συμπεριφορά στόχου : οι πράκτορες δεν αντιδρούν μόνο στα ερεθίσματα του περιβάλλοντος αλλά επιδεικνύουν και συμπεριφορά επίτευξης στόχου με το να παίρνουν πρωτοβουλίες ώστε να επιτύχουν τους στόχους τους.

Ευελιξία : Οι ενέργειες των πρακτόρων δεν είναι προκαθορισμένες με την βοήθεια σεναρίων.

Χρονική συνέχεια : οι πράκτορες δεν δέχονται μια είσοδο και εν συνέχεια κάνουν μια σειρά από ενέργειες και τερματίζουν, αλλά είναι διαδικασίες που τρέχουν ασταμάτητα.

Όλα τα παραπάνω χαρακτηριστικά προβάλλουν την *ασθενή* έννοια του πράκτορα, αφού κάνουν αρκετά δύσκολη τη διάκριση για το τι είναι ένας πράκτορας και που διαφέρει από ένα κοινό πρόγραμμα [Fra96, Etz94, Woo95b]. Αυτό είχε σαν συνέπεια αρκετοί επιστήμονες [Sho93, Goo93, Mae93], ιδιαίτερα από τον χώρο της τεχνητής νοημοσύνης : α) να περιγράψουν τα στοιχεία που διαθέτει ένας πράκτορας με όρους που συναντάμε σε βιολογικούς οργανισμούς, όπως για παράδειγμα *πεποιθήσεις, επιθυμίες, προθέσεις* [Fis94, Rao95] και β) να προσθέσουν και άλλα χαρακτηριστικά με σκοπό να προβάλλουν μια πιο *ισχυρή* έννοια του πράκτορα. Τα επιπρόσθετα χαρακτηριστικά είναι :

Ικανότητα μεταφοράς : οι πράκτορες μπορούν να μετακινούνται από μηχανή σε μηχανή μέσα σε ετερογενή δίκτυα και να αλληλεπιδρούν με άλλους στατικούς πράκτορες ή υπηρεσίες του υπολογιστή με σκοπό να φέρουν σε πέρας το στόχο τους.

Εκμάθηση - Προσαρμοστικότητα : οι πράκτορες θα πρέπει να βασίζονται στις προηγούμενες εμπειρίες τους ώστε να προσαρμόζονται στο περιβάλλον τους.

Ειλικρίνεια - Φιλαλήθεια : οι πράκτορες δεν θα πρέπει να επικοινωνούν ή να μεταδίδουν ψευδείς πληροφορίες.
Ορθολογιστές : οι πράκτορες θα ενεργούν κατά τέτοιο τρόπο ώστε να φέρουν σε πέρας τους στόχους και δεν θα κινηθούν αντίθετα προς αυτούς, εκτός φυσικά αν αντιτίθενται προς τα πιστεύω τους.

Αληθοφανείς - Πραγματικοί Χαρακτήρες : Η *συναισθηματική* κατάσταση ενός πράκτορα θα πρέπει να ταιριάζει με την συμπεριφορά του με σκοπό να γίνεται μια *"πιστευτή" προσωπικότητα.*

Φυσικά δεν έχουν όλοι οι πράκτορες τα παραπάνω χαρακτηριστικά αλλά κατά καιρούς έχουν επιδειχτεί διάφορα πρωτότυπα τα οποία διαθέτουν ένα μεγάλο μέρος ή συνδυασμό από αυτά με σκοπό να φέρουν σε πέρας τους στόχους τους.

Τα συστήματα τα οποία βασίζονται στην τεχνολογία των πρακτόρων μπορεί να αποτελούνται από έναν ή περισσοτέρους πράκτορες. Τα τελευταία χρόνια αυξάνονται οι προσπάθειες κατασκευής συστημάτων τα οποία αποτελούνται από πράκτορες με διαφορετικές ιδιότητες ο καθένας, οι οποίοι ανταλλάσσουν μηνύματα, συνεργάζονται, γενικότερα αλληλεπιδρούν μεταξύ τους με σκοπό να επιλύσουν ένα πρόβλημα. Μια τέτοια *ανοιχτή* προσέγγιση επιτρέπει την σταδιακή αύξηση του συστήματος και την επανα-χρησιμοποίηση δοκιμασμένων τμημάτων του λογισμικού σε διαφορετικές εφαρμογές [Jos99].

4.3 Κατηγοριοποίηση των Πρακτόρων Λογισμικού

Το σύνολο των ιδιοτήτων των ΠΛ, έκανε αρκετούς επιστήμονες να προσπαθήσουν να τους ταξινομήσουν σε διάφορες κατηγορίες, βάση των ιδιοτήτων τους ή άλλων κριτηρίων [Bru91, Mul95, Fra96]. Πιστεύουμε ότι μόνο η *ικανότητα μεταφοράς* δύναται από μόνη της να χωρίσει την οικογένεια των πρακτόρων λογισμικού σε δύο μεγάλες κατηγορίες, τους *κινητούς* και τους *στατικούς*. Οι υπόλοιπες ιδιότητες μπορούν να εμφανίζονται ή όχι σε διάφορες εφαρμογές στις οποίες συμμετέχουν ΠΛ. Άρα, οι πράκτορες μπορούν να ταξινομηθούν βάσει των εργασιών που φέρουν σε πέρας ή του είδους της εφαρμογής που συμμετέχουν, όπως για παράδειγμα, συστήματα υποστήριξης αποφάσεων [Cuen99, Rac99], φιλτράρισμα - υπόδειξη

πληροφοριών [Arm95, Yan99], ηλεκτρονικό εμπόριο - δημοπρασίες [Doo97, Rod97, Mae99], διαχείριση δικτύων [Bie99, Brs99, Whi98], προσωπικοί βοηθοί [Mae94], σχεδιασμός - χρονοπρογραμματισμός - παρακολούθηση διαδικασιών - διάθεση πόρων [Atk99, Bra99, Koz93], βιομηχανικές εφαρμογές - προσομοίωση [Bra98, Dra99, Par00], ιατρική διάγνωση [Hay96, Lar98, Mik97], διασκέδαση [Ayl00, Kru97, Mae95].

Από το πλήθος και τα είδη των παραπάνω ενδεικτικών εφαρμογών καταλαβαίνουμε, ότι σχεδόν δεν υπάρχει κλάδος της επιστήμης των υπολογιστών ή υπολογιστικές εφαρμογές στις οποίες να μην χρησιμοποιούνται ή τουλάχιστον να μην διερευνώνται τα πλεονεκτήματα ή μειονεκτήματα της χρήσης των ΠΛ. Στα πλαίσια της διατριβής θα εστιάσουμε μόνο στους ΠΛ οι οποίοι λειτουργούν σαν εξατομικευμένοι βοηθοί με σκοπό να βοηθήσουν τους χρήστες σε θέματα συλλογής και φιλτραρίσματος πληροφοριών από τον παγκόσμιο ιστό.

4.3.1 Κινητοί Πράκτορες Λογισμικού

Η θεωρία των κινητών πρακτόρων λογισμικού είναι μια πολλά υποσχόμενη και καινοτομική τεχνολογία, από τη σκοπιά της ανάπτυξης ανοικτών εφαρμογών, που θα λειτουργούν σε κατανεμημένα και ετερογενή περιβάλλοντα με στόχο να ξεπεράσουν τους περιορισμούς του παραδοσιακού μοντέλου πελάτη-εξυπηρετητή [Dal97, Ches97]. Το όλο σκεπτικό είναι : αντί να μεταφέρουμε τα δεδομένα στο κέντρο επεξεργασίας να μεταφέρουμε την επεξεργασία στα δεδομένα με άμεσα οφέλη, α) την μείωση του φόρτου του δικτύου, αφού δεν είναι αναγκαία η συνεχής μεταφορά των δεδομένων, και β) την άμεση ανταπόκριση σε κρίσιμες εφαρμογές αφού ο έλεγχος της όλης εργασίας γίνεται τοπικά.

Επίσης η τεχνολογία των κινητών πρακτόρων μέσω της αυτονομίας και της ασύγχρονης λειτουργίας θα προσφέρει α) λύσεις σε κινητές συσκευές, οι οποίες θα πρέπει να έχουν συνέχεια ανοικτές συνδέσεις με ένα δίκτυο, κάτι το οποίο δεν είναι πάντα εφικτό, β) αξιοπιστία και ανοχή σε σφάλματα αφού θα είναι πλέον δυνατή η αναγγελία, η μεταφορά και η συνέχεια της εκτέλεσης των κρίσιμων λειτουργιών ενός μηχανήματος που κλείνει, σε κάποιο άλλο μηχάνημα του δικτύου.

Η μη διαδομένη χρήση των κινητών πρακτόρων οφείλεται στους παρακάτω λόγους [Cre99, Gra00] : α) τα περισσότερα συστήματα που παρουσιάζονται είναι στην καλύτερη περίπτωση ερευνητικά πρωτότυπα. Κανένα σύστημα δεν έχει

υλοποιήσει ένα ικανοποιητικό ποσοστό των λειτουργιών οι οποίες θα του επιτρέπουν μια απρόσκοπτη και αποδοτική λειτουργία, β) η συγκεκριμένη τεχνολογία είναι χρήσιμη μόνο σε ένα αρκετά δυναμικό δικτυακό περιβάλλον. Κινητές συσκευές οι οποίες είναι συνδεδεμένες στο διαδίκτυο και επιπλέον έχουν αυξημένη ισχύ για να φέρουν σε πέρας τις απαραίτητες λειτουργίες άρχισαν να εμφανίζονται μόλις τα δύο τελευταία χρόνια. γ) Δεν υπάρχουν ερευνητικά αποτελέσματα τα οποία να παρουσιάζουν πλεονεκτήματα από την χρήση της συγκεκριμένης τεχνολογίας για τις εφαρμογές οι οποίες κάνουν χρήση των παραδοσιακών τεχνικών. δ) Έλλειψη ασφάλειας στο χώρο του διαδικτύου [Cor01]. Πιο συγκεκριμένα τα θέματα ασφαλείας έχουν να κάνουν με : i) την προστασία του υπολογιστή με τις υπηρεσίες από τον κινητό πράκτορα, ii) την προστασία του κινητού πράκτορα από άλλους κινητούς πράκτορες, iii) την προστασία του κινητού πράκτορα από τον υπολογιστή με τις υπηρεσίες [Kar97].

Παρ' όλα τα παραπάνω προβλήματα, ένας κινητός πράκτορας έχει την μοναδική ικανότητα να μετακινείται από μια μηχανή σε μία άλλη, στο ίδιο δίκτυο, με άμεσο πλεονέκτημα να βρίσκεται στην ίδια μηχανή με το αντικείμενο ή την υπηρεσία που θέλει να αλληλεπιδράσει και επιπλέον δεν περιορίζεται από το σύστημα στο οποίο άρχισε την εκτέλεση της λειτουργίας του. Η χρήση της συγκεκριμένης τεχνολογίας θα βρει εφαρμογή σε αρκετά πεδία όπως για παράδειγμα, το ηλεκτρονικό εμπόριο, τη διαχείριση δικτύου, την ανάκτηση πληροφοριών, την παράλληλη επεξεργασία, τα διαγράμματα ροής εργασιών, την παρακολούθηση και άμεση ειδοποίηση, τα δίκτυα τηλεπικοινωνιών [Lan99, Cab00].

4.3.2 Στατικοί Πράκτορες Λογισμικού

Ένας *στατικός* πράκτορας λογισμικού αρχίζει και τελειώνει όλες τις εργασίες του, στο ίδιο μηχάνημα που άρχισε η εκτέλεση του. Στην περίπτωση που ο πράκτορας, για να φέρει σε πέρας τους στόχους του, έχει ανάγκη από δεδομένα ή υπηρεσίες ή άλλες πληροφορίες που δεν υπάρχουν στο ίδιο μηχάνημα, θα χρησιμοποιήσει μια από τις *παραδοσιακές* τεχνικές επικοινωνίας, δηλαδή είτε κλήση διαδικασιών εξ αποστάσεως είτε ανταλλαγή μηνυμάτων μέσω κάποιου κοινού κώδικα ή πρωτοκόλλου επικοινωνίας.

4.4 Η Τεχνολογία των Πρακτόρων Λογισμικού για την Συλλογή Πληροφοριών από το Διαδίκτυο

4.4.1 Φιλτράρισμα και Συλλογή Πληροφοριών στο Διαδίκτυο

Το φιλτράρισμα των πληροφοριών δεν είναι μια εργασία η οποία προέκυψε λόγω διαδικτύου αλλά είναι γνωστή σε όλους τους ανθρώπους που ασχολούνται με το γραπτό λόγο. Για παράδειγμα, κατά την ανάγνωση μιας εφημερίδας, ενός περιοδικού ή βιβλίου θα *πρέπει να σαρώσουμε* το κείμενο για να *εξαγάγουμε* μόνο τις πληροφορίες που μας ενδιαφέρουν και να απορρίψουμε όσο το δυνατόν πιο γρήγορα όλα τα άσχετα και αδιάφορα κείμενα.

Η πληθώρα των υπηρεσιών του διαδικτύου αλλά και των πληροφοριών που δέχεται καθημερινά ένας χρήστης δημιούργησε την ανάγκη για αυτόματα συστήματα τα οποία θα κάνουν Φιλτράρισμα Πληροφοριών (Information Filtering) - ΦΠ. Η συγκεκριμένη εργασία αν και είναι διαφορετική όμως πολύ συχνά συγχέεται με την εργασία της Ανάκτησης των Πληροφοριών (Information Retrieval) - ΑΠ. Αυτή η σύγχυση ίσως να οφείλεται στο ότι και οι δύο εργασίες σκοπό έχουν να προμηθεύσουν το χρήστη με σχετικές και χρήσιμες πληροφορίες ως προς τις πληροφοριακές ανάγκες του. Οι δύο πιο σημαντικές διαφορές [Belk92] στις δύο εργασίες και κατ' επέκταση στα συστήματα που τις υποστηρίζουν είναι :

1. Οι πηγές άντλησης των πληροφοριών ενός συστήματος ΑΠ είναι στατικές, π.χ. μια βάση δεδομένων. Εν αντιθέσει σε ένα σύστημα ΦΠ οι πηγές άντλησης των πληροφοριών είναι δυναμικές π.χ. η ηλεκτρονική αλληλογραφία, τα περιεχόμενα ενός ηλεκτρονικού περιοδικού κ.λπ.

2. Ένα σύστημα ΑΠ σκοπό έχει να ικανοποιήσει την βραχυπρόθεσμη ανάγκη του χρήστη για πληροφορίες. Αυτό σημαίνει ότι ο χρήστης αλληλεπιδρά με ένα σύστημα που δέχεται τις ερωτήσεις του, και ανταποκρίνεται σε κάθε ερώτηση ξεχωριστά, δηλαδή, η προηγούμενη ερώτηση δεν έχει καμία σχέση με την τρέχουσα. Εν αντιθέσει ένα σύστημα ΦΠ προσπαθεί να ικανοποιήσει την μακροπρόθεσμη ανάγκη του χρήστη για πληροφορίες. Αυτό σημαίνει ότι το σύστημα ερευνά σε βάθος χρόνου για τα ενδιαφέροντα τα οποία διέπουν τις ερωτήσεις του χρήστη και προσπαθεί να ικανοποιήσει αυτές τις πληροφοριακές

ανάγκες.

Η κούραση λόγω υπερβολικής ενημέρωσης, που αισθάνεται ένας άνθρωπος που κάνει χρήση των υπηρεσιών του διαδικτύου, οφείλεται κυρίως σε δύο λόγους [Jen98]:

α) Στο *φιλτράρισμα της πληροφορίας*, όπου κάποιος δέχεται συνεχώς μεγάλες ποσότητες πληροφοριών από διαφορετικές πηγές (π.χ. ηλεκτρονική αλληλογραφία και νέα) αλλά μόνο ένα πολύ μικρό μέρος από αυτές πληροφορίες είναι σχετικές με τα ενδιαφέροντα του. Έτσι λοιπόν θα πρέπει να *σαρώσει* όλες τις πληροφορίες, να απορρίψει τις άσχετες και να κρατήσει μόνο τις σχετικές προς τα ενδιαφέροντα του.

β) Στην *συλλογή πληροφοριών*, όπου κάποιος πρέπει να επισκεφτεί αρκετούς κόμβους στο διαδίκτυο ώστε να βρει σχετικές πληροφορίες με τα ενδιαφέροντα του.

Όπως αναφέραμε στο προηγούμενο κεφάλαιο, στον παγκόσμιο ιστό οι δύο παραπάνω εργασίες είναι αλληλοσυνδεόμενες, αφού οι περισσότεροι χρήστες χρησιμοποιούν τις μηχανές αναζήτησης, που βρίσκονται στο διαδίκτυο για την ανάκτηση πληροφοριών. Οι ερωτήσεις των χρηστών αποτελούνται από μερικές λέξεις κλειδιά ή φράσεις, οι οποίες είναι σχετικές με το θέμα της έρευνάς τους.

Από την πλευρά τους, οι μηχανές αναζήτησης χρησιμοποιούν τα δικά τους κριτήρια για την καταλογογράφηση των κειμένων. Έτσι λοιπόν ο επισκέπτης μιας μηχανής αναζήτησης, θα πρέπει να χρησιμοποιήσει τη σωστή ορολογία για να φτάσει στο επιθυμητό αποτέλεσμα. Βέβαια θα πρέπει πάντα να είναι προετοιμασμένος να επισκεφτεί από δεκάδες μέχρι μερικές εκατοντάδες αποτελέσματα, χωρίς να ή μέχρι να εντοπίσει την επιθυμητή πληροφορία. Όπως θα δούμε στην επόμενη παράγραφο οι πράκτορες λογισμικού μπορούν να μειώσουν το φόρτο εργασίας ενός χρήστη εκτελώντας εργασίες χωρίς την άμεση συμμετοχή του χρήστη.

4.4.2 Στόχοι των Πρακτόρων Λογισμικού για Συλλογή Πληροφοριών από το Διαδίκτυο

Αρκετές ερευνητικές προσπάθειες, είτε σε ακαδημαϊκά ιδρύματα είτε σε εργαστήρια εταιριών, έχουν σαν στόχο την ανάπτυξη πρακτόρων λογισμικού για την υποστήριξη των χρηστών του διαδικτύου, σε θέματα σχετικά με την μείωση του πληροφοριακού

φόρτου. Αυτοί οι πράκτορες θα πρέπει να πληρούν μια ή περισσότερες από τις παρακάτω προϋποθέσεις [Klu01] :

Ανάκτηση και διαχείριση πληροφοριών - Ο πράκτορας θα πρέπει να είναι σε θέση να αντλεί πληροφορίες από διαφορετικές πηγές, να τις αναλύει, να φιλτράρει τα δεδομένα, να παρακολουθεί και να ενημερώνει τον χρήστη ή άλλον πράκτορα για τυχόν αλλαγές σε συγκεκριμένες πηγές πληροφοριών .

Σύνθεση και Αναπαράσταση πληροφοριών - Ο πράκτορας θα πρέπει να είναι σε θέση να συνθέτει δεδομένα από διαφορετικές πηγές και να παρουσιάζει ομαδοποιημένα, τις πολυδιάστατες όψεις των σχετικών, με τα ενδιαφέροντα του χρήστη, πληροφοριών.

Έξυπνη Βοήθεια - Ο πράκτορας θα πρέπει να προσαρμόζεται σε όλες τις τυχόν αλλαγές που συμβαίνουν είτε στα ενδιαφέροντα του χρήστη είτε στις πληροφορίες είτε στο δικτυακό περιβάλλον. Επίσης, ο πράκτορας θα πρέπει να αλληλεπιδρά με τον χρήστη, ώστε να τον υποστηρίζει σε όλο το φάσμα των ενεργειών του στο διαδίκτυο. Για αυτό το σκοπό, η χρήση έξυπνων διεπαφών ή ανθρωπόμορφων χαρακτήρων μπορεί να αυξήσει όχι μόνο την αμεσότητα επικοινωνίας του χρήστη με τον πράκτορα αλλά και την ευκολία με την οποία ο χρήστης αλληλεπιδρά με την ίδια την πληροφορία.

Πιο συγκεκριμένα, οι πράκτορες λογισμικού θα μπορούσαν να επισκεφτούν τις πηγές πληροφοριών στο διαδίκτυο και να σχηματίσουν ερωτήσεις οι οποίες σχετίζονται με τα ενδιαφέροντα του χρήστη. Εν συνεχεία, θα μπορούσαν αυτόνομα να επισκεφτούν τις σελίδες των αποτελεσμάτων, να συλλέξουν και να παρουσιάσουν μόνο εκείνες τις σελίδες που ταιριάζουν με τα ενδιαφέροντα του χρήστη.

Επίσης οι πράκτορες λογισμικού μπορούν να συνεχίσουν την έρευνα ακόμα και όταν ο χρήστης έχει ανακαλύψει την επιθυμητή πληροφορία με σκοπό να εντοπίσουν διάφορες τροποποιήσεις, αλλαγές, νέα στοιχεία που έχουν προκύψει σε σχέση πάντα με την τελευταία ενημέρωση του χρήστη. Ακόμα μπορούν να γενικεύσουν την έρευνα τους χρησιμοποιώντας ένα λεξικό κατά δύο τρόπους : α) είτε κατά την σύνταξη της ερώτησης δοκιμάζοντας διαφορετικές λέξεις με το ίδιο νόημα, β) είτε κατά το στάδιο της ανάκτησης των αποτελεσμάτων, αφού οι λέξεις που

ανήκουν στην ίδια έννοια θα έχουν την ίδια αξιολόγηση.

Είναι κατανοητό ότι η έλλειψη ενός κεντρικού καταλόγου για τα περιεχόμενα του διαδικτύου δημιουργεί αρκετά προβλήματα στον εντοπισμό πληροφοριών. Οι μηχανές αναζήτησης είτε για εμπορικούς λόγους είτε για άλλους λόγους δεν συνεργάζονται μεταξύ τους με αποτέλεσμα : α) να δημιουργείται τεράστιος φόρτος στο διαδίκτυο αφού η καθεμία δημιουργεί το δικό της κατάλογο β) να μην υπάρχει μια κοινά αποδεκτή ορολογία στην διεπαφή της αποστολής ερωτήσεων γ) να μην υπάρχει μια κοινή μορφοποίηση στην εμφάνιση των αποτελεσμάτων ώστε να μην έχει την δυνατότητα ο χρήστης να δημιουργήσει μια προσωπική σελίδα με τα αποτελέσματα από διαφορετικές πηγές πληροφοριών.

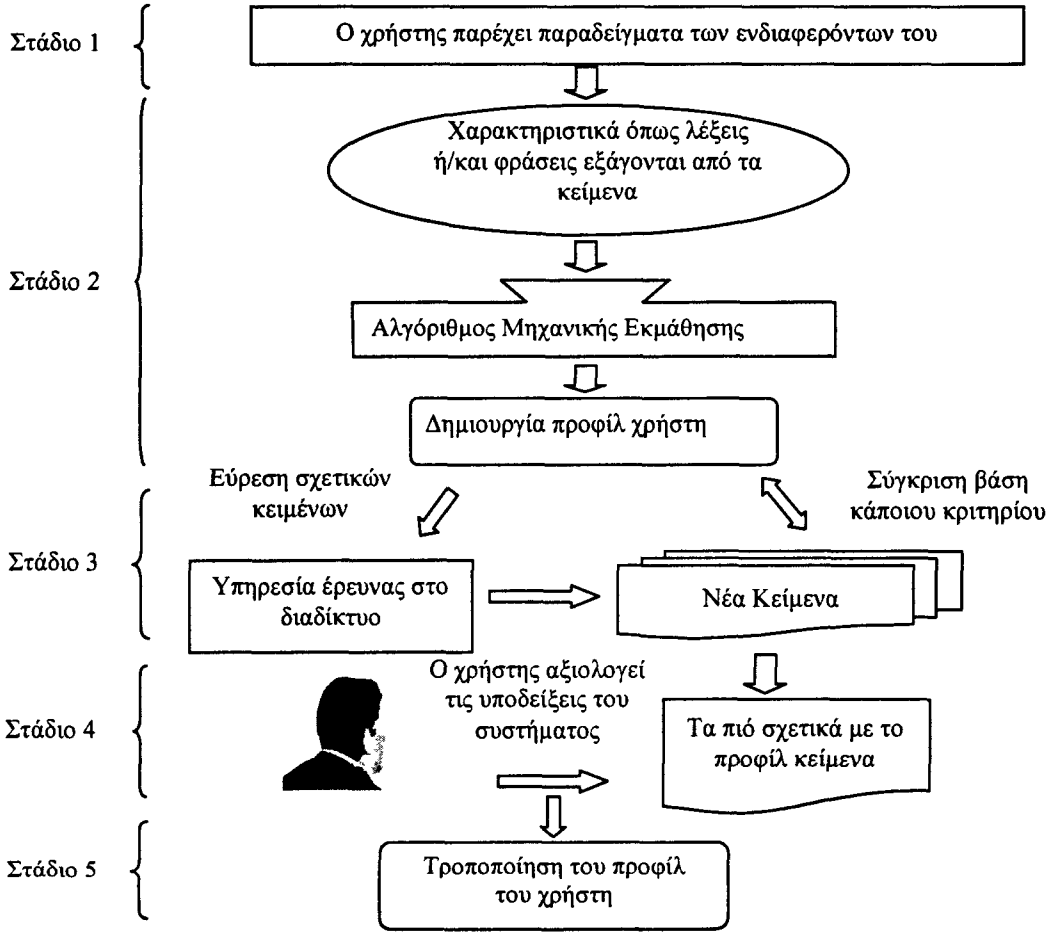
Οι ευφυείς πράκτορες θα μπορέσουν να επιλύσουν μερικά από τα παραπάνω προβλήματα δίνοντας την δυνατότητα στο χρήστη να δηλώσει τις πηγές που θα ήθελε να αντλεί δεδομένα και εν συνεχεία μόνοι τους ή σε συνεργασία με άλλους πράκτορες να συγκεντρώσουν τις πληροφορίες. Η ανταλλαγή μηνυμάτων μέσω ενός κοινού πρωτοκόλλου θα οδηγήσει τους πράκτορες στον εντοπισμό των διακομιστών που διαθέτουν τις πληροφορίες που ενδιαφέρουν τον χρήστη τους. Αυτά τα μηνύματα θα περιλαμβάνουν οδηγίες για το τρόπον που δέχεται τις ερωτήσεις ο συγκεκριμένος διακομιστής αλλά και για το τρόπον εξαγωγής των στοιχείων που συνθέτουν την πληροφορία από τις μορφοποιημένες σελίδες των αποτελεσμάτων.

4.4.3 Τα Στάδια Λειτουργίας ενός Πράκτορα Λογισμικού για Συλλογή Πληροφοριών από το Διαδίκτυο

Το διαδίκτυο προσφέρει πολλές δυνατότητες όσον αφορά την διάδοση των πληροφοριών και για αυτό το λόγο το αντιμετωπίζουμε σαν μια τεράστια δεξαμενή πληροφοριών. Όμως δεν ενδιαφέρουν όλες οι πληροφορίες όλους τους χρήστες του διαδικτύου για αυτό το λόγο θα πρέπει ο κάθε χρήστης να φιλτράρει τις πληροφορίες που δέχεται είτε μόνος του είτε με την βοήθεια άλλων χρηστών είτε με την βοήθεια ενός αυτόματου προγράμματος.

Ο φόρτος εργασίας είναι αρκετά μεγάλος ειδικά στην περίπτωση όπου ο χρήστης προσπαθεί μόνος του να φέρει σε πέρας το έργο της εύρεσης πληροφοριών και εν συνεχεία το φιλτράρισμα αυτών των πληροφοριών. Το πλεονέκτημα σε αυτή την περίπτωση είναι ότι ο χρήστης θα βρει τις πληροφορίες που τον ενδιαφέρουν την δεδομένη στιγμή.

Όπως στην καθημερινή μας ζωή έτσι και στο διαδίκτυο οι άνθρωποι δημιουργούν κοινότητες για την ανταλλαγή απόψεων, επίλυση προβλημάτων κ.λπ. Έτσι λοιπόν ένας χρήστης του διαδικτύου ο οποίος ενδιαφέρεται για ένα συγκεκριμένο θέμα έχει την δυνατότητα είτε να δημιουργήσει ένα δικτυακό τόπο με σκοπό να προσελκύσει άλλους ανθρώπους οι οποίοι έχουν τα ίδια ενδιαφέροντα είτε να εγγραφεί ο ίδιος σε έναν από τους υπάρχοντες διαδικτυακούς τόπους που διαπραγματεύονται σχετικά θέματα.



Σχήμα 4.1 *Τα στάδια λειτουργίας ενός έξυπνου βοηθού.*

Στην συλλογική περίπτωση ο φόρτος εργασίας του φιλτραρίσματος μειώνεται σημαντικά γιατί όλη η δεξαμενή πληροφοριών είναι σχετική με τα ενδιαφέροντα του χρήστη. Επιπλέον υπάρχει ένα πλήθος ανθρώπων που ερευνά για σχετικά θέματα σε άλλους δικτυακούς τόπους και ανακοινώνει τα νέα ευρήματα στην κοινότητα. Βέβαια πάντα ο τελικός κριτής είναι ο ίδιος ο χρήστης γιατί παρ' όλο ότι είναι μέλος σε μια ομάδα αυτό δεν σημαίνει ότι τα ενδιαφέροντα αλλά και το επίπεδο γνώσης του

θέματος είναι ίδιο για όλα τα μέλη.

Η συλλογή κειμένων από τον παγκόσμιο ιστό με την χρήση ενός αυτόματου προγράμματος δεν είναι εύκολο θέμα και για αυτό το λόγο ένας αριθμός από συστήματα έχουν αναπτυχθεί, τα οποία εμπεριέχουν τεχνικές από το χώρο της Τεχνητής Νοημοσύνης με σκοπό να είναι όσο το δυνατόν πιο αξιόπιστα και χρήσιμα στο χρήστη. Σε γενικές γραμμές, οι έξυπνοι βοηθοί λειτουργούν σε πέντε στάδια όπως εμφανίζεται στο σχέδιο 4.1.

Στο πρώτο στάδιο, το σύστημα συλλέγει όλα τα δεδομένα που σχετίζονται με τις προτιμήσεις του χρήστη όπως τι του αρέσει και τι όχι. Αυτά τα δεδομένα μπορούν να αποκτηθούν είτε παρατηρώντας την συμπεριφορά του χρήστη κατά την διάρκεια μιας περιήγησης στο διαδίκτυο είτε από τον ίδιο τον χρήστη ο οποίος μπορεί να καθορίσει επακριβώς μερικά παραδείγματα από τα ενδιαφέροντα του. Και στις δύο ανωτέρω περιπτώσεις, κατά την διάρκεια του δεύτερου σταδίου, τα χαρακτηριστικά από αυτά τα δεδομένα, όπως οι κοινές λέξεις και/ή φράσεις, εξάγονται και χρησιμοποιούνται σαν είσοδος σε έναν αλγόριθμο μηχανικής εκμάθησης με σκοπό να δημιουργηθεί το προφίλ του χρήστη.

Στο τρίτο στάδιο, μετά την δημιουργία του προφίλ, ο πράκτορας θα προσπαθήσει να ανακαλύψει παρόμοια κείμενα σχετικά με το προφίλ. Με σκοπό να χειριστεί αυτό το θέμα, θα πρέπει να υπάρχει ένας μηχανισμός έρευνας στο Web και πιο διαδεδομένος τρόπος είναι η χρησιμοποίηση μιας μηχανής ή μεταμηχανής αναζήτησης. Στην επόμενη φάση, τα νέα κείμενα συγκρίνονται με το προφίλ του χρήστη και τα πιο σχετικά παρουσιάζονται στο χρήστη. Η αξιολόγηση των νέων κειμένων από τον χρήστη, ως προς το πόσο σχετικά είναι τα νέα κείμενα χρησιμοποιείται από τον πράκτορα για την τροποποίηση του προφίλ του χρήστη και η ίδια λειτουργία συνεχίζεται στο στάδιο 3.

Η παραπάνω λειτουργία περιγράφει έναν προσωπικό έξυπνο βοηθό που σκοπό έχει να υποστηρίξει τις πληροφοριακές ανάγκες ενός χρήστη. Μια σημαντική διαφοροποίηση στο παραπάνω σχήμα ανάμεσα στις διάφορες εφαρμογές, είναι στο τρίτο στάδιο, όταν ο πράκτορας δεν συλλέγει κείμενα από το web αλλά από έναν κατάλογο που έχει χτιστεί από τις υποδείξεις άλλων χρηστών με παρόμοια ενδιαφέροντα. Σε αυτή την περίπτωση ο όρος φιλτράρισμα σε συνεργασία (*collaborative filtering)* περιγράφει το αντικείμενο.

Σε οποιαδήποτε από τις παραπάνω προσεγγίσεις ένας μεγάλος αριθμός από αλληλοεπιδράσεις με το χρήστη είναι απαραίτητες μέχρι το προφίλ να γίνει μια

αντανάκλαση των πραγματικών ενδιαφερόντων του. Επίσης τα πλεονεκτήματα καθώς και τα μειονεκτήματα μοιράζονται εξίσου στις δύο προσεγγίσεις και καμία από αυτές δεν είναι η καλύτερη για όλα τα περιβάλλοντα και τις εφαρμογές. Για παράδειγμα, σε μία προσέγγιση χωρίς συνεργασία, ένας έξυπνος πράκτορας μπορεί να χρειαστεί περισσότερο χρόνο να ανακαλύψει τα ενδιαφέροντα του χρήστη επειδή στερείται τις υποδείξεις από τους άλλους χρήστες. Από την άλλη πλευρά, στην προσέγγιση που μαθαίνει το προφίλ ενός και μόνο χρήστη έχει το πλεονέκτημα να αναγνωρίζει τα ιδιαίτερα ενδιαφέροντα του όταν οι κοινότητες από άλλους χρήστες δεν έχουν τις ίδιες ανησυχίες, το ίδιο επίπεδο εμπειρίας ή ο θεματικός κατάλογος δεν είναι αρκετά πλούσιος.

Επιπροσθέτως, εκτός από την ανωτέρω διαφοροποίηση, οι έξυπνοι βοηθοί διαφοροποιούνται και στο δεύτερο στάδιο και πιο συγκεκριμένα στην δημιουργία του προφίλ του χρήστη. Αρκετές μέθοδοι ανάκτησης πληροφορίας, αλγόριθμοι μηχανικής εκμάθησης και ευριστικοί κανόνες έχουν εφαρμοστεί για να χειριστούν αυτό το θέμα.

4.4.4 Τεχνικές Ανάπτυξης των Πρακτόρων Λογισμικού για Συλλογή Πληροφοριών από το Διαδίκτυο

Οι πιο διαδεδομένες τεχνικές για την ανάπτυξη πρακτόρων, οι οποίοι θα δραστηριοποιούνται σε πληροφοριακούς χώρους για λογαριασμό του χρήστη, ανήκουν σε μία από τις παρακάτω κατηγορίες :

Προγραμματισμός από το χρήστη - Η συμπεριφορά και η λειτουργία του πράκτορα προγραμματίζεται από το χρήστη με την βοήθεια κάποιας γλώσσας προγραμματισμού ή κανόνων.

Προκαθορισμένη γνώση - Η συμπεριφορά και η λειτουργία του πράκτορα έχει κωδικοποιηθεί από έναν μηχανικό γνώσης, ο οποίος έχει συγκεντρώσει όλη την απαραίτητη γνώση για το συγκεκριμένο πεδίο δράσης του πράκτορα.

Μηχανική Εκμάθηση - Ο πράκτορας χρησιμοποιεί μεθόδους και τεχνικές από το χώρο της Μηχανικής Εκμάθησης με σκοπό να συγκεντρώσει πληροφορίες για τον χρήστη ώστε να τον βοηθήσει στο συγκεκριμένο πεδίο δράσης.

Μερικά ενδεικτικά συστήματα που έχουν αναπτυχθεί, με μια ή συνδυασμό από τις παραπάνω τεχνικές, και λειτουργούν, είτε σε συνεργασία με άλλη εφαρμογή είτε αυτόνομα, στο χώρο του διαδικτύου για λογαριασμό του χρήστη, με σκοπό να του μειώσουν το φόρτο εργασίας λόγω υπερβολικής ενημέρωσης, είναι τα παρακάτω :

Το πρόγραμμα *Oval* [Mal95], το οποίο είναι η συνέχεια του προγράμματος *Object Lens* [Lai88], σκοπό έχει να βοηθήσει τον χρήστη στην περιγραφή και επίλυση των προβλημάτων τα οποία είναι σχετικά με την διανομή και ανταλλαγή των ηλεκτρονικών πληροφοριών. Το σύστημα αποτελείται από 4 δομικά στοιχεία : α) τα αντικείμενα - τα οποία αναπαριστούν αντικείμενα του περιβάλλοντος π.χ. ανθρώπους, εργασίες, μηνύματα, συναντήσεις, β) όψεις - οι οποίες παρουσιάζουν περιληπτικά τις συλλογές από αντικείμενα, π.χ. ένας χρήστης μπορεί να επιλέξει την επίδειξη ορισμένων μόνο πεδίων από μια συλλογή αντικειμένων ή την επίδειξη των σχέσεων μεταξύ των αντικειμένων, γ) πράκτορες - οι οποίοι βασίζονται σε κανόνες και εκπληρώνουν εργασίες χωρίς την άμεση συμμετοχή του χρήστη. Οι κανόνες περιέχουν περιγραφές για τα αντικείμενα στα οποία θα επιδράσουν καθώς επίσης και τις δυνατές ενέργειες στα αντικείμενα δ) σύνδεσμοι - οι οποίοι αναπαριστούν σχέσεις ανάμεσα στα αντικείμενα. Κατ' αυτόν τον τρόπο είναι δυνατή η δημιουργία μιας ταχυδρομικής λίστας, η περιγραφή της επαγγελματικής σχέσης μεταξύ των ανθρώπων, ή της σχέσης μεταξύ μηνυμάτων και των απαντήσεων.

Κάνοντας χρήση των παραπάνω βασικών στοιχείων, ο χρήστης του συστήματος μπορεί να κατασκευάσει ένα σύνολο από κανόνες, με σκοπό την ταξινόμηση των εισερχόμενων ηλεκτρονικών μηνυμάτων σε διαφορετικούς φακέλους ή το φιλτράρισμα των εισερχόμενων μηνυμάτων ανάλογα με τον αποστολέα τους, το θέμα τους ή τα περιεχόμενα τους, την άμεση απάντηση σε ένα μήνυμα.

Η εφαρμογή *Stanford Information Filtering Tool* [Yan99], φιλτράρει μηνύματα από υπηρεσίες συζητήσεων και άρθρα από ταχυδρομικές λίστες, σύμφωνα με το προφίλ του χρήστη. Ο χρήστης αφού δημιουργήσει μόνος του το προφίλ του, εν συνεχεία το στέλνει στον πράκτορα, μέσω του ηλεκτρονικού ταχυδρομείου ή με την βοήθεια μιας φόρμας HTML. Το προφίλ του χρήστη μπορεί να αποτελείται, είτε από λέξεις και τους λογικούς τελεστές, της ένωσης και της άρνησης, είτε μόνο από λέξεις.

Στην πρώτη περίπτωση, γίνεται επακριβώς ταύτιση των λέξεων στα άρθρα, με τις λέξεις και τις σχέσεις που συνθέτουν το προφίλ, ενώ στην δεύτερη περίπτωση,

χρησιμοποιείται η συχνότητα εμφάνισης των λέξεων στα άρθρα. Ο πράκτορας θεωρεί ότι όλες οι λέξεις, όπου και αν εμφανίζονται σε ένα άρθρο, έχουν την ίδια αξία. Μια τιμή κατώφλι από μηδέν έως ένα, η οποία είναι προκαθορισμένη από το σύστημα σε περίπτωση που δεν την έχει υποδείξει ο χρήστης, δηλώνει την επιθυμητή ομοιότητα που θα πρέπει να έχει το άρθρο της συλλογής, με το προφίλ του χρήστη.

Ο χρήστης, σε καθημερινή βάση, παραλαμβάνει μία ταξινομημένη λίστα από είκοσι άρθρα, τα οποία έχει φιλτράρει ο πράκτορας, και θεωρούνται από όλη την συλλογή, σαν τα πιο σχετικά με το προφίλ του. Σε οποιοδήποτε στάδιο, ο χρήστης έχει την δυνατότητα να τροποποιήσει το προφίλ του ή να δημιουργήσει πολλά προφίλ με σκοπό να καλύψει τις πληροφοριακές ανάγκες του. Η ανάπτυξη αλγορίθμων ομαδοποίησης των προφίλ των χρηστών, βοήθησε την απόδοση του συστήματος τόσο από πλευράς ταχύτητας όσο και από φόρτο εργασίας.

Το πρόγραμμα *Maxims* [Mae94] απευθύνεται στους χρήστες του ηλεκτρονικού ταχυδρομείου. Ο σκοπός του είναι να βοηθήσει τον χρήστη σε θέματα όπως, κατηγοριοποίηση της αλληλογραφίας βάσει της προτεραιότητας, διαγραφή, αρχειοθέτηση, προώθηση των μηνυμάτων. Ο πράκτορας παρακολουθεί και μαθαίνει τις ενέργειες του χρήστη καθώς αυτός αλληλεπιδρά με το πρόγραμμα που χρησιμοποιεί για την ηλεκτρονική του αλληλογραφία. Πιο συγκεκριμένα, καταγράφονται οι ενέργειες του χρήστη σε σχέση πάντα με τα χαρακτηριστικά των μηνυμάτων, όπως για παράδειγμα, τον αποστολέα, τον παραλήπτη, την λίστα των παραληπτών, την κατάσταση του μηνύματος (έχει γίνει ανάγνωση ή όχι), το θέμα, αν είναι απάντηση σε προηγούμενο μήνυμα κ.λπ.

Εν συνεχεία με την παραλαβή ενός νέου μηνύματος, ο πράκτορας κάνει μια πρόβλεψη για το τι θα έκανε ο χρήστης στη συγκεκριμένη περίπτωση. Τα χαρακτηριστικά του νέου μηνύματος, συγκρίνονται με τα χαρακτηριστικά των μηνυμάτων που έχει σαν παραδείγματα στην μνήμη του, με σκοπό να βρει την ακριβή αντιστοιχία ή παραπλήσια (γειτονική) κατάσταση.

Αν ο πράκτορας είναι αρκετά σίγουρος για την επιτυχία της πρόβλεψης, τότε και μόνο τότε εκτελεί την ενέργεια, διαφορετικά κάνει την υπόδειξη του και περιμένει την ενέργεια του χρήστη. Η μηχανή εκμάθησης του πράκτορα είναι αρκετά γενική και είναι εύκολη η προσαρμογή της σε εφαρμογές όπου διαχειρίζονται επαναλαμβανόμενες εργασίες, οι οποίες διαφοροποιούνται από τους περιορισμούς ή τις προσωπικές συνήθειες του χρήστη, όπως για παράδειγμα στον

χρονοπρογραμματισμό συσκέψεων [Koz93, Mae94].

Ο πράκτορας *SwiftFile* [Seg00] βοηθάει τους χρήστες του προγράμματος Lotus Notes να οργανώσουν τα μηνύματα του ηλεκτρονικού ταχυδρομείου σε φακέλους. Ο πράκτορας χρησιμοποιεί έναν λεξικό ταξινομητή για να μάθει τις συνήθειες του χρήστη και εν συνεχεία να προβλέψει σε ποιόν φάκελο θα τοποθετήσει ο χρήστης τα εισερχόμενα μηνύματα. Οι προβλέψεις του πράκτορα εμφανίζονται στο χρήστη με την μορφή συντομεύσεων, για τους τρεις επικρατέστερους φακέλους. Αν η πρόβλεψη του πράκτορα είναι σωστή, τότε η προσπάθεια για την αρχειοθέτηση ενός μηνύματος, γίνεται απλά με το πάτημα στο κουμπί της συντόμευσης.

Αρχικά, το προφίλ δημιουργείται από τις λέξεις που υπάρχουν στα μηνύματα που έχει ταξινομήσει ο χρήστης στους υπάρχοντες φακέλους. Οι λέξεις που εμπεριέχονται τόσο στα εισερχόμενα όσο και στα μηνύματα που διαγράφονται, χρησιμοποιούνται από τον πράκτορα για να προσαρμόσει το προφίλ του χρήστη, στα νέα ενδιαφέροντα του.

Ο πράκτορας *WebWatcher* [Joa97], λειτουργεί σαν *τουριστικός ξεναγός* σε ένα δικτυακό κόμβο, ακολουθώντας το χρήστη κατά την διάρκεια της πλοήγησης του, με σκοπό να του προτείνει το καλύτερο μονοπάτι που θα τον οδηγήσει σε σελίδες που είναι σχετικές με τα ενδιαφέροντα του. Αρχικά, ο πράκτορας απαιτεί από το χρήστη, τον στόχο του, ο οποίος περιγράφεται από λέξεις κλειδιά, καθώς και την ηλεκτρονική διεύθυνση αλληλογραφίας του, με σκοπό να τον ενημερώσει για τυχόν αλλαγές στα κείμενα που είναι σύμφωνα με τα ενδιαφέροντα του. Εν συνεχεία, ο πράκτορας ενσωματώνει στο πάνω μέρος της σελίδας που επισκοπεί ο χρήστης, στο πρόγραμμα πλοήγησης, ένα μενού από εντολές καθώς και υποδείξεις, με την μορφή συνδέσμων, που θα οδηγήσουν τον χρήστη στο στόχο του.

Όλες τις ενέργειες του χρήστη καταγράφονται σε ένα αρχείο λειτουργίας στον εξυπηρετητή. Οι υποδείξεις βασίζονται στην τρέχουσα σελίδα που επισκοπεί ο χρήστης, στην ανάδραση του χρήστη σε προηγούμενες υποδείξεις καθώς και σε επιτυχημένες διαδρομές προηγούμενων χρηστών που είχαν παρόμοια ενδιαφέροντα.

Ο πράκτορας *Learning Information Retrieval Agent* [Balv95], λειτουργεί σε έναν εξυπηρετητή, όπως και ο πράκτορας *WebWatcher*, αλλά αντίθετα από αυτόν δεν προτείνει μια διαδρομή αλλά κείμενα τα οποία είναι σύμφωνα με τα ενδιαφέροντα

του χρήστη. Ο χρήστης δίνει στον πράκτορα μια λίστα με διευθύνσεις που τις θεωρεί ενδιαφέρουσες, με σκοπό να δημιουργηθεί ένα αρχικό προφίλ. Ο πράκτορας αναπαριστά τόσο τα κείμενα όσο και το προφίλ του χρήστη με τεχνικές από το χώρο της ανάκτησης πληροφοριών. Η σύγκριση του προφίλ του χρήστη με τα κείμενα τα οποία προκύπτουν από την έρευνα του πράκτορα, με την μέθοδο του βέλτιστου κόμβου, θα δημιουργήσει μια λίστα από ενδιαφέρουσες διευθύνσεις. Η άμεση βαθμολογία του χρήστη στις υποδείξεις του συστήματος θα χρησιμοποιηθεί για την τροποποίηση του προφίλ του, και κατά συνέπεια στην βελτίωση της απόδοσης του πράκτορα.

Η επέκταση της εφαρμογής ώστε να επιτρέπει την συλλογή πληροφοριών με την συνεργασία των χρηστών, ονομάζεται *Fab* [Balv97, Balv98]. Διαφορετικά είδη πρακτόρων ερευνούν τον παγκόσμιο ιστό και συγκεντρώνουν τα αποτελέσματα τους σε έναν κεντρικό εξυπηρετητή, όπου εκεί είναι αποθηκευμένα και τα προφίλ όλων των χρηστών. Ο προσωπικός πράκτορας του κάθε χρήστη, ερευνά τα κείμενα στον εξυπηρετητή καθώς και τα προφίλ των άλλων χρηστών που μοιράζονται κοινά ενδιαφέροντα, ώστε να προτείνει τα κείμενα τα οποία έχουν την καλύτερη βαθμολογία.

Ο πράκτορας *Letizia* [Lie97] παρατηρεί την συμπεριφορά του χρήστη κατά την διάρκεια της πλοήγησης του στο διαδίκτυο και προτείνει ηλεκτρονικές διευθύνσεις που υπάρχουν στην τρέχουσα σελίδα. Ο πράκτορας δεν γνωρίζει το στόχο του χρήστη, όπως στην εφαρμογή *WebWatcher*, αλλά υποθέτει τα ενδιαφέροντα του χρήστη από την μέχρι τώρα στρατηγική πλοήγησης του, δηλαδή από τις διευθύνσεις που ακολούθησε για να βρεθεί στην τρέχουσα σελίδα. Οι διευθύνσεις που προτείνει ο πράκτορας, προέρχονται από την έρευνα με την μέθοδο του βέλτιστου κόμβου που την υλοποιεί, αρχίζοντας από τις διευθύνσεις της τρέχουσας σελίδας και συνεχίζοντας στα επόμενα επίπεδα τους. Οι υποδείξεις περιέχουν κοινές λέξεις με τις διευθύνσεις που έχει ακολουθήσει ο χρήστης μέχρι στιγμής.

Ο πράκτορας Syskill & Webert [Ack97, Paz97] εκπαιδεύει έναν Bayesian ταξινομητή ώστε να δημιουργήσει ένα πιθανοτικό προφίλ του χρήστη και βάσει αυτού να του προτείνει σελίδες από μια συλλογή προεπιλεγμένων σελίδων ή διευθύνσεων. Όλη η συλλογή ανήκει σε μια συγκεκριμένη θεματική ενότητα (π.χ. βιοϊατρική) και συντηρείται από έναν ή μια ομάδα ειδικών. Ο χρήστης αφού εξετάσει τις σελίδες εν

συνεχεία τις βαθμολογεί σε μια κλίμακα του τρία. Η βαθμολογία του χρήστη μαζί με το περιεχόμενο της σελίδας, χρησιμοποιούνται από τον πράκτορα για την εκμάθηση των ενδιαφερόντων του χρήστη. Το προφίλ του χρήστη μπορεί να χρησιμοποιηθεί με δύο τρόπους. Πρώτον, να προτείνει ο πράκτορας συνδέσμους στον κατάλογο οι οποίοι θα είναι ενδιαφέροντες στο χρήστη να τους εξερευνήσει και δεύτερον, μπορεί να χτίσει μια ερώτηση στην μηχανή αναζήτησης του Lycos για να βρει άλλες σχετικές σελίδες.

Τα κείμενα που έχει βαθμολογήσει ο χρήστης χωρίζονται σε δύο κατηγορίες, στα θετικά και στα αρνητικά παραδείγματα, και χρησιμοποιούνται για την εκμάθηση του ταξινομητή. Κάθε νέο κείμενο που πρόκειται να υποδειχτεί στον χρήστη, εισάγεται πρώτα στον ταξινομητή, ο οποίος και θα υπολογίσει την πιθανότητα το κείμενο να είναι σύμφωνο με τα ενδιαφέροντα του χρήστη.

4.5 Συμπεράσματα

Στο κεφάλαιο αυτό πραγματοποιήθηκε μια σύντομη παρουσίαση της τεχνολογίας των πρακτόρων λογισμικού δίνοντας έμφαση στις ιδιότητες και τα χαρακτηριστικά που διαθέτουν οι πράκτορες λογισμικού. Επίσης, κάναμε μια κατηγοριοποίηση και παρουσίαση των πεδίων εφαρμογής των πρακτόρων λογισμικού και εστιάσαμε στους ευφυείς πράκτορες που έχουν σαν στόχο το φιλτράρισμα και την συλλογή πληροφοριών από το διαδίκτυο.

Οι πράκτορες λογισμικού μπορούν να μειώσουν το φόρτο εργασίας της αναζήτησης και συλλογής πληροφοριών με την εκτέλεση εργασιών καθ' όλη την διάρκεια της ημέρας, χωρίς την άμεση συμμετοχή του χρήστη. Για παράδειγμα, οι πράκτορες λογισμικού αφού δημιουργήσουν μια αναπαράσταση των ενδιαφερόντων του χρήστη, εν συνεχεία μπορούν να επισκεφτούν τους κόμβους αναζήτησης πληροφοριών, να σχηματίσουν ερωτήσεις, να αντλήσουν την λίστα των αποτελεσμάτων, να επισκεφτούν αρκετές ιστοσελίδες και να παρουσιάσουν στο τέλος μόνο εκείνες που ταιριάζουν στα ενδιαφέροντα του χρήστη.

Στο επόμενο κεφάλαιο, θα παρουσιάσουμε το σύστημα Webnaut το οποίο βασίζεται στην τεχνολογία των πρακτόρων λογισμικού και μειώνει το φόρτο εργασίας ενός χρήστη του διαδικτύου σε θέματα συλλογής πληροφοριών.

Κεφάλαιο 5

ΤΟ ΣΥΣΤΗΜΑ WEBNAUT

5.1 Εισαγωγή

Η ανάπτυξη ενός συστήματος, που σκοπό θα έχει την εκμάθηση των ενδιαφερόντων ενός χρήστη και την αναζήτηση σχετικών κειμένων στο διαδίκτυο, είναι μια δύσκολη εργασία. Τα προβλήματα εντοπίζονται τόσο σε θέματα ανάπτυξης του λογισμικού όσο και σε θέματα αλληλεπίδρασης του χρήστη με το σύστημα. Στο διαδίκτυο υπάρχουν αρκετά πρωτόκολλα και μέθοδοι επικοινωνίας. Γι' αυτό το λόγο, η σχεδίαση του συστήματος θα πρέπει να επιτρέπει την καλή επικοινωνία των διαφόρων τμημάτων του λογισμικού, ώστε να δίνουν την εικόνα ενός *συμπαγούς* συστήματος. Αυτό έχει ιδιαίτερη αξία για τον χρήστη, ο οποίος δεν θέλει να ασχολείται με λεπτομέρειες αλλά να κάνει την εργασία του, αδιαφορώντας ποίο συγκεκριμένο κομμάτι του λογισμικού θα την φέρει εις πέρας.

Οι πληροφορίες στο διαδίκτυο είναι διασκορπισμένες σε διαφορετικούς κόμβους και σε διαφορετικές μορφές. Η όλη σχεδίαση του συστήματος θα πρέπει να επιτρέπει την εύκολη προσάρτηση νέων λειτουργιών με σκοπό την πρόσβαση του συστήματος σε πληροφορίες που βρίσκονται σε νέους δικτυακούς τόπους. Αυτό έχει δύο πλεονεκτήματα, πρώτον, ο μηχανισμός εκμάθησης παρέχει ποιοτικές πληροφορίες και δεύτερον, αυξάνει την εμπιστοσύνη του χρήστη προς το σύστημα επειδή αισθάνεται ότι ενημερώνεται συνεχώς με νέες πληροφορίες.

Όσον αφορά την αλληλεπίδραση με το χρήστη, πιστεύουμε ότι, ένα *ευφυές* σύστημα δεν θα πρέπει να αλλάζει τον τρόπο πρόσβασης των χρηστών στο διαδίκτυο, αλλά να τον υποστηρίζει συνεχώς καθ' όλη την διάρκεια της πλοήγησης του. Για αυτό το λόγο καλό θα είναι το σύστημα να συνεργάζεται με τα ήδη γνωστά προγράμματα φυλλομέτρησης.

Ακόμα, πιστεύουμε ότι οι *πρωτοβουλίες* θα πρέπει να ανήκουν στον χρήστη, τόσο στην παροχή πληροφοριών σχετικά με τα ενδιαφέροντα του, όσο και με την επίδειξη των υποδείξεων του συστήματος. Το διαδίκτυο παρέχει πολλές και ενδιαφέρουσες πληροφορίες και είναι αρκετά εύκολο για ένα χρήστη να ακολουθήσει ένα μονοπάτι συνδέσμων, το οποίο δεν ήταν στις αρχικές του προθέσεις. Μια τέτοια

συμπεριφορά κάνει αρκετά δύσκολη την εκμάθηση των ενδιαφερόντων ενός χρήστη γιατί υπάρχουν πληροφορίες που δεν σχετίζονται άμεσα με τα μακροχρόνια ενδιαφέροντα του.

Η μελέτη της σχετικής βιβλιογραφίας, καθώς και η δική μας γνώση και εμπειρία ως χρήστες του διαδικτύου μας βοήθησαν στο να καθορίσουμε τις προδιαγραφές του συστήματος, οι οποίοι συνοψίζονται παρακάτω :

- Το σύστημα θα πρέπει να υποστηρίζει τον χρήστη σε θέματα φιλτραρίσματος και αναζήτησης πληροφοριών στο διαδίκτυο. Θα πρέπει να εξειδικεύεται στα ενδιαφέροντα του χρήστη καθώς επίσης να εξερευνά και να υποδεικνύει νέους δικτυακούς τόπους που πιθανόν ενδιαφέρουν τον χρήστη.

- Το σύστημα δεν θα πρέπει να απευθύνεται στους χρήστες ενός συγκεκριμένου γνωστικού αντικειμένου αλλά θα πρέπει να είναι γενικό, δηλαδή να έχει εφαρμογή σε όλα τα γνωστικά πεδία και να απευθύνεται σε όλους τους χρήστες, ανεξαρτήτως εμπειρίας ή γνώσεως.

- Το σύστημα θα πρέπει να προσαρμόζεται στα ενδιαφέροντα του χρήστη καθώς αυτά μεταβάλλονται με τον χρόνο και την εμπειρία.

- Το σύστημα θα πρέπει να επιτρέπει στον χρήστη να καθορίσει νέες πηγές πληροφοριών.

- Ο χρήστης θα πρέπει να έχει τον πλήρη έλεγχο των λειτουργιών του συστήματος και τη δυνατότητα της απενεργοποίησης των διαφόρων υπηρεσιών του.

- Το σύστημα θα πρέπει να μην αλλάζει τον τρόπο πλοήγησης του χρήστη στο διαδίκτυο και να συνεργάζεται με όλους τους δημοφιλείς φυλλομετρητές, ακόμα και σε διαφορετικά λειτουργικά συστήματα.

- Η αρχιτεκτονική του συστήματος θα πρέπει να είναι ανοικτή ώστε να είναι αρκετά εύκολη υπόθεση η ενσωμάτωση νέων υπηρεσιών.

- Η διεπαφή αλληλοεπίδρασης του συστήματος θα πρέπει να είναι ελκυστική στους

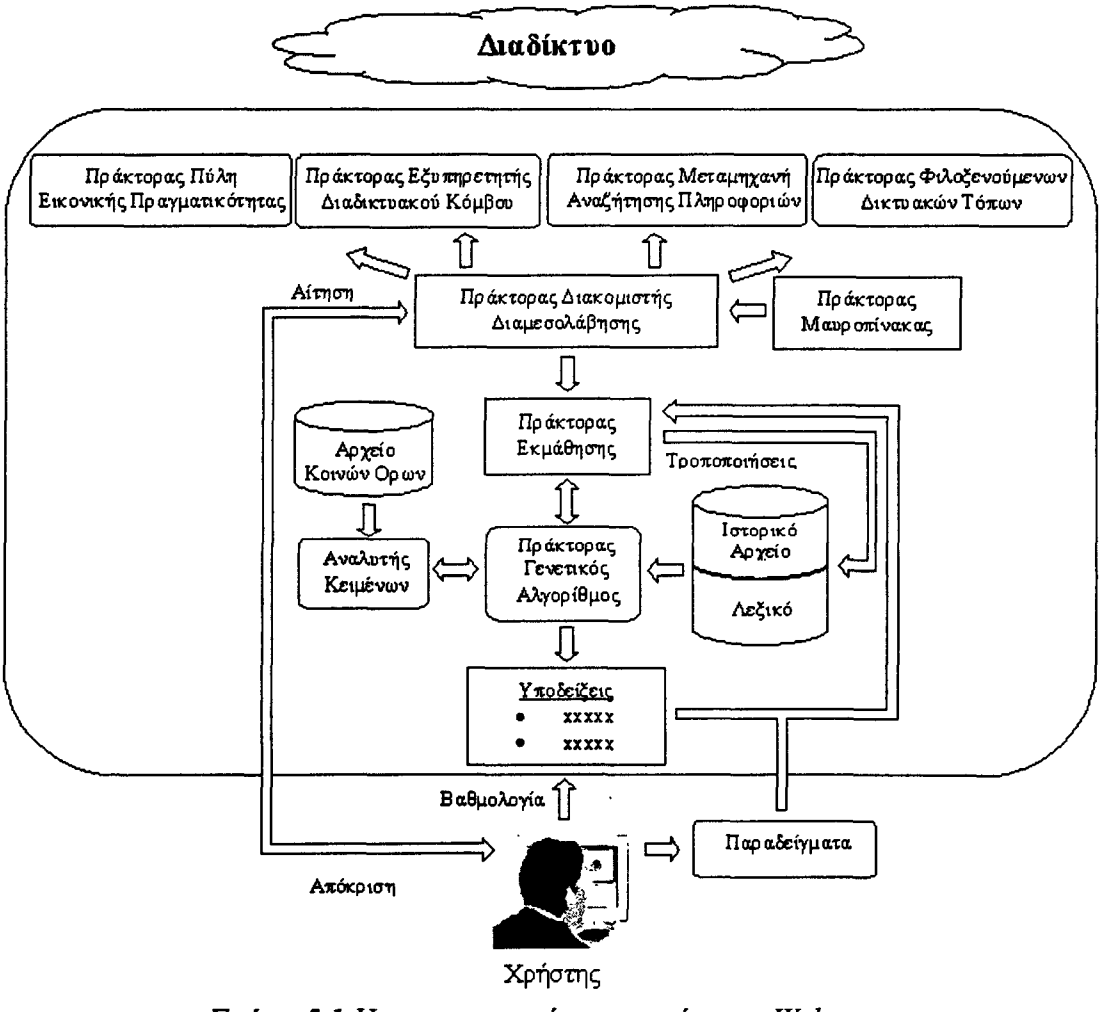
χρήστες και να προσφέρει επιπρόσθετες υπηρεσίες που δεν υπάρχουν στα προγράμματα πλοήγησης.

- Δεν πρέπει σε καμία περίπτωση να υπάρχει θέμα ασφάλειας και παραβίασης των προσωπικών δεδομένων του χρήστη.

Όλες οι παραπάνω απαιτήσεις και προδιαγραφές υλοποιούνται στο ευφυές σύστημα Webnaut.

5.2 Η Αρχιτεκτονική του Συστήματος Webnaut

Το σύστημα Webnaut αποτελείται από ένα σύνολο πρακτόρων λογισμικού οι οποίοι συνεργάζονται μεταξύ τους για να δημιουργήσουν ένα ομοιογενούς σύστημα.



Σχήμα 5.1 Η αρχιτεκτονική του συστήματος Webnaut.

Τα βασικά στοιχεία που συνθέτουν το λογισμικό καθώς και η αρχιτεκτονική του συστήματος εμφανίζονται στο σχήμα 5.1. Ο χρήστης αλληλεπιδρά με το σύστημα μέσω οποιουδήποτε προγράμματος πλοήγησης για το διαδικτύο, το οποίο υποστηρίζει πλαίσια (HTML Frames) [Γ12.a] και JAVASCRIPT [Γ10]. Μετά την διαμόρφωση των επιλογών του προγράμματος πλοήγησης ώστε να χρησιμοποιήσει σαν εξυπηρετητή διαμεσολάβησης τον πράκτορα διακομιστή διαμεσολάβησης - δες παρακάτω - είναι δυνατό στον χρήστη να πλοηγηθεί στο διαδίκτυο με την *έξυπνη* καθοδήγηση του λογισμικού μας. Οι πράκτορες που συνθέτουν το σύστημα *Webnaut*, εμφανίζονται παρακάτω :

- Ο Πράκτορας Διακομιστής Διαμεσολάβησης
- Ο Πράκτορας Φιλοξενούμενων Δικτυακών Τόπων
- Ο Πράκτορας Μαυροπίνακας
- Ο Πράκτορας Εξυπηρετητής Διαδικτυακού Κόμβου
- Ο Πράκτορας Πύλη Εικονικής Πραγματικότητας
- Ο Πράκτορας Μεταμηχανή Αναζήτησης Πληροφοριών
- Ο Πράκτορας Εκμάθησης
- Ο Πράκτορας Γενετικός Αλγόριθμος

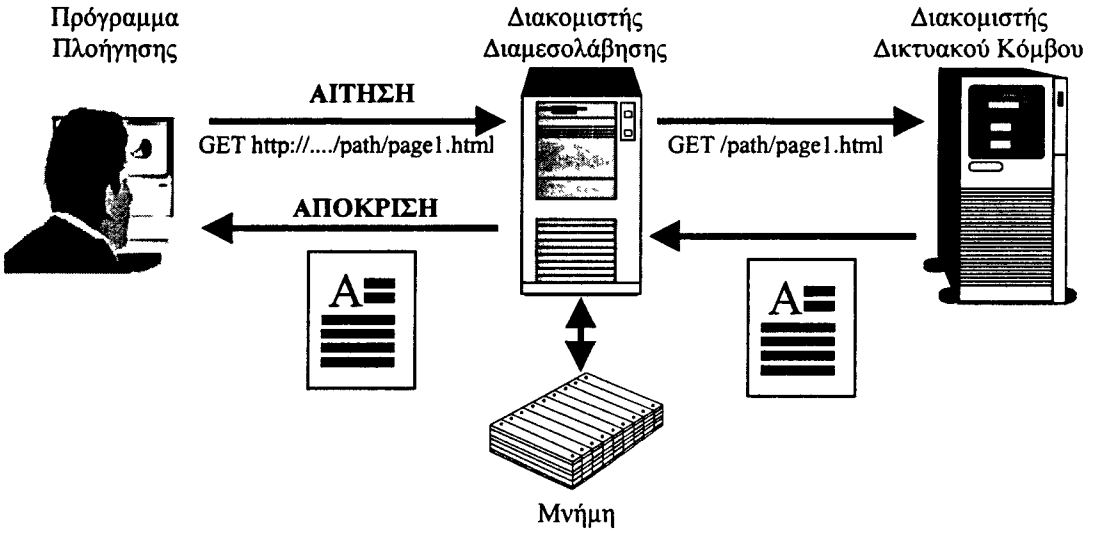
Στην συνέχεια του κεφαλαίου θα παρουσιάσουμε για τον κάθε πράκτορα την σχεδίαση, την δομή, τον τρόπο λειτουργίας του, τα υποσυστήματα του, καθώς και την διεπαφή – όπου αυτή υπάρχει – για την αλληλεπίδραση με το χρήστη.

5.3 Σχεδίαση του Πράκτορα Διακομιστή Διαμεσολάβησης

Ο Πράκτορας Διακομιστής Διαμεσολάβησης - ΠΔΔ, είναι ο συνδετικός κρίκος που ενώνει όλους τους πράκτορες του συστήματος. Γενικά, ένας διακομιστής διαμεσολάβησης είναι μια εφαρμογή ή οποία βρίσκεται ανάμεσα στο πρόγραμμα πλοήγησης του χρήστη και στον πραγματικό εξυπηρετητή στο διαδίκτυο. Ο διακομιστής διαμεσολάβησης εξετάζει αν μπορεί ο ίδιος να ικανοποιήσει όλες τις αιτήσεις του προγράμματος πλοήγησης και αν όχι τότε και μόνο τότε προωθεί την αίτηση στον πραγματικό εξυπηρετητή.

Τις περισσότερες φορές ένας διακομιστής διαμεσολάβησης μπορεί να χειριστεί

αρκετά πρωτόκολλα επικοινωνίας, π.χ. HTTP [Γ1], Gopher [Γ2], FTP [Γ3], WAIS [Γ5].



Σχήμα 5.2 *Η συναλλαγή του προγράμματος πλοήγησης με έναν εξυπηρετητή διαμεσολάβησης*

Το σχήμα 5.2, παρουσιάζει την συναλλαγή ενός προγράμματος πλοήγησης με τον εξυπηρετητή διαμεσολάβησης, για την ανάκτηση ενός αρχείου από έναν απομακρυσμένο εξυπηρετητή. Τα σημαντικότερα πλεονεκτήματα από την χρήση ενός εξυπηρετητή διαμεσολάβησης είναι τα παρακάτω:

Αύξηση Απόδοσης : Οι διακομιστές διαμεσολάβησης μπορούν να αυξήσουν την ταχύτητα εξυπηρέτησης των χρηστών στο διαδίκτυο λόγω του ότι αποθηκεύουν όλες τις αιτήσεις των χρηστών για ένα συγκεκριμένο χρονικό διάστημα. Για παράδειγμα ας θεωρήσουμε δύο χρήστες Α και Β οι οποίοι συνδέονται στο διαδίκτυο μέσω ενός εξυπηρετητή διαμεσολάβησης. Έστω ότι ο χρήστης Α απευθύνει μια αίτηση για μια συγκεκριμένη σελίδα την οποία θα ονομάσουμε Σελίδα 1. Ο διακομιστής διαμεσολάβησης ολοκληρώνει την αίτηση και εξυπηρετεί τον χρήστη Α.

Αν μετά από λίγη ώρα και ο χρήστης Β ζητήσει την ίδια σελίδα τότε ο διακομιστής διαμεσολάβησης θα εξυπηρετήσει τον χρήστη Β, όχι προωθώντας την αίτηση στο πραγματικό εξυπηρετητή, μια αρκετά χρονοβόρα διαδικασία, όπως έκανε με τον χρήστη Α, αλλά ανακαλώντας την Σελίδα 1 από την μνήμη του. Ένας διακομιστής διαμεσολάβησης μπορεί να εξυπηρετεί ταυτόχρονα πάρα πολλούς χρήστες και μάλιστα οι μεγάλοι παροχείς πρόσβασης στο διαδίκτυο διαθέτουν συστοιχίες από διακομιστές διαμεσολάβησης.

Φιλτράρισμα Αιτήσεων : Οι διακομιστές διαμεσολάβησης μπορούν να χρησιμοποιηθούν σαν φίλτρο για την απόρριψη ή όχι της εξυπηρέτησης μιας αίτησης σε επίπεδο πρωτοκόλλου ή για συγκεκριμένους κόμβους στο διαδίκτυο. Κατ' αυτόν τον τρόπο μπορούμε να έχουμε έλεγχο στους κόμβους που θα επισκέπτονται οι χρήστες ενός εξυπηρετητή διαμεσολάβησης.

Ασφάλεια : Από τα πρώτα χρόνια της ανάπτυξης του διαδικτύου όλες οι εταιρίες και οι οργανισμοί ενδιαφέρονται και προσέχουν ιδιαίτερα την ασφάλεια του εσωτερικού τους δικτύου. Ένας πολύ απλός και σίγουρος τρόπος προστασίας του εσωτερικού τους δικτύου ήταν να το αποσυνδέσουν εντελώς από το διαδίκτυο, με το σκεπτικό ότι αν κάποιος δεν μπορεί να συνδεθεί με τα μηχανήματα τους, είναι φυσικό ότι δεν θα μπορεί να τους προκαλέσει κάποια ζημιά.

Το αρνητικό όμως μιας τέτοιας λύσης είναι ότι οι χρήστες τους δεν μπορούν να έχουν πρόσβαση στους διακομιστές του διαδικτύου. Έτσι λοιπόν, οι διαχειριστές του εσωτερικού δικτύου, εγκαθιστούν σε ένα μηχάνημα της εταιρίας, το οποίο έχει πρόσβαση στο παγκόσμιο δίκτυο, έναν διακομιστή διαμεσολάβησης και όλες οι αιτήσεις των χρηστών για το διαδίκτυο εξυπηρετούνται μέσω αυτού του μηχανήματος.

Η ενσωμάτωση ενός διακομιστή διαμεσολάβησης στο σύστημα πολλών πρακτόρων Webnaut επιλύει πολλά ζητήματα, όπως για παράδειγμα :

- τον πλήρη έλεγχο των αιτήσεων του χρήστη κατά την διάρκεια της πλοήγησης του στο διαδίκτυο
- συμβατότητα με όλα τα γνωστά προγράμματα πλοήγησης
- την πλήρη αξιοποίηση των δυνατοτήτων των συγχρόνων προγραμμάτων πλοήγησης
- την δυνατότητα εξυπηρέτησης του χρήστη ανεξαρτήτου πλατφόρμας λειτουργικού συστήματος
- σταδιακή ενσωμάτωση νέων λειτουργιών

- δημιουργία ομοιογενούς συστήματος με την ανακατεύθυνση των αιτήσεων προς τις υπηρεσίες του συστήματος

Στην παρούσα έκδοση του συστήματος υποστηρίζουμε το πρωτόκολλο HTTP αφού είναι το πιο διαδεδομένο ανάμεσα στα πρωτόκολλα του διαδικτύου για τη εμφάνιση πληροφοριών σε μορφή κειμένου. Επίσης, θα πρέπει να τονίσουμε ότι για την αύξηση της ταχύτητας εξυπηρέτησης των αιτήσεων του πελάτη, ο ΠΔΔ δεν υλοποιεί την λειτουργία της τοπικής μνήμης. Εξάλλου δεν υπάρχει κανένα όφελος αφού όλα τα σύγχρονα προγράμματα πλοήγησης διαθέτουν την δική τους μνήμη, όπου σε αυτήν αποθηκεύουν αρχεία (σελίδες, εικόνες κ.λπ.) ώστε να αποφεύγουν τις συχνές συνδέσεις, είτε πρόκειται για ένα απομακρυσμένο δικτυακό κόμβο, όταν λειτουργούν αυτόνομα στο διαδίκτυο, είτε μέσω εξυπηρετητή διαμεσολάβησης.

Ένα σημαντικό πλεονέκτημα του ΠΔΔ είναι η δυνατότητα χρήσης ενός επιπλέον εξυπηρετητή διαμεσολάβησης. Πρακτικά, μπορούμε να ρυθμίσουμε το σύστημα Webnaut να απευθύνει όλες τις αιτήσεις του πελάτη σε άλλον εξυπηρετητή διαμεσολάβησης, δηλαδή να δημιουργήσουμε μια αλυσίδα σύνδεσης, όπου το πρόγραμμα πλοήγησης συνδέεται με το ΠΔΔ, και αυτός με την σειρά του συνδέεται και απευθύνει όλες τις αιτήσεις στον κεντρικό διακομιστή διαμεσολάβησης της εταιρίας ή του οργανισμού.

5.3.1 Λειτουργικότητα του Πράκτορα Διακομιστή Διαμεσολάβησης

Η κύρια λειτουργία του ΠΔΔ. όπως και κάθε διακομιστή διαμεσολάβησης, είναι να αναλύει και να εξυπηρετεί τις αιτήσεις του προγράμματος πλοήγησης του χρήστη για το διαδίκτυο. Στο σχήμα 5.3 εμφανίζεται το διάγραμμα δραστηριοτήτων του ΠΔΔ μετά την επίτευξη σύνδεσης ενός πελάτη με το διακομιστή. Οι ενέργειες του ΠΔΔ με την σειρά με την οποία εκτελούνται, είναι οι εξής :

Ανάλυση Αίτησης

Αρχικά ελέγχεται η αίτηση του πελάτη όσον αφορά την εγκυρότητα της ως προς το πρωτόκολλο HTTP. Στην συνέχεια, ο πελάτης, είτε ανακατευθύνεται σε μία σελίδα που περιγράφει το σφάλμα είτε στο επόμενο βήμα.

Ανάλυση Διακομιστή Διεύθυνσης

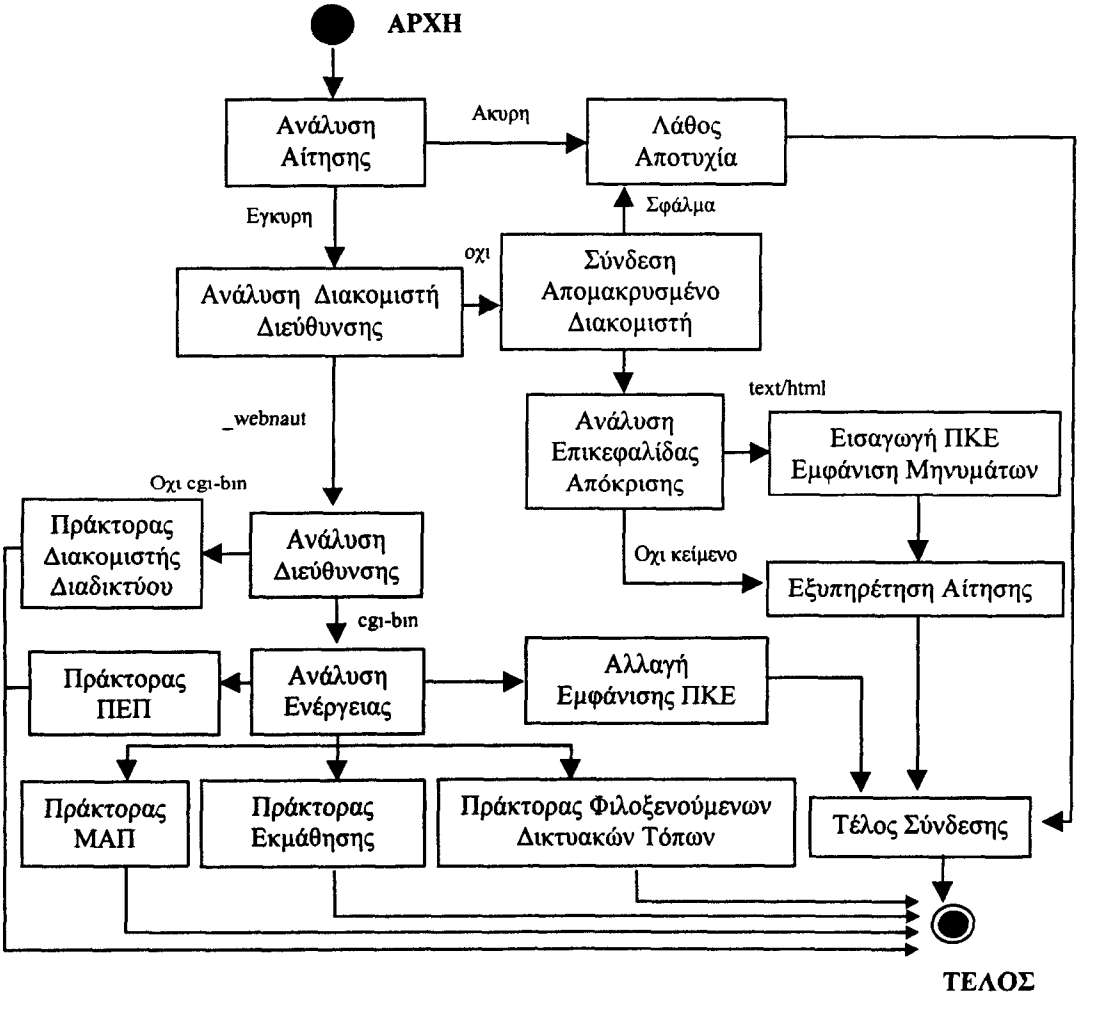
Εν συνεχεία θα γίνει έλεγχος της διεύθυνσης της αίτησης και πιο συγκεκριμένα, αφού όλοι οι πόροι στο διαδίκτυο έχουν την παρακάτω γενική μορφή :

Πρωτόκολλο://Υπολογιστής[:πόρτα]/Θέση

με απλή ανάλυση της διεύθυνσης της αίτησης μπορούμε να εντοπίσουμε το διακομιστή εξυπηρέτησης. Εάν, για παράδειγμα, μία αίτηση απευθύνεται στο σύστημα Webnaut τότε στην θέση του υπολογιστή θα εμφανίζεται η λέξη _webnaut και ολόκληρη η αίτηση θα μπορούσε να έχει την εξής μορφή :

GET http://_webnaut/action.html HTTP/1.0

Αν ο διακομιστής εξυπηρέτησης είναι ο _webnaut τότε θα συνεχίσει με την ανάλυση της διεύθυνσης ή διαφορετικά θα δημιουργήσει σύνδεση με τον απομακρυσμένο διακομιστή.



Σχήμα 5.3 Το διάγραμμα δραστηριοτήτων του ΠΔΔ

Ανάλυση Διεύθυνσης

Στα πλαίσια αυτής της δραστηριότητας ο ΠΔΔ θα ελέγξει την αρχική θέση του πόρου της αίτησης του πελάτη. Πιο συγκεκριμένα, αν ο πόρος βρίσκεται στο φάκελο cgi-bin τότε η αίτηση απευθύνεται σε κάποιον από τους πράκτορες του συστήματος. Για παράδειγμα, μια αίτηση προς τον Πράκτορα Εκμάθησης, θα έχει την μορφή

GET http://_webnaut/cgi-bin/LearningServer/action.html HTTP/1.0

Ο φάκελος cgi-bin είναι ένας εικονικός φάκελος ο οποίος χρησιμοποιείται εσωτερικά από το σύστημα αποκλειστικά και μόνο για να διαχωρίζει τις αιτήσεις που απευθύνονται προς τον πράκτορα Πράκτορα Διακομιστή Διαδικτύου - δες παρακάτω - από τους άλλους πράκτορες του συστήματος. Ο μόνος πόρος που αρχίζει με το συνθετικό cgi-bin και τον εξυπηρετεί πλήρως ο ΠΔΔ, είναι η αίτηση για την αλλαγή εμφάνισης της πινακίδας κεντρικού ελέγχου - δες στην συνέχεια.

Σύνδεση Απομακρυσμένου Διακομιστή

Αν ο υπολογιστής εξυπηρέτησης δεν είναι ο _webnaut, τότε ο ΠΔΔ δημιουργεί μια σύνδεση με αυτόν, με σκοπό να εξυπηρετήσει την αίτηση του πελάτη. Αν η σύνδεση δεν είναι επιτυχής τότε ενημερώνεται ο πελάτης με συγκεκριμένο μήνυμα λάθους, διαφορετικά συνεχίζει με την επόμενη ενέργεια.

Ανάλυση Επικεφαλίδας Απόκρισης

Ο ΠΔΔ διαβάζει και αναλύει όλη την επικεφαλίδα που αποστέλλει ο απομακρυσμένος υπολογιστής. Στην φάση αυτή προσπαθεί να ανακαλύψει το είδος της απόκρισης του υπολογιστή, δηλαδή αν θα αποστείλει δεδομένα ή μόνο την επικεφαλίδα (π.χ. ανακατεύθυνση κ.λπ.) ή συνδυασμό τους. Στην περίπτωση αποστολής δεδομένων η τιμή της παραμέτρου, τύπος περιεχομένου [Γ11], που περιγράφεται στην επικεφαλίδα χρησιμοποιείται για την εισαγωγή, η μη, της Πινακίδας του Κεντρικού Ελέγχου (ΠΚΕ). Αν η αίτηση αναφέρεται σε ένα αρχείο πολυμέσων, όπως για παράδειγμα μια εικόνα, ήχο κ.λπ., ή για μια εφαρμογή τύπου JAVA [Γ9] για το πρόγραμμα πλοήγησης (applet), τότε σε αυτή την περίπτωση η αίτηση εξυπηρετείται χωρίς παραπέρα επεξεργασία, δηλαδή ο ΠΔΔ παραλαμβάνει το αρχείο και εν συνεχεία το προωθεί στο πρόγραμμα πλοήγησης.

Εισαγωγή ΠΚΕ/Εμφάνιση Μηνυμάτων

Αν η αίτηση αναφέρεται σε μια σελίδα τύπου περιεχομένου κειμένου, δηλαδή περιγράφεται στην επικεφαλίδα σαν text/html, τότε ο ΠΔΔ αφού την ανακτήσει από τον απομακρυσμένο εξυπηρετητή, την αναλύει και εισάγει κώδικα σε γλώσσα HTML [Γ12] και JAVASCRIPT [Γ10] μετά το διακριτικό <BODY> της HTML σελίδας. Η συγκεκριμένη λειτουργία είναι υπεύθυνη για δύο ενέργειες : α) Την εμφάνιση της ΠΚΕ στην κορυφή της σελίδας που έχει ζητήσει ο χρήστης, και β) Την εμφάνιση σε νέο αυτοαναδυόμενο παράθυρο, αν φυσικά υπάρχουν, διαφόρων μηνυμάτων από το σύστημα προς τον χρήστη.

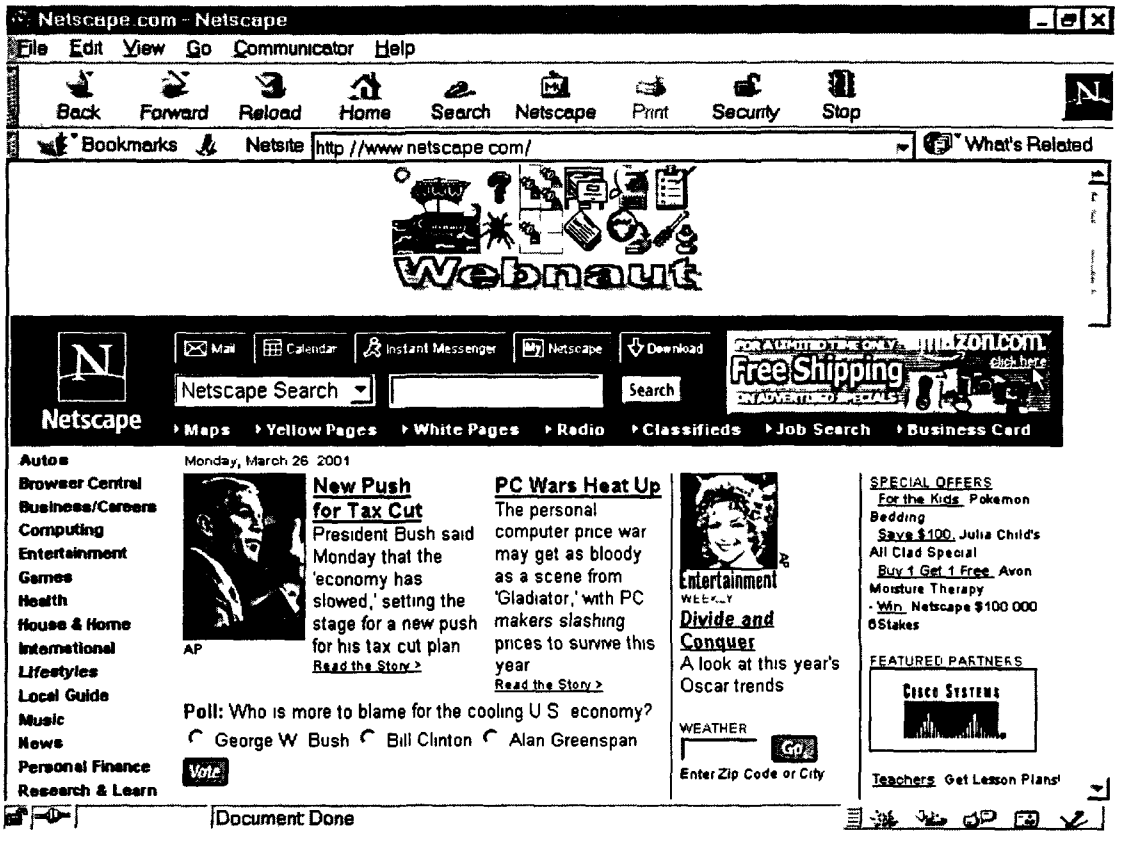
Θα πρέπει να τονίσουμε ότι η σύνδεση του ΠΔΔ με τον πελάτη, όπως φαίνεται και στο διάγραμμα δραστηριοτήτων, τερματίζει μόνο σε μερικές περιπτώσεις - δες σύνδεση με απομακρυσμένο εξυπηρετητή, αλλαγή εμφάνισης ΚΠΕ - και αυτό συμβαίνει γιατί, όπως θα δούμε παρακάτω, ο έλεγχος της αίτησης περνάει αυτούσιος σε ένα από τους πράκτορες που μπορεί να την ολοκληρώσει και κατά συνέπεια, αυτός ο πράκτορας είναι υπεύθυνος για τον τερματισμό της σύνδεσης.

5.3.2 Η Πινακίδα Κεντρικού Ελέγχου

Η πινακίδα κεντρικού ελέγχου είναι το κανάλι επικοινωνίας μεταξύ του χρήστη και του συστήματος αφού, όλες οι λειτουργίες του συστήματος απεικονίζονται με εικονίδια που μπορεί να επιλέξει ο χρήστης. Με την επιλογή κάποιου εικονιδίου θα ενεργοποιηθεί και η αντίστοιχη λειτουργία. Έτσι λοιπόν, ο χρήστης καθ' όλη την διάρκεια της πλοήγησης του στο διαδίκτυο βρίσκεται συνεχώς υπό την καθοδήγηση του συστήματος μας. Το σχήμα 5.4 παρουσιάζει την πινακίδα κεντρικού ελέγχου όπως εμφανίζεται κατά την επίσκεψη του χρήστη στο δικτυακό τόπο της εταιρίας Netscape (www.netscape.com)

Η ΠΚΕ εμφανίζεται στο πάνω μέρος της σελίδας που επισκοπεί ο χρήστης και πάντα υπάρχει η δυνατότητα της απόκρυψης της. Πιστεύουμε ότι η εμφάνιση των λειτουργιών του συστήματος μας, με την μορφή εικονιδίων μέσα στην σελίδα που βλέπει ο χρήστης και όχι σε κάποιο άλλο ανεξάρτητο παράθυρο, είναι προς όφελος του χρήστη, αφού χωρίς να χάνει το ενδιαφέρον για τη σελίδα που βρίσκεται μπροστά του, μπορεί την ίδια στιγμή να απευθύνει στον πράκτορα μια εντολή, όπως για παράδειγμα να κατεβάσει ένα ολόκληρο δικτυακό τόπο ή να του στείλει ένα μήνυμα,

όπως για παράδειγμα "μου αρέσει αυτή η σελίδα".



Σχήμα 5.4 Η πινακίδα κεντρικού ελέγχου επιτρέπει τον έλεγχο των λειτουργιών του συστήματος μέσω εικόνων.

Όλες οι εικόνες της ΠΚΕ καθώς οι λειτουργίες που ενεργοποιούν εμφανίζονται παρακάτω :

1. ..... Εμφάνιση της βοήθειας του συστήματος.

Μέσω της συγκεκριμένης επιλογής εμφανίζεται στο πρόγραμμα πλοήγησης του χρήστη, οδηγίες τόσο για την εγκατάσταση όσο και για την λειτουργία του συστήματος Webnaut. Επίσης παρέχονται πληροφορίες για την λειτουργία της εκμάθησης καθώς και για την μεθοδολογία ενημέρωσης του συστήματος για τα ενδιαφέροντα του χρήστη.

2. ..... Εμφάνιση της διεπαφής του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών.

Η συγκεκριμένη επιλογή δίνει στο χρήστη την δυνατότητα να εμφανίσει την διεπαφή του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών με σκοπό να απευθύνει ερωτήσεις προς τους δικτυακούς τόπους αναζήτησης πληροφοριών.

3. ..... Εμφάνιση της ΠΚΕ σε όλα τα πλαίσια [Γ12.α] της σελίδας.

Πολλές σελίδες στο διαδίκτυο χρησιμοποιούν τα πλαίσια με σκοπό να δημιουργήσουν διάφορες ενδιαφέρουσες παρουσιάσεις. Η συγκεκριμένη επιλογή δίνει στο χρήστη την δυνατότητα να εμφανίσει την ΠΚΕ σε όλα τα πλαίσια της τρέχουσας σελίδας με σκοπό είτε να αποθηκεύσει μια παρουσίαση στο τοπικό δίσκο είτε να υποδείξει μια συγκεκριμένη σελίδα της παρουσίασης, ως παράδειγμα των ενδιαφερόντων του.

4. ..... Εμφάνιση της ΠΚΕ στο πρώτο πλαίσιο της σελίδας.

Η συγκεκριμένη επιλογή δίνει στο χρήστη την δυνατότητα να εμφανίσει την ΠΚΕ μόνο στο πρώτο πλαίσιο της σελίδας.

5. ..... Εμφάνιση των περιεχομένων των δικτυακών τόπων που έχουν αποθηκευτεί στο τοπικό δίσκο με την βοήθεια του Πράκτορα Φιλοξενούμενων Δικτυακών Τόπων.

Ο χρήστης κατά την διάρκεια της πλοήγησης του στο διαδίκτυο, έχει την δυνατότητα να ενεργοποιήσει την λειτουργία της αποθήκευσης ορισμένων σελίδων ή και ολόκληρων κόμβων στο τοπικό δίσκο ενώ παράλληλα συνεχίζει την πλοήγηση του σε άλλους δικτυακούς κόμβους. Με την συγκεκριμένη επιλογή ο χρήστης έχει την δυνατότητα να πλοηγηθεί σε έναν από τους τοπικά αποθηκευμένους δικτυακούς τόπους.

6. ..... Εμφάνιση των υποδείξεων του συστήματος

Η συγκεκριμένη επιλογή εμφανίζει μία λίστα με όλες τις υποδείξεις του συστήματος οι οποίες θεωρούνται σαν σχετικές με τα ενδιαφέροντα του χρήστη, και έχουν επιλεγεί με την βοήθεια του Πράκτορα Εκμάθησης. Ο χρήστης αφού επιθεωρήσει την κάθε υπόδειξη του συστήματος, στην συνέχεια δίνει την βαθμολογία του η οποία αποδίδει το βαθμό ικανοποίησης του, για την συγκεκριμένη υπόδειξη.

7. ..... Εμφάνιση της διεπαφής του Πράκτορα Φιλοξενούμενων Δικτυακών Τόπων.

Η συγκεκριμένη επιλογή εμφανίζει την διεπαφή του Πράκτορα Φιλοξενούμενων Δικτυακών Τόπων με σκοπό να ορίσει ο χρήστης τις παραμέτρους για την αποθήκευση ενός δικτυακού τόπου στο τοπικό δίσκο.

8. ..... Αμεση αποθήκευση του τρέχοντος δικτυακού τόπου.

Η συγκεκριμένη επιλογή είναι μια συντόμευση της ανωτέρω λειτουργίας, όπου ο χρήστης χωρίς να ορίσει παραμέτρους, αποθηκεύει άμεσα τον τρέχοντα δικτυακό τόπο στο τοπικό δίσκο.

9. ..... Καταγραφή της τρέχουσας σελίδας ως ένα παράδειγμα των ενδιαφερόντων του χρήστη.

Με την συγκεκριμένη επιλογή ο χρήστης έχει την δυνατότητα να ενημερώσει το σύστημα ότι τα περιεχόμενα της τρέχουσας σελίδας είναι σύμφωνα με τα ενδιαφέροντα του.

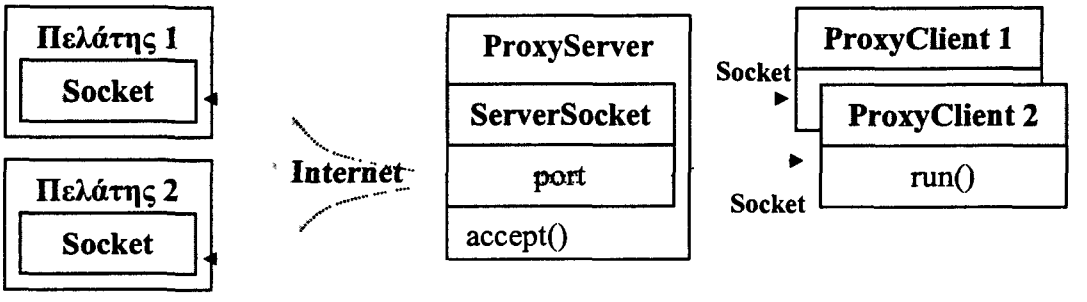
10. ..... Εμφάνιση της διεπαφής για την ρύθμιση των παραμέτρων λειτουργίας του Πράκτορα Διακομιστή Διαμεσολάβησης καθώς και του Πράκτορα Εκμάθησης

Με την συγκεκριμένη επιλογή ο χρήστης έχει την δυνατότητα να αλλάξει τις παραμέτρους λειτουργίας του Πράκτορα Διακομιστή Διαμεσολάβησης καθώς και του Πράκτορα Εκμάθησης.

5.3.3 Λεπτομερής Σχεδιασμός του Πράκτορα Διακομιστή Διαμεσολάβησης

Ο ΠΔΔ αποτελείται από δύο κλάσεις : α) την κλάση *ProxyServer* [Β1] και β) την κλάση *ProxyClient* [Β2]. Η κλάση *ProxyServer* υλοποιεί την βασική δομή του διακομιστή σε δύο στάδια. Πρώτον, με την δημιουργία ενός *ServerSocket* [Α1] αντικειμένου, το οποίο καταλαμβάνει μια δικτυακή πόρτα για την εξυπηρέτηση των πελατών, και δεύτερον, χρησιμοποιώντας την μέθοδο *accept*, της κλάσης *ServerSocket*, για την αναμονή των επερχόμενων συνδέσεων.

Η μέθοδος *accept* επιστρέφει ένα στιγμιότυπο της κλάσης *Socket* [A1], η οποία αποτελεί και την σύνδεση με τον πελάτη. Το στιγμιότυπο αυτό χρησιμοποιείται σαν όρισμα για την δημιουργία ενός αντικειμένου της κλάσης *ProxyClient,* που εκτελείται σαν ανεξάρτητη διεργασία - μέσω της υλοποίησης της διεπαφής *Runnable* [A2] - για την εξυπηρέτηση της αίτησης του πελάτη. Το σχήμα 5.5 παρουσιάζει το πλαίσιο λειτουργίας του ΠΔΔ.



Σχήμα 5.5 *Το πλαίσιο λειτουργίας του ΠΔΔ*

Σε αυτό το σημείο θα πρέπει να σημειώσουμε ότι η ενημέρωση και αρχικοποίηση των παραμέτρων του διακομιστή γίνεται με την κλήση της μεθόδου *ReadProxyProperties.* Στο σώμα της εν λόγου μεθόδου δημιουργείται ένα αντικείμενο της βοηθητικής κλάσης *TagBuilder* [B23], το οποίο παίρνει σαν όρισμα το μονοπάτι ενός αρχείου κειμένου, το οποίο περιέχει όλες τις παραμέτρους του πράκτορα σε μία συγκεκριμένη μορφοποίηση. Η δημιουργία αρχείων παραμέτρων κρίθηκε απαραίτητη από τα πρώτα βήματα ανάπτυξης του λογισμικού, ώστε να μπορούμε να έχουμε έλεγχο στις παραμέτρους ενός πράκτορα και εν συνεχεία να μπορούμε να κάνουμε γρήγορες αλλαγές σ' αυτές χωρίς να επεμβαίνουμε στον κώδικα του προγράμματος.

Σε όλο το πρόγραμμα χρησιμοποιούνται δύο είδη αρχείων παραμέτρων τα οποία διαφοροποιούνται ως προς την μορφοποίηση τους. Πιο συγκεκριμένα, οι παράμετροι σε αυτά τα αρχεία θα έχουν μία από τις παρακάτω δύο μορφές :

α) <=s=> παράμετρος <==> τιμή <=e=> είτε

β) <=s=> παράμετρος <==> τιμή <==> τύπος <=e=>

Το πρώτο είδος μορφοποίησης χρησιμοποιείται από όλους τους πράκτορες για την περιγραφή ζευγαριών παραμέτρων - τιμών, π.χ. ένα τέτοιο ζευγάρι είναι το ακόλουθο

<=s=> PROXYPORT <==> 8080 <=e=>

όπου περιγράφουμε την δικτυακή πόρτα που θα καταλάβει ο *ProxyServer* και θα δέχεται σ' αυτή τις αιτήσεις των πελατών. Το δεύτερο είδος αρχείων παραμέτρων χρησιμοποιείται μόνο από τον Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών - δες παρακάτω - για την περιγραφή της μεθοδολογίας της ανάλυσης των αποτελεσμάτων από τις μηχανές αναζήτησης.

Και στις δύο περιπτώσεις χρησιμοποιούμε αντικείμενα της κλάση *TagBuilder* για να διαβάσουμε αρχικά τα αρχεία παραμέτρων και εν συνεχεία, με την χρήση των μεθόδων *getHashTags* και *getVectTags,* ανάλογα με το είδος των αρχείων, να πάρουμε αντίστοιχα, είτε όλα τα ζευγάρια παραμέτρων - τιμών σε ένα *Hashtable* [A3] είτε όλες τις τριάδες, παράμετρος - τιμή - τύπος, σε ένα *Vector* [A3] όπου η κάθε θέση του περιλαμβάνει ένα στιγμιότυπο της κλάσης *TagText* [B24]. Στην δεύτερη περίπτωση η πρόσβαση στην επιθυμητή πληροφορία γίνεται μέσω των αντίστοιχων μεθόδων, *getTag, getCode, getType,* της κλάσης *TagText.*

Επιπλέον, θα πρέπει να αναφέρουμε ότι ο κάθε πράκτορας του συστήματος χρησιμοποιεί το δικό του αρχείο παραμέτρων, το οποίο το διαβάζει μόνο μια φορά και τα ζευγάρια παράμετρος και τιμή - όπως αναφέραμε παραπάνω - αποθηκεύονται σε ένα *Hashtable* για γρήγορη πρόσβαση. Ακόμα, ο *Hashtable* είναι στατικός για να μοιράζονται όλα τα στιγμιότυπα της κλάσης πράκτορας τις ίδιες παραμέτρους.

Έτσι, λοιπόν το διάβασμα των παραμέτρων του κάθε πράκτορα ενεργοποιείται από το πρώτο στιγμιότυπο της κάθε κλάσης, και μάλιστα με το κάλεσμα της μεθόδου *initialize* η οποία και επιστρέφει μια τιμή αληθή η ψευδή ανάλογα με το αν έγινε έγκυρη αρχικοποίηση των παραμέτρων. Αυτή η ειδική ενεργοποίηση και αρχικοποίηση των πρακτόρων γίνεται στο σώμα της μεθόδου *StartModules* της κλάσης *ProxyServer.*

Αφού έχει επιτευχθεί η σύνδεση του πελάτη με το ΠΔΔ, στην συνέχεια μπορούμε να αποκτήσουμε τον έλεγχο της ροής των δεδομένων εισόδου και εξόδου. Οι κλάσεις *HttpResponse* [B5] και *HttpRequest* [B6] χρησιμοποιούνται για να μεταφέρουν τα δεδομένα ανάμεσα στον πελάτη και το διακομιστή. Και οι δύο κλάσεις δέχονται σαν παράμετρο ένα στιγμιότυπο της κλάσης *Socket,* και εν συνεχεία κάνουν χρήση των μεθόδων *getInputStream*[A1] και *getOutputStream*[A1] της κλάσης *Socket,* για να αποκτήσουν πρόσβαση στα κανάλια των δεδομένων εισόδου και εξόδου αντίστοιχα.

Η κλάση *HttpRequest* υλοποιεί μεθόδους για να διαβάσει και να

αποκωδικοποιήσει την αίτηση του χρήστη, είτε πρόκειται για μια αίτηση GET [Γ1] είτε για μια αίτηση POST [Γ1]. Η αποκωδικοποίηση των δεδομένων εισόδου επιτυγχάνεται με την βοήθεια της βοηθητικής κλάσης *URLDecoder* [B22], η οποία μετατρέπει, μέσω της μεθόδου *decode,* τα αλφαριθμητικά τα οποία είχαν την μορφοποίηση *x-www-form-urlEncoded* σε κανονικά κείμενα. Η μέθοδος *CGIProcess* της κλάσης *HttpRequest* ελέγχει και αποκωδικοποιεί την αίτηση ανάλογα με το είδος της.

Η εν λόγω μέθοδος επιστρέφει μια τιμή *true* ή *false* με σκοπό να παρέχει μια γρήγορη ενημέρωση σχετικά με την αίτηση του πελάτη, δηλαδή αν υπάρχει κάποιο σφάλμα ή όχι. Μια τιμή *true* μας δίνει το δικαίωμα να καλέσουμε εν συνεχεία μία από τις μεθόδους *getHttpMethod*, *getHttpURL*, *getHttpVersion* για να ενημερωθούμε σχετικά με την μέθοδο, την διεύθυνση και την έκδοση της αίτησης του πελάτη σύμφωνα πάντα με το HTTP πρωτόκολλο.

Αντίθετα, η κλάση *HttpResponse* είναι υπεύθυνη για το κανάλι εξόδου, δηλαδή των δεδομένων που θα αποσταλούν στο χρήστη μετά από μία αίτηση του. Έτσι λοιπόν υλοποιεί μεθόδους για την αποστολή τόσο των ακατέργαστων bytes, όσο και σελίδων μορφοποιημένων σε γλώσσα HTML. Επίσης η κλάση *HttpResponse*, υλοποιεί μεθόδους, όπως η *connectToServer*, για την απευθείας σύνδεση του πελάτη με έναν απομακρυσμένο διακομιστή. Ακόμα υλοποιεί μεθόδους όπως η *HttpError* και η *Redirect,* για την ενημέρωση του χρήστη μετά από ένα σφάλμα σύνδεσης, όσο και για την ανακατεύθυνση του σε νέα τοποθεσία του διαδικτύου. Η υλοποίηση του ΠΔΔ σαν πολύ-διεργασιακό (multithreaded) εξυπηρετητή μας επιτρέπει να εξυπηρετούμε ταυτόχρονα τις αιτήσεις του προγράμματος πλοήγησης.

5.4 Σχεδίαση του Πράκτορα Φιλοξενούμενων Δικτυακών Τόπων

Ο Πράκτορας Φιλοξενούμενων Δικτυακών Τόπων - ΠΦΔΤ, είναι ένας άλλος πράκτορας ο οποίος δεν χρησιμοποιεί τεχνικές Μηχανικής Εκμάθησης αλλά σκοπό έχει την διευκόλυνση του χρήστη σε μία σειρά από χρονοβόρα θέματα κατά την διάρκεια της πλοήγησης του στο διαδίκτυο. Για παράδειγμα, ένας ερευνητής όταν βρεθεί μπροστά σε μία ενδιαφέρουσα παρουσίαση ή σε ένα πολυσέλιδο άρθρο, σε μορφή html, το οποίο θέλει να αποθηκεύσει στον τοπικό του δίσκο για να το μελετήσει κάποια στιγμή αργότερα, δεν έχει άλλη επιλογή από το να κατεβάσει μία προς μία όλες τις σελίδες του άρθρου και στην συνέχεια να επέμβει στον κώδικα της

κάθε μίας σελίδας ώστε οι σύνδεσμοι να δείχνουν στα τοπικά αρχεία.

Ο ΠΦΔΤ μπορεί να βοηθήσει τον χρήστη σε τέτοιες περιπτώσεις και να κατεβάσει όλο το άρθρο ή ακόμα και ένα ολόκληρο δικτυακό τόπο και να το αποθηκεύσει στον Πράκτορα Διακομιστή Διαδικτυακού Κόμβου - δες παρακάτω - με σκοπό να τον εξερευνήσει κάποια στιγμή αργότερα, χωρίς να απαιτείται άμεση σύνδεση στο διαδίκτυο, ενώ ο χρήστης καθ' όλη την διάρκεια της όλης εργασίας συνεχίζει το ταξίδι του στον κυβερνοχώρο.

5.4.1 Λεπτομερής Σχεδιασμός του Πράκτορα Φιλοξενούμενων Δικτυακών Τόπων

Ο ΠΦΔΤ αποτελείται από τρεις κλάσεις : α) την κλάση *SiteMirror* [B12] β) την κλάση *SiteMirrorClient* [B13], γ) την κλάση *SiteMirrorRobot* [14]. Η κλάση *SiteMirror* έχει όλες τις προδιαγραφές να λειτουργήσει σαν αυτόνομος διακομιστής εξυπηρέτησης αιτήσεων αλλά, στα πλαίσια του συστήματος Webnaut, δέχεται τις αιτήσεις του ΠΔΔ μέσω της μεθόδου *HandleRequest* και εν συνεχεία δημιουργεί στιγμιότυπα της κλάσης *SiteMirrorClient,* τα οποία λειτουργούν σαν ανεξάρτητες διεργασίες αφού υλοποιούν την διεπαφή *Runnable.*

Όλες οι παράμετροι λειτουργίας του ΠΦΔΤ αρχικοποιούνται μέσω της μεθόδου *ReadServerProperties* και εν συνεχεία αποθηκεύονται σε ένα στατικό *Hashtable.* Το αποτέλεσμα της μεθόδου *ReadServerProperties,* το οποίο δείχνει την έγκυρη ή μη ενεργοποίηση του ΠΦΔΤ, επιστρέφει η μέθοδος *initialize,* την οποία έχει καλέσει η μέθοδος *StartModules* της κλάσης *ProxyServer.*

Η ανεξάρτητη διεργασία *SiteMirrorClient* κάνει δύο ενέργειες α) ελέγχει την αίτηση του χρήστη αν είναι έγκυρη ή μη. Στην περίπτωση που υπάρχει κάποιο σφάλμα στην αίτηση του πελάτη ή υπάρχει έλλειψη στοιχείων τότε ενημερώνεται ο πελάτης με το ανάλογο μήνυμα και φυσικά τερματίζει την σύνδεση. β) Στην περίπτωση που δεν υπάρχει κάποιο σφάλμα, τότε δημιουργεί ένα αντικείμενο της κλάσης *SiteMirrorRobot,* το οποίο λειτουργεί σαν ανεξάρτητη διεργασία μέσω της διεπαφής *Runnable* και ταυτόχρονα πριν κλείσει την σύνδεση ανακατευθύνει τον πελάτη σε μία σελίδα του Πράκτορα Διακομιστή Διαδικτυακού Κόμβου - δες παρακάτω - όπου εμφανίζεται ένα μήνυμα που δηλώνει ότι η αίτηση του έγινε κατανοητή και έχει αρχίσει η διαδικασία για την ολοκλήρωση της.

Η κλάση *SiteMirrorRobot* περιέχει όλη την λογική που απαιτείται για να αποθηκεύσει τις σελίδες στον τοπικό δίσκο σύμφωνα πάντα με τις προδιαγραφές που

έχει θέσει ο χρήστης.

5.4.2 Λειτουργικότητα του Πράκτορα Φιλοξενούμενων Δικτυακών Τόπων

Στο σχήμα 5.6 εμφανίζεται το διάγραμμα δραστηριοτήτων του ΠΦΔΤ. Οι ενέργειες του πράκτορα με την σειρά με την οποία εκτελούνται, είναι οι εξής :

Ελεγχος των Παραμέτρων Αίτησης

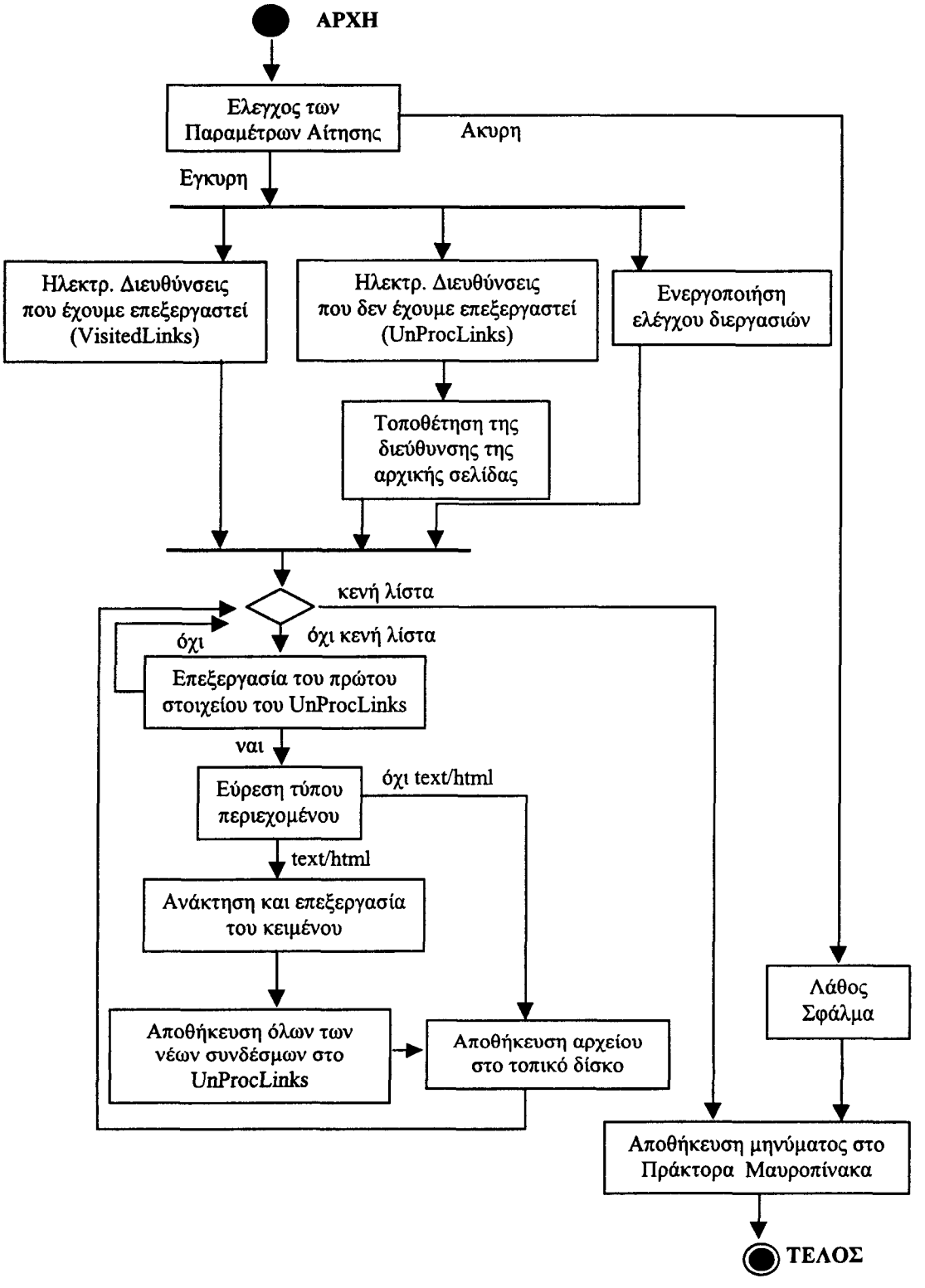
Σαν πρώτη ενέργεια ελέγχονται όλες οι παράμετροι της αίτησης, π.χ. αν τα πεδία της αίτησης έχουν αποδεκτές τιμές ή όχι. Σε περίπτωση λάθους δημιουργείται ένα μήνυμα το οποίο περιγράφει το σφάλμα και στην συνέχεια το μήνυμα τοποθετείται στον Πράκτορα Μαυροπίνακα, για να εμφανιστεί στο χρήστη, μέσα από ένα αυτοαναδυόμενο παράθυρο.

Δημιουργία Αποθηκευτικών Χώρων και Ελέγχου Διεργασιών

Αν η αίτηση είναι έγκυρη, τότε δημιουργούμε δύο αποθηκευτικούς χώρους σε μορφή *Vector* και στον ένα αποθηκευτικό χώρο, που ονομάζουμε *VisitedLinks*, αποθηκεύουμε τις ηλεκτρονικές διευθύνσεις που έχουμε επισκεφτεί και στον άλλο αποθηκευτικό χώρο, που ονομάζουμε *UnProcLinks*, τοποθετούμε τις ηλεκτρονικές διευθύνσεις που πρόκειται να επισκεφτούμε, ή πιο απλά δεν τις έχουμε ακόμα επεξεργαστεί. Στο πρώτο *Vector* οι θέσεις καταλαμβάνονται από διευθύνσεις που αποθηκεύονται σε μορφή αλφαριθμητικού ενώ στο δεύτερο *Vector* οι θέσεις καταλαμβάνονται από στιγμιότυπα της κλάσης *HyperLink* [B15]. Η εν λόγω κλάση λειτουργεί σαν μια δομή χωρίς μεθόδους αφού, σε αυτήν αποθηκεύουμε την κάθε διεύθυνση χρησιμοποιώντας την κλάση *URL* [A1], και το βάθος της σε σχέση πάντα με την αρχική σελίδα.

Η διαδικασία της αποθήκευσης ενός άρθρου σε μορφή HTML ή ακόμη και ενός ολόκληρου δικτυακού τόπου είναι μια αρκετά χρονοβόρα διαδικασία και καλό θα είναι να γίνονται αρκετές εργασίες ταυτόχρονα. Πιο συγκεκριμένα η αποθήκευση ενός και μόνο αρχείου θα πρέπει να γίνεται σε μία ανεξάρτητη διεργασία. Όμως η ανεξέλεγκτη δημιουργία πολλών ταυτόχρονων διεργασιών θα έχει αντίστροφα αποτελέσματα, αφού θα καταναλώσει όλους τους υπολογιστικούς πόρους. Άρα η βέλτιστη λύση είναι η ελεγχόμενη δημιουργία ανεξάρτητων διεργασιών, κάτι που στην περίπτωση μας επιτυγχάνεται με την δημιουργία ενός αντικειμένου της κλάσης

ThreadGuard [B16].



Σχήμα 5.6 *Το διάγραμμα δραστηριοτήτων του ΠΦΔΤ*

Το κάθε αντικείμενο αρχικοποιείται με ένα συγκεκριμένο αριθμό διεργασιών που επιτρέπεται να τρέχουν ταυτόχρονα καθ' όλη την διάρκεια ζωής του αντικειμένου *SiteMirrorRobot,* η πιο απλά, η κάθε αίτηση του χρήστη θα εξυπηρετηθεί από ένα

και μόνο συγκεκριμένο αριθμό ανεξάρτητων διεργασιών. Η αποθήκευση ενός αρχείου θα ενεργοποιείται μόλις η μέθοδος *IsAvailable* [B16], επιστρέφει μια τιμή *αληθής* και με το πέρας της αποθήκευσης του αρχείου, μετά την κλήση της μεθόδου *IncreaseAvailable* [B16], θα αυξάνονται οι προς διάθεση δεσμεμεύμενες διεργασίες.

Τέλος, προσθέτουμε την διεύθυνση της αρχική σελίδας στον *Vector* που περιέχει τους συνδέσμους που πρόκειται να επεξεργαστούμε, δηλαδή τον *UnProcLinks*. Όσο ο συγκεκριμένος Vector δεν είναι άδειος, τότε συνεχίζουμε την επαναληπτική διαδικασία όπως περιγράφεται στην συνέχεια.

Επεξεργασία του πρώτου στοιχείου του UnProcLinks

Όσο υπάρχουν διευθύνσεις προς επεξεργασία, παίρνουμε την πρώτη διεύθυνση (την κορυφή) και εν συνεχεία την διαγράφουμε από την λίστα. Έπειτα κάνουμε δύο ελέγχους : α) αν η συγκεκριμένη διεύθυνση δεν ανήκει στις διευθύνσεις που έχουμε επισκεφτεί δηλαδή στην λίστα *VisitedLinks* και β) αν θα πρέπει να την επεξεργαστούμε. Για παράδειγμα, ο χρήστης έχει βάλει σαν περιορισμό να μην ακολουθήσουμε διεύθυνση η οποία δείχνει εκτός του συγκεκριμένου υπολογιστή και η συγκεκριμένη διεύθυνση δείχνει σε εξωτερικό δικτυακό τόπο, ή, η συγκεκριμένη διεύθυνση έχει μεγαλύτερο βάθος από το επιτρεπτό. Αν η απάντηση είναι αρνητική σε οποιοδήποτε από τα παραπάνω ερωτήματα τότε επιστρέφουμε στο προηγούμενο βήμα και συνεχίζουμε με την επόμενη διεύθυνση.

Εύρεση τύπου περιεχομένου

Αν η απάντηση είναι θετική στα δύο ερωτήματα του προηγούμενου βήματος τότε συνεχίζουμε την εύρεση του τύπου του περιεχομένου [Γ11] της συγκεκριμένης διεύθυνσης. Πιο συγκεκριμένα, αν ο τύπος δεν είναι αρχείο κειμένου τότε το αρχείο δεν περιέχει άλλες διευθύνσεις και συνεχίζουμε στο στάδιο που περιγράφεται σαν *Αποθήκευση αρχείου στον τοπικό δίσκο* - δες παρακάτω.

Ανάκτηση και επεξεργασία του κειμένου

Αν ο *τύπος περιεχομένου* του αρχείου περιγράφεται σαν *text/html* τότε, αφού ανακτήσουμε το αρχείο εν συνεχεία το σαρώνουμε, για να αποθηκεύσουμε όλες τις διευθύνσεις που υπάρχουν σε αυτό και περιγράφονται από τα διακριτικά της γλώσσας HTML, <a href, <img, <frame, την δήλωση background στα διακριτικά <table, <body, την δήλωση src στο διακριτικό <embed, και τέλος την δήλωση code στο

διακριτικό <applet. Όλες αυτές οι διευθύνσεις θα προστεθούν στην λίστα των διευθύνσεων προς επεξεργασία αν και εφόσον αν δεν υπάρχουν στην λίστα με τις διευθύνσεις που έχουμε επισκεφθεί και αν δεν υπάρχει άλλο κώλυμα που περιγράφεται στην αίτηση του χρήστη. Τέλος το κάθε αρχείο κειμένου αποθηκεύεται στον τοπικό δίσκο αφού πρώτα οι διευθύνσεις έχουν αλλάξει ώστε να *δείχνουν* στα τοπικά αρχεία.

Αποθήκευση αρχείου στον τοπικό δίσκο

Κάθε αρχείο το αποθηκεύουμε στον τοπικό δίσκο δημιουργώντας ένα στιγμιότυπο της κλάσης *URLHandle* [B25] το οποίο εκτελείται σαν ανεξάρτητη διεργασία αφού κληρονομεί όλες τις μεθόδους της κλάσης *Thread* [A2], και αρχικοποιείται με τρεις παραμέτρους : α) την πλήρη ηλεκτρονική διεύθυνση του αρχείου β) το μονοπάτι αποθήκευσης στον τοπικό δίσκο γ) το στιγμιότυπο της κλάσης *ThreadGuard,* το οποίο και ελέγχει το πλήθος των διεργασιών.

Αποθήκευση Μηνύματος στον Πράκτορα Μαυροπίνακα

Αφού έχει ολοκληρωθεί η διαδικασία τότε δημιουργείται ένα μήνυμα που περιγράφει το τυχόν σφάλμα ή την επιτυχία της συγκεκριμένης αίτησης του χρήστη. Αυτό το μήνυμα καταχωρείται στον Πράκτορα Μαυροπίνακα - δες παρακάτω - ώστε να είναι διαθέσιμο προς ενημέρωση του χρήστη.

Όλες οι σελίδες και τα αρχεία που έχει *κατεβάσει* από το διαδίκτυο, ο ΠΦΔΤ, αποθηκεύονται - εκτός και αν ο χρήστης επιλέξει κάτι διαφορετικό - στον Πράκτορα Διακομιστή Διαδικτυακού Κόμβου, ώστε να παρέχει την αίσθηση στον χρήστη ότι βρίσκεται συνεχώς συνδεδεμένος στο διαδίκτυο.

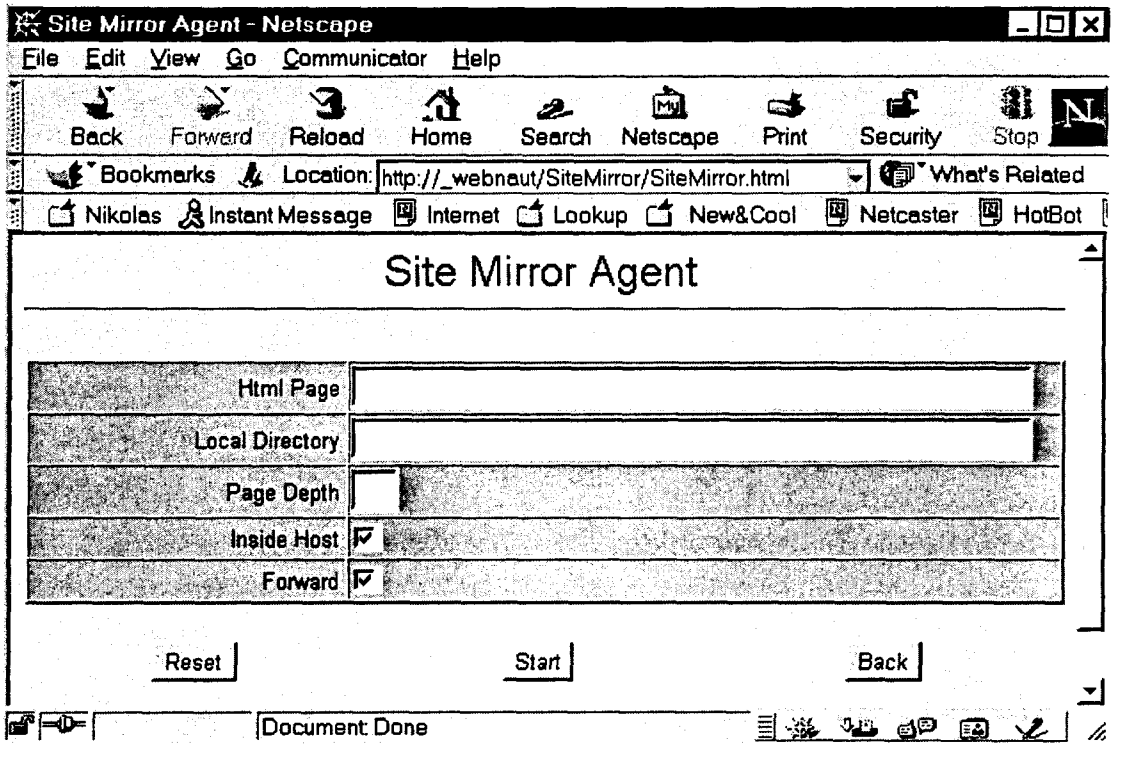
5.4.3 Σχεδίαση Διεπαφής του Πράκτορα Φιλοξενούμενων Δικτυακών Τόπων

Η λειτουργία του ΠΦΔΤ ενεργοποιείται με δύο τρόπους μέσα από τα αντίστοιχα εικονίδια στην πινακίδα κεντρικού ελέγχου. Στην πρώτη περίπτωση ο χρήστης μέσω μίας διεπαφής, όπως εμφανίζεται στο σχήμα 5.7, μπορεί να ορίσει όλες τις παραμέτρους της λειτουργίας του πράκτορα, όπως :

- την ηλεκτρονική διεύθυνση της αρχικής σελίδας του άρθρου ή του δικτυακού

κόμβου που θέλουμε να αποθηκεύσουμε στο τοπικό δίσκο

- τον τοπικό φάκελο αποθήκευσης
- το βάθος του μονοπατιού
- τον περιορισμό της ανάκτησης αρχείων που βρίσκονται εκτός του εξυπηρετητή της αρχικής διεύθυνσης
- την περιορισμό της ανάκτησης αρχείων που βρίσκονται σε μικρότερο βάθος μονοπατιού από το βάθος του καταλόγου που περιέχει την αρχική διεύθυνση



Σχήμα 5.7 Η διεπαφή του ΠΦΔΤ

Στην δεύτερη περίπτωση, η ενεργοποίηση επιτυγχάνεται πάλι πατώντας απλά ένα κουμπί στην πινακίδα κεντρικού ελέγχου γεγονός που ερμηνεύεται από τον πράκτορα σαν εντολή αποθήκευσης της τρέχουσας σελίδας και όλων των συνδέσμων που περιέχει ανεξαρτήτως βάθους με μόνο περιορισμό να μην ακολουθήσει σύνδεσμο που δείχνει έξω από τον τρέχον εξυπηρετητή του διαδικτύου.

5.5 Σχεδίαση του Πράκτορα Μαυροπίνακα

Ο Πράκτορας Μαυροπίνακας είναι ένας πράκτορας λογισμικού ο οποίος δεν διαθέτει κάποια διεπαφή για την απευθείας επικοινωνία με το χρήστη αλλά χρησιμοποιείται

εσωτερικά από το σύστημα. Η δημιουργία του συγκεκριμένου πράκτορα προέκυψε κατά την φάση του ελέγχου των διεργασιών ως προς την επικοινωνία τους με το χρήστη και πιο συγκεκριμένα στο στάδιο της ολοκλήρωσης ορισμένων χρονοβόρων εργασιών.

Για παράδειγμα, κάποια χρονική στιγμή ο χρήστης απευθύνει μια αίτηση στο σύστημα για να αποθηκεύσει ένα ενδιαφέρον άρθρο στον τοπικό του δίσκο. Ο έλεγχος και η ολοκλήρωση της συγκεκριμένης ενέργειας θα γίνει από τον Πράκτορα Φιλοξενούμενων Διακτυακών Τόπων - ΠΦΔΤ. Όμως η συγκεκριμένη ενέργεια, τις περισσότερες φορές, είναι αρκετά χρονοβόρα και για αυτό το λόγο τρέχει σαν ανεξάρτητη διεργασία όπως είδαμε στην προηγούμενη ενότητα. Ο ΠΦΔΤ με την παραλαβή της αίτησης του χρήστη αφού ενεργοποιήσει την διεργασία εν συνεχεία ανακατευθύνει τον χρήστη σε μία στατική σελίδα του συστήματος με σκοπό να τον ενημερώσει ότι η αίτηση έγινε κατανοητή και σε λίγη ώρα θα έχει τελειώσει την εργασία που του ανέθεσε ο χρήστης.

Η εν λόγω εργασία μπορεί να ολοκληρωθεί μετά από αρκετή ώρα αλλά δεν υπάρχει τρόπος να ενημερωθεί ο χρήστης, ο οποίος εκείνη την στιγμή μπορεί να αλληλεπιδρά με άλλον πράκτορα, ότι η προηγούμενη εντολή έχει ολοκληρωθεί με επιτυχία ή ότι προέκυψαν διάφορα σφάλματα. Έτσι λοιπόν θα πρέπει να υπάρχει ένας κοινός αποθηκευτικός χώρος όπου όλοι οι πράκτορες γράφουν διάφορα μηνύματα τα οποία φροντίζει να τα εμφανίζει ο Πράκτορας Διακομιστής Διαμεσολάβησης κατά την φάση της εμφάνισης της πινακίδας κεντρικού ελέγχου.

5.5.1 Λεπτομερής Σχεδιασμός του Πράκτορα Μαυροπίνακα

Ο Πράκτορας Μαυροπίνακας αποτελείται από την κλάση *Blackboard* [B19]. Όλες οι παράμετροι λειτουργίας του πράκτορα αρχικοποιούνται μέσω της μεθόδου *ReadProperties* και εν συνεχεία αποθηκεύονται σε ένα στατικό *Hashtable.* Το αποτέλεσμα της μεθόδου *ReadProperties,* η οποία δείχνει την έγκυρη ή μη ενεργοποίηση του συγκεκριμένου πράκτορα, επιστρέφει η δημόσια μέθοδος *initialize* η οποία με την σειρά της έχει ενεργοποιηθεί από την μέθοδο *StartModules* της κλάσης *ProxyServer.* Η υλοποίηση της δημόσιας μεθόδου *addEntry* παρέχει την δυνατότητα σε οποιοδήποτε άλλο πράκτορα να προσθέσει ένα οποιοδήποτε μορφοποιημένο μήνυμα στον εσωτερικό μαυροπίνακα ο οποίος ονομάζεται *BBMessages* και έχει την μορφή *Vector* [A23].

Η δεύτερη δημόσια μέθοδος *getMessages* επιστρέφει κώδικα σε γλώσσα Javascript [Γ10] ώστε να ανοίξει ένα νέο παράθυρο στο πρόγραμμα πλοήγησης του χρήστη και εκεί μέσα να εμφανίσει όλα τα μηνύματα που υπάρχουν μέχρι εκείνη τη στιγμή στο μαυροπίνακα. Η μέθοδος *getMessages,* προτού επιστρέψει τον κώδικα ο οποίος θα ενσωματωθεί στην σελίδα που βλέπει ο χρήστης, αδειάζει το *Vector BBMessages* ώστε να γεμίσει με τα νέα μηνύματα των πρακτόρων. Την μέθοδο *getMessages* καλεί κάθε φορά πριν την εισαγωγή της πινακίδας κεντρικού ελέγχου, το στιγμιότυπο της κλάσης *ProxyClient* που χειρίζεται την τρέχουσα αίτηση του χρήστη [Β2].

5.6 Σχεδίαση του Πράκτορα Εξυπηρετητή Διαδικτυακού Κόμβου

Ο Πράκτορας Εξυπηρετητής Διαδικτυακού Κόμβου - ΠΕΔΚ, δημιουργεί έναν εικονικό κόμβο εξυπηρέτησης κλήσεων του πρωτοκόλλου HTTP, ο οποίος χρησιμοποιείται από όλους του άλλους πράκτορες και σκοπό έχει να υποστηρίξει τοπικές πληροφορίες του συστήματος. Για παράδειγμα, όλες οι σελίδες που είναι σχετικές με την διεπαφή του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών και του Πράκτορα Φιλοξενούμενων Δικτυακών Τόπων καθώς επίσης και οι σελίδες σχετικά με τις ρυθμίσεις και την βοήθεια του συστήματος, κάποιες στατικές σελίδες που περιγράφουν σφάλματα, βρίσκονται αποθηκευμένες στο ΠΕΔΚ.

Θα πρέπει να τονίσουμε ότι ο εν λόγω πράκτορας επέλυσε πολλά ζητήματα κατά την ανάπτυξη και ενοποίηση του συστήματος Webnaut, αφού διάφορα μηνύματα που θα έπρεπε να κωδικοποιηθούν εσωτερικά στους πράκτορες τώρα είναι διαθέσιμα από το ΠΕΔΚ.

Το μεγαλύτερο όφελος από την ανάπτυξη και χρήση του ΠΕΔΚ είναι ότι αυξάνει την ομοιογένεια και την συνοχή του συστήματος αφού δίνει την εντύπωση στον χρήστη ότι βρίσκεται συνδεδεμένος στο διαδίκτυο αλλά και από την άλλη πλευρά διευκολύνει την μετατροπή του συστήματος στις ιδιαίτερες ανάγκες κάποιων ειδικών ομάδων χρηστών αφού ή διεπαφή είναι ανεξάρτητη από την κύρια λειτουργία του συστήματος.

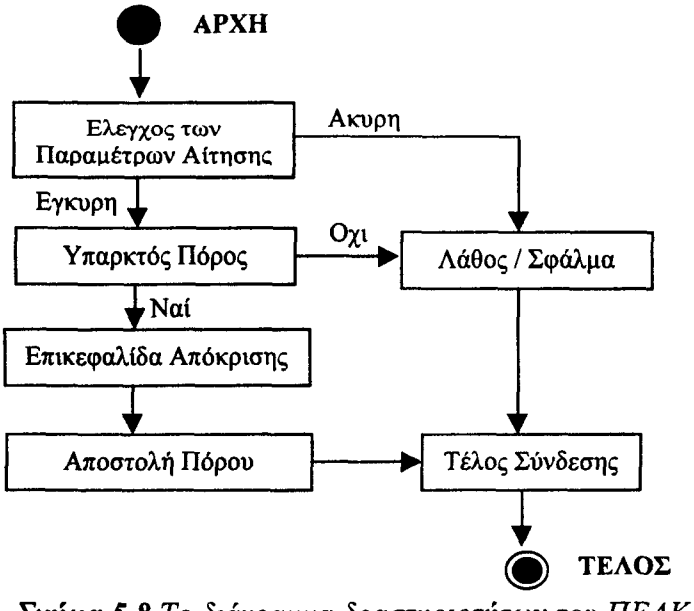
5.6.1 Λεπτομερής Σχεδιασμός του Πράκτορα Εξυπηρετητή Διαδικτυακού Κόμβου

Ο ΠΕΔΚ αποτελείται από δύο κλάσεις : α) την κλάση *WWWServer* [B3] και β) την κλάση *WWWClient* [B4]. Η κλάση *WWWServer* έχει όλες τις προδιαγραφές να λειτουργήσει σαν αυτόνομος διακομιστής εξυπηρέτησης αιτήσεων αλλά στα πλαίσια του συστήματος Webnaut, η μόνη εργασία της είναι να αρχικοποιεί μέσω της μεθόδου *ReadWWWProperties,* τις παραμέτρους λειτουργίας του διακομιστή και εν συνεχεία να επιστρέφει μία τιμή σαν ένδειξη έγκυρης ή μη ενεργοποίησης του διακομιστή. Η μέθοδος *StartModules* της κλάσης *ProxyServer,* δημιουργεί το πρώτο αντικείμενο της κλάσης *WWWServer* και εν συνεχεία καλεί την μέθοδο *initialize*, η οποία με την σειρά της επιστρέφει το αποτέλεσμα της μεθόδου *ReadWWWProperties.*

Ο ΠΔΔ δημιουργεί απευθείας αντικείμενα της κλάσης *WWWClient* για να χειριστούν τις αιτήσεις του προγράμματος πλοήγησης σχετικά με τους τοπικούς πόρους. Όλες αυτές οι αιτήσεις εξυπηρετούνται από ανεξάρτητες διεργασίες αφού η εν λόγω κλάση κληρονομεί τις μεθόδους της κλάσης *WWWServer* και υλοποιεί την διεπαφή *Runnable.*

5.6.2 Λειτουργικότητα του Πράκτορα Εξυπηρετητή Διαδικτυακού Κόμβου

Το σχήμα 5.8 παρουσιάζει το διάγραμμα δραστηριοτήτων του ΠΕΔΚ. Οι ενέργειες του πράκτορα με την σειρά με την οποία εκτελούνται, είναι οι εξής :



Σχήμα 5.8 Το διάγραμμα δραστηριοτήτων του ΠΕΔΚ

Έλεγχος των Παραμέτρων Αίτησης

Αρχικά, η ανεξάρτητη διεργασία *WWWClient* κάνει δύο ενέργειες α) ελέγχει την αίτηση του χρήστη αν είναι έγκυρη. Ο έλεγχος δεν έγκειται μόνο στο πρωτόκολλο που χρησιμοποιείται αλλά και στο είδος της αίτησης, και πιο συγκεκριμένα ο ΠΕΔΚ, εξυπηρετεί μόνο αιτήσεις GET και HEAD. Σε διαφορετική περίπτωση ο πελάτης ανακατευθύνεται σε μία σελίδα σφάλματος αφού φυσικά έχει τοποθετηθεί στην επικεφαλίδα ο κωδικός απόκρισης 405, σαν ένδειξη ότι έχει χρησιμοποιηθεί στην αίτηση μια μη υποστηριζόμενη μέθοδος.

Υπαρκτός Πόρος

Στην περίπτωση που η αίτηση είναι έγκυρη θα γίνει ένας έλεγχος αν η αίτηση αφορά κάποιο υπαρκτό αντικείμενο, δηλαδή αν υπάρχει η όχι ο πόρος της αίτησης. Σε περίπτωση που δεν υπάρχει, θα τοποθετηθεί στην επικεφαλίδα ο κωδικός απόκρισης 404, σαν ένδειξη ότι δεν υπάρχει ο πόρος στον τοπικό διακομιστή.

Επικεφαλίδα Απόκρισης

Αν η αίτηση αφορά υπαρκτό πόρο του διακομιστή τότε ο κωδικός απόκρισης παίρνει τιμή 200 για να επισημάνει ότι η αίτηση έγινε κατανοητή. Εν συνεχεία, ανάλογα με την κατάληξη του πόρου, θέτουμε τον τύπο περιεχομένου, το μέγεθος της απόκρισης, διάφορα στοιχεία σχετικά με το διακομιστή και τέλος την τοπική ημ/νια.

Αποστολή Πόρου

Σε αυτό το στάδιο γίνεται η αποστολή του πόρου που αιτείται ο χρήστης. Η πιο συνηθισμένη περίπτωση είναι η αίτηση να αναφέρεται σε κάποιο συγκεκριμένο αρχείο οπότε και τα περιεχόμενα του αρχείου αποστέλλονται στον πελάτη. Στην περίπτωση όμως που η αίτηση δεν καθορίζει κάποιο συγκεκριμένο αρχείο αλλά κάποιο φάκελο του διακομιστή, τότε γίνεται ένας έλεγχος αν στον εν λόγω φάκελο υπάρχει κάποιο προκαθορισμένο αρχείο (π.χ. index.html) και αν η απάντηση είναι θετική τότε αποστέλλεται το συγκεκριμένο αρχείο, διαφορετικά αποστέλλονται τα περιεχόμενα του εν λόγω φακέλου.

5.7 Σχεδίαση του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών

Ένα σύστημα που απευθύνεται στους χρήστες του διαδικτύου με σκοπό την υπόδειξη

πηγών πληροφοριών οι οποίες είναι σύμφωνες με τα ενδιαφέροντα τους θα πρέπει να διαθέτει έναν αυτόνομο μηχανισμό αναζήτησης πληροφοριών. Οι μηχανές αναζήτησης αποτελούν το πρώτο βήμα για τον εντοπισμό ενός κόμβου που διαθέτει πληροφορίες σύμφωνες με τα ενδιαφέροντα ενός χρήστη.

Οι μηχανές αναζήτησης, χρησιμοποιώντας ειδικά προγράμματα που ονομάζονται spiders, robots ή απλώς bots, προσπαθούν να οργανώσουν την διαθέσιμη πληροφορία του διαδικτύου σε καταλόγους ώστε να είναι εύκολη ή έρευνα και ο εντοπισμός των διαδικτυακών τόπων που ενδιαφέρουν έναν χρήστη. Οι μετα-μηχανές αναζήτησης έρχονται να καλύψουν την ανάγκη των χρηστών για α) ομαδοποίηση των αποτελεσμάτων μιας ερώτησης από διαφορετικές μηχανές αναζήτησης, β) την έλλειψη πρόσβασης στον κατάλογο μιας μηχανής αναζήτησης λόγω δικτυακών προβλημάτων, γ) την δημιουργία κοινής διεπαφής για αρκετές μηχανές αναζήτησης, και το σημαντικότερο από όλα, δ) την κάλυψη όσο το δυνατό περισσότερων θεμάτων αφού όπως είναι γνωστό καμία μηχανή αναζήτησης δεν καλύπτει όλα τα θέματα.

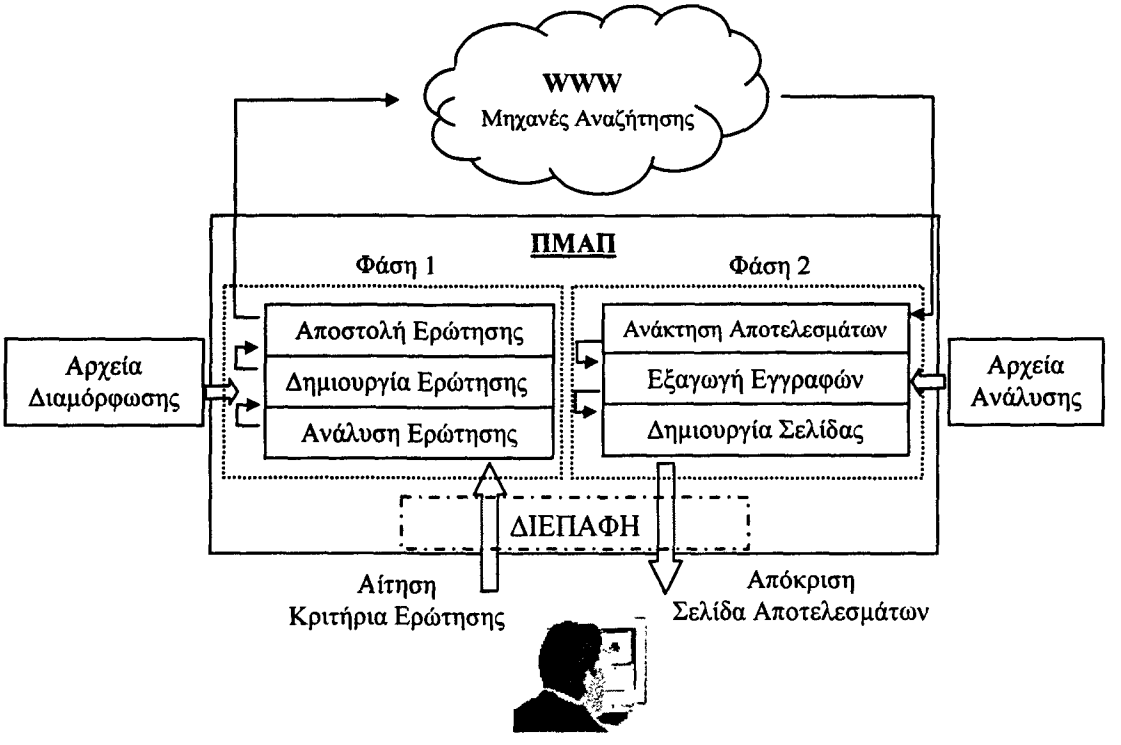
Ο Πράκτορας Μεταμηχανή Αναζήτησης Πληροφοριών - ΠΜΑΠ, είναι μια μεταμηχανή αναζήτησης πληροφοριών στο διαδίκτυο που σκοπό έχει να προσφέρει στον χρήστη μια κοινή διεπαφή ώστε να απευθύνει ερωτήσεις σε πέντε αρκετά γνωστές μηχανές αναζήτησης - AltaVista [Γ14], Excite [Γ15], Hotbot [Γ16], Lycos [Γ17], Yahoo [Γ18].

5.7.1 Λειτουργικότητα του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών

Κατά την αρχική σχεδίαση του εν λόγω πράκτορα θέλαμε η λειτουργία του να είναι όσο το δυνατόν πιο ευέλικτη και επιπλέον να υπάρχει η δυνατότητα αναβάθμισης, δηλαδή να μπορούμε να ενσωματώσουμε και νέες μηχανές αναζήτησης αν αυτό κριθεί αναγκαίο. Έτσι λοιπόν, τον τρόπο λειτουργίας του ΠΜΑΠ όπως φαίνεται και στο σχήμα 5.9 τον έχουμε χωρίσει σε δύο φάσεις.

Φάση 1 : Αυτή θα μπορούσαμε να την ονομάσουμε και φάση ερώτησης αφού κατά την διάρκεια της θα συνταχθεί η ερώτηση του χρήστη σύμφωνα με τα πρότυπα της κάθε μηχανής αναζήτησης. Επιπλέον, κατά την φάση αυτή θα γίνει και η αποστολή της ερώτησης στην συγκεκριμένη μηχανή αναζήτησης. Το πρώτο στάδιο της πρώτης φάσης είναι η *Ανάλυση της Ερώτησης* του χρήστη. Σε αυτό το στάδιο, όλα τα δεδομένα από την αίτηση του χρήστη συλλέγονται και ελέγχονται ως προς την

πληρότητα και την ορθότητα τους. Στην περίπτωση που η αίτηση εμπεριέχει σφάλματα τότε ο χρήστης ενημερώνεται μέσω μιας ενημερωτικής σελίδας που περιγράφει το είδος του σφάλματος.



Σχήμα 5.9 *Οι δύο φάσεις λειτουργίας του ΠΜΑΠ*

Κατά την διάρκεια της *Δημιουργίας της Ερώτησης*, η ερώτηση του χρήστη θα τροποποιηθεί κατάλληλα ώστε να εμπεριέχει τους λογικούς τελεστές στην μορφή που είναι αποδεκτοί από την μηχανή αναζήτησης που θα χρησιμοποιηθεί για την άντληση των αποτελεσμάτων. Για παράδειγμα, άλλες μηχανές αναζήτησης δέχονται την λέξη AND για την σύζευξη των λέξεων ενώ άλλες δέχονται το σύμβολο +. Στην υλοποίηση μας, υποστηρίζουμε τους ακόλουθους τελεστές "AND", "OR", "NOT", "EXACT","EXACT NOT" όταν φυσικά τους υποστηρίζει και η αντίστοιχη μηχανή αναζήτησης.

Το τελευταίο στάδιο της πρώτης φάσης είναι η *Απόστολή της Ερώτησης*, όπου ο ΠΜΑΠ εμπλουτίζει την ερώτηση με συγκεκριμένες ετικέτες, που είναι αναγκαίες για το σχηματισμό της ερώτησης, και χρησιμοποιώντας την ενδεδειγμένη μέθοδο αποστολής της φόρμας προωθεί την ερώτηση στο πρόγραμμα της μηχανής αναζήτησης. Όλες αυτές οι πληροφορίες για την κάθε μηχανή αναζήτησης, δηλαδή ποία είναι η διεύθυνση της, την πόρτα που πρέπει να συνδεθεί ο πελάτης, τους λογικούς τελεστές που υποστηρίζει, μαζί με άλλες πληροφορίες περιγράφονται στα

Αρχεία Διαμόρφωσης - σχήμα 5.10.

```
<=s=>  WEBNAME   <==>  ALTAVISTA          <=e=>
<=s=>  WEBHOST   <==>  www.altavista.com   <=e=>
.....
<=s=>  WEBLINK   <==>  /cgi-bin/query      <=e=>
<=s=>  WEBPORT   <==>  80                  <=e=>
```

Σχήμα 5.10 *Τμήμα του Αρχείου Διαμόρφωσης της μηχανής αναζήτησης AltaVista*

Φάση 2 : Αυτή η φάση είναι η συνέχεια της προηγούμενης και θα μπορούσαμε να την ονομάσουμε φάση ανάλυσης των αποτελεσμάτων. Στην διάρκεια της δεύτερης φάσης, θα συλλέξουμε τα αποτελέσματα της κάθε μηχανής αναζήτησης. Αφού η κάθε μηχανή αναζήτησης χρησιμοποιεί διαφορετικό τρόπο παρουσίασης των αποτελεσμάτων θα πρέπει να χρησιμοποιηθεί και διαφορετική μεθοδολογία για την εξαγωγή των αποτελεσμάτων. Όπως και η πρώτη φάση έτσι και αυτή χωρίζεται σε τρία στάδια.

Το στάδιο της *Ανάκτησης των Αποτελεσμάτων* περιλαμβάνει την ανάκτηση των αποτελεσμάτων από την κάθε μηχανή αναζήτησης ξεχωριστά και την προώθηση τους προς επεξεργασία.

Στο επόμενο στάδιο, που ονομάζεται *Εξαγωγή Εγγραφών* τα αποτελέσματα εξάγονται από τις σελίδες των αποτελεσμάτων. Η κάθε μηχανή αναζήτησης χρησιμοποιεί διαφορετική μορφοποίηση αποτελεσμάτων και μάλιστα περιλαμβάνει και άσχετες πληροφορίες που δεν έχουν να κάνουν σε τίποτα με την ερώτηση του χρήστη (π.χ. διαφημίσεις, σύνδεσμοι σε άλλους δικτυακούς τόπους, βοήθεια σχηματισμού ερωτήσεων κ.λπ.)

```
<=s=>  str     <==>  [<b>]  <==>  s    <=e=>
<=s=>  mch     <==>  [.]    <==>  ec   <=e=>
<=s=>  ign     <==>  [>]    <==>  c    <=e=>
.....
<=s=>end       <==>  [br>]  <==>  es   <=e=>
```

Σχήμα 5.11 *Τμήμα του Αρχείου Ανάλυσης της μηχανής αναζήτησης AltaVista*

Έτσι λοιπόν, κατά την διάρκεια αυτού του σταδίου, ανάλογα με την μηχανή αναζήτησης θα χρησιμοποιηθεί και το κατάλληλο *Αρχείο Ανάλυσης* το οποίο περιγράφει την ακριβή μεθοδολογία εξαγωγής ενός και μόνο αποτελέσματος από την συγκεκριμένη μηχανή αναζήτησης. Ο ΠΜΑΠ αναλαμβάνει την επαναληπτική εφαρμογή της μεθοδολογίας σε ολόκληρη την σελίδα των αποτελεσμάτων μέχρι να

εξαχθούν όλα τα αποτελέσματα. Το σχήμα 5.11 παρουσιάζει ένα τμήμα του *Αρχείου Ανάλυσης* της μηχανής αναζήτησης AltaVista.

Στο τελευταίο στάδιο, της *Δημιουργίας Σελίδας,* όλα τα αποτελέσματα από όλες τις μηχανές αναζήτησης οργανώνονται και παρουσιάζονται σε μία ενιαία σελίδα. Η παρούσα υλοποίηση παρουσιάζει όλα τα αποτελέσματα από την κάθε μηχανή αναζήτησης το ένα μετά το άλλο με την μορφή <Τίτλος><Ηλεκτρονική Διεύθυνση><Περιγραφή>. Επίσης κατά την διάρκεια αυτού του σταδίου, απομακρύνονται, αν φυσικά και ο χρήστης το επιθυμεί, όλα τα αποτελέσματα τα οποία έχουν τον ίδιο τίτλο ή ηλεκτρονική διεύθυνση.

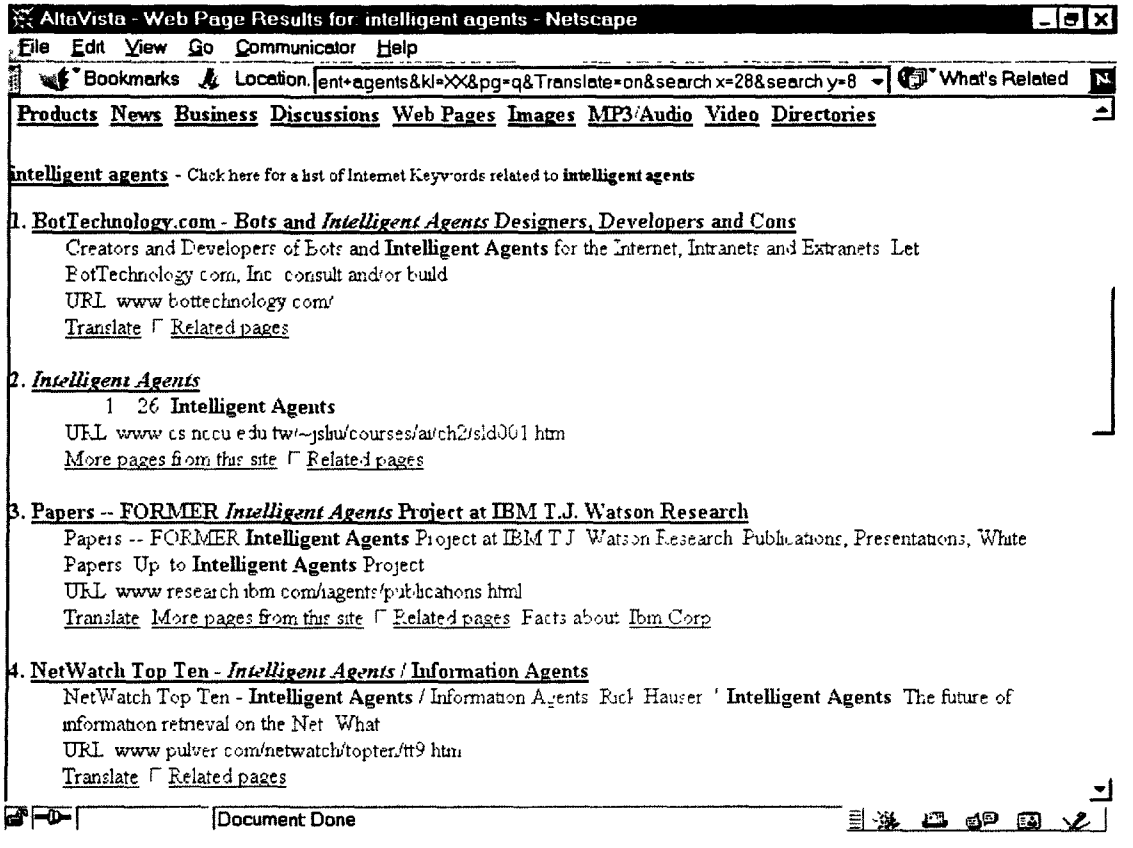
5.7.2 Λεπτομερής Σχεδιασμός του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών

Ο ΠΜΑΠ αποτελείται από πέντε κλάσεις : α) *SpiderServer*[B7], β) *SpiderClient* [B8], γ) *SpiderParser* [B9], δ) *SpiderEngines* [B10], ε) *SpiderLink* [B11]. Η κλάση *SpiderServer* έχει όλες τις προδιαγραφές να λειτουργήσει σαν αυτόνομος διακομιστής εξυπηρέτησης αιτήσεων αλλά στα πλαίσια του συστήματος Webnaut, δύο είναι οι εργασίες που επιτελεί : α) να αρχικοποιεί μέσω της μεθόδου *ReadSpiderProperties,* τις παραμέτρους λειτουργίας του διακομιστή και εν συνεχεία να επιστρέφει μία τιμή σαν ένδειξη της έγκυρης ή μη ενεργοποίησης του.

Η μέθοδος *StartModules* της κλάσης *ProxyServer,* δημιουργεί το πρώτο αντικείμενο της κλάσης *SpiderServer* και εν συνεχεία καλεί την μέθοδο *initialize,* η οποία με την σειρά της επιστρέφει την ένωση των αποτελεσμάτων από δύο μεθόδους i) το αποτέλεσμα της μεθόδου *ReadSpiderProperties* και ii) το αποτέλεσμα της μεθόδου *initialize* [B10] μετά την δημιουργία του πρώτου και μοναδικού αντικειμένου της κλάσης *SpiderEngines* β) να δημιουργεί στιγμιότυπα της κλάσης *SpiderClient,* με ορίσματα τα αντικείμενα *HttpRequest* και *HttpResponse* που παρέλαβε από τον ΠΔΔ.

Η κλάση *SpiderEngines* περιέχει όλα εκείνα τα στοιχεία που περιγράφουν μια μηχανή αναζήτησης. Πιο συγκεκριμένα, όπως βλέπουμε στο σχήμα 5.9 κατά την διάρκεια των δύο φάσεων λειτουργίας του ΠΜΑΔ χρησιμοποιούνται δύο είδη διαφορετικών αρχείων, τα *Αρχεία Διαμόρφωσης* και τα *Αρχεία Ανάλυσης.* Η κάθε μηχανή αναζήτησης περιγράφεται από ένα τέτοιο ζευγάρι αρχείων όπου στο μεν πρώτο περιγράφονται γενικά στοιχεία για την μηχανή αναζήτησης, όπως για

παράδειγμα, ο δικτυακός τόπος, το είδος της αίτησης, οι τελεστές που υποστηρίζει κ.λπ. και στο δεύτερο αρχείο περιγράφεται ο τρόπος με τον οποίο μπορούμε να αντλήσουμε ένα αποτέλεσμα από την σελίδα των αποτελεσμάτων. Για την ανάγνωση των δύο αρχείων χρησιμοποιούνται οι βοηθητικές κλάσεις *TagBuilder* [B23] και *TagText* [B24].



Σχήμα 5.12 *Η απόκριση της μηχανής αναζήτησης AltaVista στην ερώτηση "Intelligent Agents"*

Η μορφοποίηση των *Αρχείων Διαμόρφωσης* ακολουθεί την μορφοποίηση των αρχείων παραμέτρων, των πρακτόρων του συστήματος. Όλα αυτά τα ζευγάρια, παράμετρος - τιμή, αποθηκεύονται σε ένα *Hashtable,* ο οποίος με την σειρά του αποθηκεύεται σε έναν άλλον *Hashtable,* ο οποίος έχει σαν *κλειδί* το όνομα της μηχανής αναζήτησης. Αυτός ο δεύτερος *Hashtable Options* είναι στατικός ώστε να τον μοιράζονται όλα τα στιγμιότυπα της κλάσης *SpiderEngines.*

Ο δεύτερος *Hashtable της κλάσης SpiderEngines* ονομάζεται *Parsers* και είναι και αυτός στατικός και μάλιστα - όπως και ο προηγούμενος - έχει στο κάθε κλειδί το όνομα μιας μηχανής αναζήτησης,, αλλά στην θέση της τιμής έχει ένα *Vector* από αντικείμενα *TagTexts,* και το κάθε ένα αντικείμενο περιγράφει μία σειρά από το

Αρχείο Ανάλυσης της μηχανής αναζήτησης που δείχνει το κλειδί του.

Για να μπορούμε να διαβάσουμε την χρήσιμη πληροφορία σε μία σελίδα αποτελεσμάτων μιας μηχανής αναζήτησης θα έπρεπε κατ' αρχήν να ορίσουμε ποία είναι η χρήσιμη πληροφορία. Έτσι λοιπόν παρατηρώντας το σχήμα 5.12 μπορούμε να ορίσουμε ότι υπάρχουν 4 αποτελέσματα - αναφέρουμε όσα φαίνονται στο σχήμα - και το κάθε ένα από αυτά έχει 3 πεδία, α) μια *Ηλεκτρονική Διεύθυνση*, β) μία *Περιγραφή*, *γ)* μία *Περίληψη*. Το καθένα αποτέλεσμα είναι μια *εγγραφή*, δηλαδή είναι ένα πακέτο πληροφοριών. Ο πίνακας 5.1 δείχνει τις συγκεκριμένες τιμές που έχει η πρώτη εγγραφή στην σελίδα των αποτελεσμάτων.

Πίνακας 5.1 *Κάθε εγγραφή είναι ένα πακέτο πληροφοριών.*

Στοιχεία Εγγραφής	**Συγκεκριμένες Τιμές**
Ηλ. Διεύθυνση	http://www.bottechnology.com/
Περιγραφή	BotTechnology.com - Bots and Intelligent Agents Designers, Developers and Cons
Περίληψη	Creators and Developers of Bots and Intelligent Agents for the Internet, Intranets and Extranets. Let BotTechnology.com, Inc. consult and/or build ...

Σχεδόν όλες οι μηχανές αναζήτησης παρουσιάζουν, μέσω μιας επαναληπτικής διαδικασίας, τα αποτελέσματα τους περίπου στην μορφή της σχήματος 5.12. Η άντληση των εγγραφών, σε μία οποιαδήποτε σελίδα αποτελεσμάτων, δεν μπορεί να γίνει χωρίς να μελετήσουμε την μορφοποίηση τους, σε γλώσσα HTML, και να ορίσουμε ειδικούς τελεστές για την αναγνώριση μιας και μόνο εγγραφής, κάτι δηλαδή σαν πρότυπο. Και στην συνέχεια, αφού τα αποτελέσματα παράγονται μέσω μιας επαναληπτικής διαδικασίας, να εφαρμόσουμε το πρότυπο επαναληπτικά στην σελίδα των αποτελεσμάτων.

Αν παρατηρήσουμε το σχήμα 5.13 το οποίο παρουσιάζει την πρώτη εγγραφή της σελίδας των αποτελεσμάτων της μηχανής αναζήτησης AltaVista, στην ερώτηση "Intelligent Agents", θα μπορούσαμε αν είχαμε ορίσει έναν εικονικό δείκτη να τον μετακινούμε στην σελίδα και να διαβάζουμε τα τμήματα που αποτελούν μία εγγραφή.

Για παράδειγμα, μια μετακίνηση αρχικά στο διακριτικό της γλώσσας HTML **<b>** θα μπορούσε να ενεργοποιήσει την αρχή αναζήτησης μιας εγγραφής. Έπειτα, μια μετακίνηση μέχρι του σημείου =" και εν συνεχεία αρχίζουμε να διαβάζουμε την διεύθυνση της εγγραφής μέχρι να συναντήσουμε το χαρακτήρα ". Συνεχίζοντας κατά

αυτόν τον τρόπο μπορούμε να δημιουργήσουμε ένα πρότυπο εξαγωγής των εγγραφών από μια σελίδα αποτελεσμάτων για μια συγκεκριμένη μηχανή αναζήτησης.

```
.....
.....
<b class=txt2>1.</b>
<b>
<a href="http://www.bottechnology.com/">BotTechnology.com - Bots and
<EM>Intelligent Agents</EM> Designers, Developers and Cons</a></b>
<dd>
Creators and Developers of Bots and <B>Intelligent Agents</B> for the Internet,
Intranets and Extranets. Let BotTechnology.com, Inc. consult and/or build
...<br>
<span class=ft>
 URL: www.bottechnology.com/
</span>
.....
```

Σχήμα 5.13 *Μέρος του πηγαίου κώδικα της σελίδας των αποτελεσμάτων της μηχανής αναζήτησης AltaVista.*

Άρα λοιπόν τα *Αρχεία Ανάλυσης* περιγράφουν τα πρότυπα εξαγωγής των εγγραφών και αποτελούνται από τριάδες :

<=s=> παράμετρος <==> τιμή <==> τύπος <=e=>

Οι παράμετροι που έχουμε ορίσει είναι οι παρακάτω :

str (start) : Μετακίνηση του δείκτη στο αλφαριθμητικό (ή χαρακτήρα) που περιγράφει το πεδίο τιμή. Πρακτικά, τοποθετούμε τον εικονικό δείκτη στην αρχή της εγγραφής. Αν δεν υπάρχει το αλφαριθμητικό τιμή, αυτό σημαίνει ότι τελείωσε η διαδικασία ανάλυσης των αποτελεσμάτων.

mch (match) : οι επόμενοι χαρακτήρες θα πρέπει να είναι αριθμός ο οποίος να ακολουθείται από το αλφαριθμητικό (ή χαρακτήρα) που περιγράφει το αντίστοιχο πεδίο τιμή. Πρακτικά, χρησιμοποιείται σαν παράμετρος - φύλακας για τις μηχανές αναζήτησης που χρησιμοποιούν αρίθμηση κατά την παρουσίαση των αποτελεσμάτων τους.

ign (ignore) : Μετακίνηση του εικονικού δείκτη στο αλφαριθμητικό (ή χαρακτήρα) που περιγράφει το πεδίο τιμή.

lnk (link) : Η ηλεκτρονική διεύθυνση της εγγραφής είναι όλοι οι χαρακτήρες από το

σημείο του δείκτη μέχρι του αλφαριθμητικού (ή χαρακτήρα) που περιγράφει το πεδίο τιμή. Μετακίνησε το δείκτη στο τέλος του πεδίου τιμή.

dsc (description) : Η περιγραφή - τίτλος - μίας εγγραφής είναι όλοι οι χαρακτήρες από το σημείο του δείκτη μέχρι του αλφαριθμητικού (ή χαρακτήρα) που περιγράφει το αντίστοιχο πεδίο τιμή. Μετακίνησε το δείκτη στο τέλος του πεδίου τιμή.

Πίνακας 5.2 *Οι διαθέσιμοι τύποι των παραμέτρων και οι αντίστοιχες ενέργειες τους.*

ΤΥΠΟΣ	**ΕΝΕΡΓΕΙΑ**
c	Μετακίνηση του δείκτη στην πρώτη εμφάνιση του χαρακτήρα
s	Μετακίνηση του δείκτη στην πρώτη εμφάνιση του αλφαριθμητικού
cU	Μετακίνηση του δείκτη στην πρώτη εμφάνιση του χαρακτήρα ανεξαρτήτως μορφής
sU	Μετακίνηση του δείκτη στην πρώτη εμφάνιση του αλφαριθμητικού ανεξαρτήτως μορφής
ec	Ταίριαξε επακριβώς τον επόμενο χαρακτήρα. Μετακίνηση του δείκτη στο τέλος του χαρακτήρα
es	Ταίριαξε επακριβώς το επόμενο αλφαριθμητικό. Μετακίνηση του δείκτη στο τέλος του αλφαριθμητικού
ecU	Ταίριαξε επακριβώς τον επόμενο χαρακτήρα ανεξαρτήτως μορφής. Μετακίνηση του δείκτη στο τέλος του χαρακτήρα
esU	Ταίριαξε επακριβώς το επόμενο αλφαριθμητικό ανεξαρτήτως μορφής. Μετακίνηση του δείκτη στο τέλος του αλφαριθμητικού

sum (summary) : Η περίληψη μιας εγγραφής είναι όλοι οι χαρακτήρες από το σημείο του δείκτη μέχρι του αλφαριθμητικού (ή χαρακτήρα) που περιγράφει το αντίστοιχο πεδίο τιμή. Μετακίνησε το δείκτη στο τέλος του πεδίου τιμή.

end (end) : Ταίριαξε το αλφαριθμητικό (ή χαρακτήρα) που περιγράφει το αντίστοιχο πεδίο τιμή. Επιπλέον χρησιμοποιείται για να δηλώσει ότι ολοκληρώθηκε η διαδικασία για την συμπλήρωση των πεδιών μιας εγγραφής και ότι ο αναλυτής θα πρέπει να συνεχίσει με την παράμετρο str.

brk : (break) Χρησιμοποιείται για να δηλώσει ότι ολοκληρώθηκε η διαδικασία για την συμπλήρωση των πεδίων μιας εγγραφής και ότι ο αναλυτής θα πρέπει να συνεχίσει με την παράμετρο str.

Αφού έχουμε ορίσει τις παραμέτρους και έχουμε θέσει τις τιμές τους, οι τύποι καθώς

και οι αντίστοιχες ενέργειες τους περιγράφονται στον πίνακα 5.2.

Οι τύποι χωρίζονται σε δύο κατηγορίες, τους χαρακτήρες και τα αλφαριθμητικά. Επίσης διαφοροποιούνται ανάλογα με την μορφοποίηση (κεφαλαία ή μικρά) του συγκεκριμένου τύπου. Έχοντας ορίσει το πρότυπο ενός *Αρχείου Ανάλυσης* μπορούμε να μετακινήσουμε τον εικονικό δείκτη σε συγκεκριμένες θέσεις στη σελίδα των αποτελεσμάτων και κατ' επέκταση να συλλέξουμε όλες τις εγγραφές.

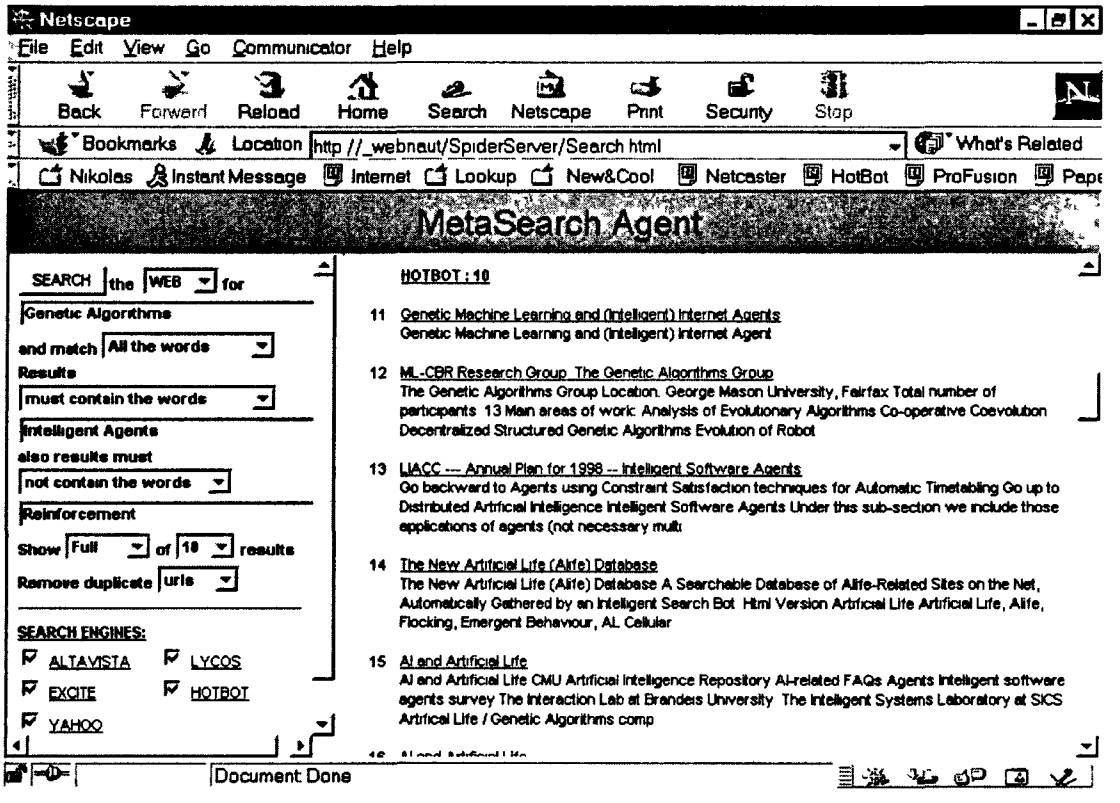
Η κλάση *SpiderClient* [B8] υποδέχεται και εξυπηρετεί όλες τις αιτήσεις, είτε ο πελάτης αλληλεπιδρά μέσω της διεπαφής του ΠΜΑΠ - δες παρακάτω το σχήμα 5.14 - είτε είναι ένας άλλος πράκτορας του συστήματος. Η μόνη διαφορά τους είναι ότι μέσω της διεπάφής ο ΠΔΔ θα δημιουργήσει ένα αντικείμενο της κλάσης *SpiderServer,* το οποίο με την σειρά του θα δημιουργήσει ένα στιγμιότυπο της κλάσης *SpiderClient για* να χειριστεί την αίτηση σαν μια ανεξάρτητη διεργασία αφού υλοποιεί την διεπαφή *Runnable.* Από την άλλη πλευρά, όταν ένας πράκτορας θέλει να απευθύνει μια ερώτηση και στη συνέχεια να πάρει τα αποτελέσματα των μηχανών αναζήτησης, αφού δημιουργήσει ένα αντικείμενο της κλάσης *SpiderServer* στη συνέχεια αντλεί τα αποτελέσματα μέσω της μεθόδου *getResults* [B8]. Η εν λόγω μέθοδος θα δημιουργήσει τόσα αντικείμενα της κλάσης *SpiderParser* [B9] όσες είναι και οι μηχανές αναζήτησης.

Το κάθε αντικείμενο της κλάσης *SpiderParser* θα διαμορφώσει την ερώτηση του πελάτη σύμφωνα με τα πρότυπα μιας συγκεκριμένης μηχανής αναζήτησης, θα απευθύνει την ερώτηση και εν συνεχεία θα αναλύσει τη σελίδα των αποτελεσμάτων. Το κάθε αντικείμενο της κλάσης *SpiderParser* γνωρίζει την σύνταξη των *Αρχείων Διαμόρφωσης και των Αρχείων Ανάλυσης* αφού κληρονομεί τις μεθόδους της κλάσης *SpiderEngines.*

Η μέθοδος *getResults* [B8] επιστρέφει όλα τα αποτελέσματα σε ένα *Hashtable,* όπου σαν κλειδιά έχει τα ονόματα των μηχανών αναζήτησης που χρησιμοποιήθηκαν κατά την ερώτηση και σαν τιμές έχει λίστες, με την μορφή *Vector,* όπου τις αντίστοιχες θέσεις της λίστας καταλαμβάνουν αντικείμενα της κλάσης *Spiderlink,* τα οποία με την σειρά τους, περιγράφουν μία εγγραφή στην σελίδα των αποτελεσμάτων της αντίστοιχης μηχανής αναζήτησης. Η πρόσβαση στην πληροφορία μίας εγγραφής γίνεται μέσω των δημοσίων μεθόδων *getHLink, getDescription, getSummary,* της κλάσης *Spiderlink.*

5.7.3 Σχεδίαση Διεπαφή του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών

Η διεπαφή που έχει στην διάθεση του, ο χρήστης του συστήματος Webnaut, για να σχηματίσει μια ερώτηση και στην συνέχεια να την απευθύνει σε μερικές ή όλες τις μηχανές αναζήτησης που υποστηρίζει το πρόγραμμα, εμφανίζεται στο σχήμα 5.14. Η αλληλεπίδραση του χρήστη με το ΠΜΑΠ γίνεται με την βοήθεια των πλαισίων της γλώσσας HTML [Γ12.α].



Σχήμα 5.14 Η διεπαφή του ΠΜΑΠ

Η σελίδα που εμφανίζεται στο πρόγραμμα πλοήγησης του χρήστη είναι χωρισμένη σε τρία πλαίσια. Το κεντρικό πλαίσιο βρίσκεται στο επάνω μέρος της οθόνης και εκτείνεται από το ένα έως το άλλο άκρο της οθόνης. Μέσα σε αυτό το πλαίσιο τοποθετούμε τους τίτλους της διεπαφής. Ακριβώς κάτω από το κεντρικό πλαίσιο είναι τοποθετημένα δύο πλαίσια. Στο αριστερό πλαίσιο ο χρήστης περιγράφει το θέμα για το οποίο ενδιαφέρεται να βρει σχετικές πληροφορίες ενώ στο δεξιό πλαίσιο εμφανίζονται τα αποτελέσματα της έρευνας.

Για το σχηματισμό της ερώτησης, ο χρήστης έχει στην διάθεση του τρία κουτιά κειμένου, μέσα στο οποία μπορεί να ορίσει όλες τις λέξεις-κλειδιά που περιγράφουν το θέμα της έρευνας του. Το κάθε κουτί κειμένου σχετίζεται με ένα κουτί επιλογών μέσω του οποίου ορίζουμε τους λογικούς τελεστές που συνδέουν τις αντίστοιχες

λέξεις. Για παράδειγμα, αν η επιλογή είναι "*όλες οι λέξεις*" τότε ο λογικός τελεστής AND παρεμβάλλεται ανάμεσα στις λέξεις κατά την φάση του σχηματισμού της ερώτησης. Ενώ, αν η επιλογή ήταν "*οποιαδήποτε από τις λέξεις*" τότε ο λογικός τελεστής OR θα παρεμβαλλόταν ανάμεσα στις λέξεις.

Ο χρήστης έχει την δυνατότητα να ορίσει και λέξεις ή φράσεις οι οποίες δεν θα ήθελε να εμφανίζονται στα αποτελέσματα της έρευνας. Κατά αυτό το τρόπο ο χρήστης μπορεί να περιγράψει κατά το καλύτερο τρόπο το θέμα της έρευνας του, δημιουργώντας σύνθετες ερωτήσεις. Τέλος, υπάρχει η δυνατότητα μορφοποίησης των αποτελεσμάτων της έρευνας, ο περιορισμός των αποτελεσμάτων που θα εμφανίζονται, η απομάκρυνση των διπλών εγγραφών, ο καθορισμός των μηχανών αναζήτησης που θα συμμετέχουν στην έρευνα.

5.7.4 Ερευνητικές Προεκτάσεις και Εφαρμογές της Λειτουργίας του Πράκτορα Μεταμηχανή Αναζήτησης Πληροφοριών

Η λειτουργία του ΠΜΑΠ δεν διαφέρει από άλλες αντίστοιχες εφαρμογές στο διαδίκτυο. Εκείνο που πιστεύουμε όμως ότι είναι αρκετά ενδιαφέρον είναι ότι η όλη λειτουργία της εφαρμογής ελέγχεται μέσω των δύο αρχείων. Έτσι λοιπόν η προσθήκη μιας νέας μηχανής αναζήτησης γίνεται αυτόματα με την προσθήκη των δύο αυτών αρχείων. Οπότε είναι αρκετά εύκολο για ένα διαχειριστή ενός δικτυακού τόπου να προσθέσει νέες δυνατότητες στην ήδη υπάρχουσα μεταμηχανή αναζήτησης.

Το σχήμα περιγραφής των *Αρχείων Διαμόρφωσης* είναι αρκετά ευέλικτο ώστε να μπορούμε να περιγράψουμε και νέες υπηρεσίες που παρέχονται από μια μηχανή αναζήτησης, αρκεί φυσικά να υπάρχει μια διεπαφή μέσω διαδικτύου για να μπορεί ο χρήστης να απευθύνει ερωτήσεις. Το σχήμα περιγραφής των *Αρχείων Ανάλυσης* είναι επίσης αρκετά ευέλικτο και αυτό φαίνεται από το ότι, ενώ ο αρχικός σχεδιασμός τους ήταν για την διαχείριση σελίδων με μορφοποίηση σε γλώσσα HTML μπορεί άνετα να επεκταθεί και σε άλλες πηγές πληροφόρησης (π.χ. Βάσεις Δεδομένων - Βιβλιογραφικούς καταλόγους κ.λπ.), αρκεί να χρησιμοποιείται η ίδια επανάληψη στον τρόπο παρουσίασης των αποτελεσμάτων και επίσης να υπάρχουν αναγνωριστικά σημεία που διαχωρίζουν την μία εγγραφή από την άλλη.

Επίσης ένα άλλο σημείο που έχει αρκετό ενδιαφέρον είναι ότι η όλη μεθοδολογία μπορεί να χρησιμοποιηθεί από τις ήδη υπάρχουσες μηχανές αναζήτησης ώστε να

ενημερώνουν ένα πρόγραμμα πλοήγησης για το διαδίκτυο ή έναν πράκτορα για τις δυνατότητες που παρέχουν και φυσικά τον τρόπο με τον οποίο μπορούν να αντλήσουν τα αποτελέσματα. Πιο συγκεκριμένα, θα μπορούσαν τα προγράμματα πλοήγησης να παρέχουν την λειτουργία που παρέχει η συγκεκριμένη εφαρμογή, δηλαδή να έχουν μια διεπαφή παρόμοια με αυτή που εμφανίζεται στο σχήμα 5.14. Από την πλευρά τους οι μηχανές αναζήτησης θα μπορούσαν να προσφέρουν τα δύο αρχεία που θα πρέπει να χρησιμοποιήσει το πρόγραμμα πλοήγησης για να συντάξει και να απευθύνει ερωτήσεις και στην συνέχεια να κάνει εξαγωγή των αποτελεσμάτων. Αυτό θα μπορούσε να υλοποιηθεί αρκετά εύκολα, αν για παράδειγμα, οι μηχανές αναζήτησης προσθέσουν μερικές γραμμές στο τίτλον της σελίδας που παρέχει την διεπαφή αναζήτησης, όπως εμφανίζονται παρακάτω :

```
<meta name="SEARCH ENGINE" content="true">
<meta name="OPTIONS" content="http://site.searchengine.com/.../options.txt">
<meta name="PARSER"  content="http://site.searchengine.com/.../parser.txt">
```

Όσο για τα δύο αρχεία, θα μπορούσαν να ακολουθούν την μορφοποίηση που χρησιμοποιούμε είτε θα μπορούσαν να κωδικοποιηθούν σε xml μορφή όπως εμφανίζονται παρακάτω :

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<ENGINE>
        <WEB>
                <NAME> ALTAVISTA </NAME>
                <HOST> www.altavista.com</HOST>
                ....
        </WEB>
        .....
</ENGINE>
```

Τα πλεονεκτήματα μιας τέτοιας προσέγγισης είναι πάρα πολλά αφού οι χρήστες θα χρησιμοποιήσουν πάντα την ίδια διεπαφή - αυτή που θα παρέχει το πρόγραμμα πλοήγησης - για να απευθύνουν τις ερωτήσεις τους αφού δεν θα χρειάζεται πλέον να μάθουν κανένα άλλο συνδυασμό λέξεων κλειδιών με λογικούς τελεστές ή να μάθουν την σύνταξη δύσχρηστων εντολών για να σχηματίσουν τις ερωτήσεις τους.

Από την άλλη πλευρά, και οι πράκτορες λογισμικού θα επωφεληθούν από μία τέτοια προσέγγιση αφού θα χρειάζεται να γνωρίζουν μόνο την ηλεκτρονική διεύθυνση της μηχανής αναζήτησης. Στην συνέχεια ζητώντας τα δύο αρχεία θα είναι σε θέση να απευθύνουν ερωτήσεις και να αντλήσουν πληροφορίες από οπουδήποτε

στο διαδίκτυο. Έτσι θα μπορούν οι πράκτορες λογισμικού να ανταλλάσσουν πληροφορίες μεταξύ τους και να ενημερώνουν ο ένας τον άλλον για νέες μηχανές αναζήτησης.

Βέβαια σχεδόν όπως σε κάθε εφαρμογή έτσι και σε αυτήν την πρώτη έκδοση της εφαρμογής δεν θα μπορούσαν να λείπουν τα προβλήματα ή τα μειονεκτήματα μερικά εκ των οποίων τα παραθέτουμε παρακάτω :

α) Δεν υπάρχει δυνατότητα να ζητήσουμε από μία μηχανή αναζήτησης την επόμενη σελίδα των αποτελεσμάτων. Οι μηχανές αναζήτησης χρησιμοποιούν τη δική τους μεθοδολογία για να βαθμολογήσουν τις σελίδες που είναι αποθηκευμένες στην Βάση Δεδομένων τους. Έτσι πολλές φορές η εγγραφή που ενδιαφέρει το χρήστη βρίσκεται στην δεύτερη ή τρίτη σελίδα των αποτελεσμάτων.

Η πρακτική του να επιδεικνύονται συνολικά ή τμηματικά μόνο τα αποτελέσματα της πρώτης σελίδας χρησιμοποιείται από αρκετές μεταμηχανές αναζήτησης. Δηλαδή συλλέγουν όσα περισσότερες εγγραφές μπορούν να αντλήσουν από τις μηχανές αναζήτησης και στην συνέχεια τα επιδεικνύουν τμηματικά σε σελίδες. Βέβαια η δυνατότητα μιας μεταμηχανής αναζήτησης να αντλήσει διαδοχικά αποτελέσματα αυξάνει και την ποιότητα της αλλά και την χρησιμότητα της. Στις μελλοντικές εργασίες ανήκει και η επέκταση του πρωτοκόλλου ώστε να μπορεί η εφαρμογή να ζητήσει τις επόμενες σελίδες των αποτελεσμάτων.

β) Δεν υπάρχει η δυνατότητα να περιγράψουμε όλες τις δυνατότητες που προσφέρουν οι μηχανές αναζήτησης. Για παράδειγμα αρκετές μηχανές αναζήτησης μπορούν να αντλήσουν αποτελέσματα από συγκεκριμένους δικτυακούς τόπους (π.χ. edu, com), χώρες (π.χ. GREECE), αρχεία (π.χ. gif, MP3) κ.λπ. Στην παρούσα φάση μελετάμε τον τρόπο με τον οποίο θα μπορούσαμε να ενσωματώσουμε στην επόμενη έκδοση μερικά αν όχι όλα από τα ποίο δημοφιλή χαρακτηριστικά που προσφέρουν οι μεγαλύτερες από τις μηχανές αναζήτησης.

γ) Χρειάζεται ένας έμπειρος χρήστης και γνώστης του HTTP πρωτοκόλλου και της HTML γλώσσας για να περιγράψει τα δύο αρχεία αφού πρώτα μελετήσει την διεπαφή καθώς επίσης και την σελίδα των αποτελεσμάτων των μηχανών αναζήτησης. Βέβαια θα μπορούσαν όλες οι εφαρμογές που είναι εγκατεστημένες στο διαδίκτυο να

ενημερώνονται από ένα κεντρικό διακομιστή ώστε να μπορεί να γίνει εύκολα η ενημέρωση και η εγκατάσταση των δύο αρχείων. Η κατασκευή ενός τέτοιου δικτυακού κόμβου όπου οι χρήστες - διαχειριστές ή πράκτορες λογισμικού - μπορούν να ενημερώνονται και να *κατεβάζουν* τις νέες εκδόσεις των *Αρχείων Διαμόρφωσης* και *Ανάλυσης* είναι επίσης ένα από τα αντικείμενα των μελλοντικών μας εργασιών.

δ) Δεν υπάρχει ένας και μοναδικός τρόπος για την κατασκευή του *Αρχείου Ανάλυσης*. Πρακτικά η περιγραφή της εγγραφής μπορεί να γίνει με περισσότερες ή με λιγότερες εντολές και αυτό εξαρτάται από την συγκεκριμένη μηχανή αναζήτησης δηλαδή την μορφοποίηση των αποτελεσμάτων αλλά και από την εμπειρία του συγγραφέα του συγκεκριμένου αρχείου. Η κατασκευή ενός εργαλείου που θα προσφέρει την δυνατότητα της εποπτικής κατασκευής είναι επίσης μια από τις προγραμματισμένες μελλοντικές εργασίες μας.

Τέλος θα πρέπει να τονίσουμε ότι το αποτέλεσμα της λειτουργίας του ΠΜΑΠ είναι αρκετά ικανοποιητικό αφού υπάρχει η δυνατότητα να αντλήσουμε αρκετές πληροφορίες από τις δημοφιλέστερες μηχανές αναζήτησης και να προτείνουμε στο χρήστη νέες δικτυακές τοποθεσίες, οι οποίες θα είναι σύμφωνες με τα ενδιαφέροντα του.

5.8 Σχεδίαση του Πράκτορα Πύλη Εικονικής Πραγματικότητας

Ένας από τους στόχους της τεχνητής νοημοσύνης είναι και η ανάπτυξη φιλικών και εύχρηστων διεπαφών, οι οποίες θα μας επιτρέπουν να αλληλεπιδρούμε με φυσικό και φιλικό τρόπο με τα διάφορα υπολογιστικά συστήματα. Ειδικότερα, την τελευταία δεκαετία, λόγω της ανάπτυξης του διαδικτύου αλλά και του εμπορικού ενδιαφέροντος που παρουσιάζει το όλο εγχείρημα, έχει σημειωθεί σημαντική πρόοδος στην ανάπτυξη και δημιουργία χώρων Εικονικής Πραγματικότητας - ΕΠ, που ο χρήστης προσεγγίζει μέσω του διαδικτύου [Car93, Mac97, Bra98, Vie99].

Ο όρος ΕΠ αναφέρετε στην εξερεύνηση, σε πραγματικό χρόνο, χώρων φανταστικών ή πραγματικών που έχουν δημιουργηθεί με την βοήθεια του υπολογιστή. Η εξερεύνηση περιλαμβάνει και την αλληλεπίδραση του χρήστη με τα αντικείμενα του εικονικού χώρου καθώς και την επικοινωνία του με τους χαρακτήρες, που δραστηριοποιούνται σε αυτό το περιβάλλον. Συνήθως για την απεικόνιση του

εικονικού χώρου γίνεται χρήση κειμένου [Chu99, Bel00], πολυμέσων [Cze99, Carr98] καθώς και γραφικών στις δύο ή τρεις διαστάσεις. Επίσης είναι αρκετά διαδεδομένη η χρήση ειδικών συσκευών για την αλληλεπίδραση και την διαχείριση των αντικειμένων [Bow01].

Τα οφέλη από την ανάπτυξη αυτών των εφαρμογών είναι πολλαπλά, όπως για παράδειγμα, α) οπτικοποιούνται σύνθετες εργασίες ή δεδομένα με τρόπο κατανοητό από τους ανθρώπους, β) σε εκπαιδευτικά περιβάλλοντα, οι χρήστες μπορούν να μελετήσουν κάθε χώρο και διαδικασία στο περιβάλλον μέσα από πραγματικά σενάρια, ενώ μέσω της επανάληψης αυξάνονται οι δεξιότητες των εκπαιδευομένων και εξαλείφονται οι όποιες δυσκολίες υπάρχουν στην χρήση ενός υπολογιστικού συστήματος ή διαδικασίας, γ) μειώνεται σημαντικά το κόστος εκπαίδευσης, δ) υπάρχει η δυνατότητα αυτοδιδασκαλίας και ξενάγησης, ε) υπάρχει η δυνατότητα πρόσβασης σε πολλούς και απομακρυσμένους χρήστες την ίδια χρονική στιγμή.

Σε έναν εικονικό κόσμο, η διαχείριση των αντικειμένων, η επικοινωνία, η συμπεριφορά και η αλληλεπίδραση με τους χαρακτήρες - πρόσωπα, είναι ένα αρκετά σημαντικό και ενδιαφέρον πεδίο έρευνας. Ο στόχος είναι να δημιουργηθεί η αίσθηση στον χρήστη ότι επικοινωνεί και συμπεριφέρεται όπως και στο φυσικό περιβάλλον. Για την ανάπτυξη αυτών των διεπαφών, η πιο συνηθισμένη προσέγγιση, είναι η δημιουργία χαρακτήρων που δανείζονται στοιχεία είτε ανθρωπόμορφα είτε από το φυσικό περιβάλλον [Bal97, Joh98, Pan99, Pand01].

Οι χρήστες των τρισδιάστατων τεχνητών χώρων, χρησιμοποιούν για την πλοήγηση τους, αλλά και για τον καλύτερο έλεγχο των κινήσεων τους κατά την αλληλεπίδραση τους με τα αντικείμενα, ειδικές συσκευές - χειριστήρια, όπως για παράδειγμα ειδικά κράνη (head-mounted displays), γάντια (data gloves) κ.λπ., [Bow01, Coo97]. Όμως, για τους περισσότερους ανθρώπους η επικοινωνία μέσω φυσικής γλώσσας θεωρείται ο πιο εύκολος και φυσικός τρόπος για την αλληλεπίδραση τους, είτε με πρόσωπα είτε με αντικείμενα σε εικονικούς χώρους. Ο λόγος είναι ότι οι άνθρωποι δεν γνωρίζουν ή δεν αισθάνονται άνετα με τον χειρισμό των ειδικών συσκευών ή με τη χρήση ειδικών κανόνων επικοινωνίας [God99, Sch00, Jon95].

Όπως δείχνουν τα αποτελέσματα σχετικών ερευνών [Nas94, Les97, Sta98, Joh00], όταν οι χαρακτήρες ή το περιβάλλον δράσης, διαθέτουν στοιχεία που είναι οικεία στους χρήστες, τότε και οι χρήστες αντιδρούν διαφορετικά, και επιδεικνύουν πιο "*ανθρώπινη*" συμπεριφορά κατά την αλληλεπίδραση τους με τον υπολογιστή. Μια

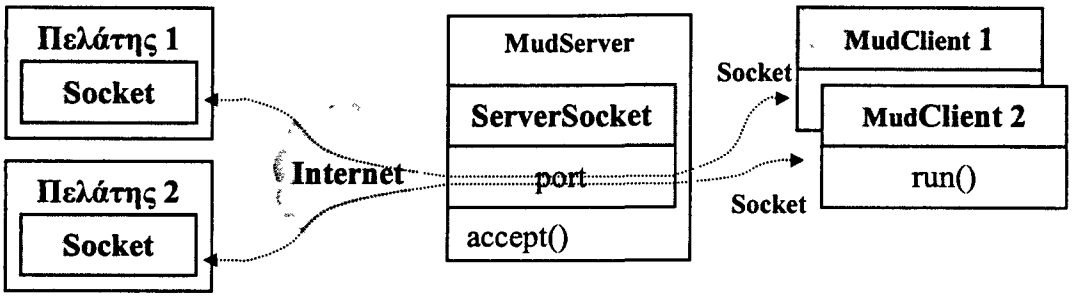
τέτοια συμπεριφορά, είναι επιθυμητή σε πολλές εφαρμογές και ιδιαίτερα στον χώρο των πρακτόρων λογισμικού, όπου ο χρήστης θα πρέπει να αντιμετωπίσει την εφαρμογή σαν συνεργάτη και όχι σαν εργαλείο.

Ο Πράκτορας Πύλη Εικονικής Πραγματικότητας - ΠΠΕΠ, είναι ένας πράκτορας λογισμικού, ο οποίος δίνει πρόσβαση στο χρήστη του συστήματος Webnaut σε χώρους συνδιασκέψεων και τεχνητά περιβάλλοντα. Ο σκοπός μας δεν είναι να εστιάσουμε στην κίνηση και την απεικόνιση των χαρακτήρων με τρισδιάστατα γραφικά, αλλά στο μοντέλο επικοινωνίας και εκμάθησης των ενδιαφερόντων ενός χρήστη, που χρησιμοποιεί την φυσική γλώσσα κατά την διάρκεια της πλοήγησης του, σε εικονικούς χώρους.

5.8.1 Λεπτομερής Σχεδιασμός του Εξυπηρετητή της Υπηρεσίας Εικονικής Πραγματικότητας

Για την επίδειξη της λειτουργίας αλλά και του τρόπου επικοινωνίας του ΠΠΕΠ με τους διαδικτυακούς εξυπηρετητές που θα προσφέρουν υπηρεσίες ΕΠ ή συνδιάσκεψης έχουμε σχεδιάσει έναν εξυπηρετητή ο οποίος προσφέρει στους πελάτες του, ένα περιβάλλον εικονικής πραγματικότητας. Ο εξυπηρετητής αποτελείται από δύο κλάσεις : α) την κλάση *Mudserver* [B37], β) την κλάση *MudClient* [B38], και λειτουργεί αυτόνομα σε έναν οποιοδήποτε δικτυακό κόμβο. Η κλάση *Mudserver* υλοποιεί την βασική δομή του εξυπηρετητή, ο οποίος καταλαμβάνει μια δικτυακή πόρτα για την εξυπηρέτηση των πελατών, και εν συνεχεία δημιουργεί αντικείμενα της κλάσης *MudClient,* για την εξυπηρέτηση των πελατών. Το σχήμα 5.15 παρουσιάζει το πλαίσιο λειτουργίας του *Mudserver.*

Όλες οι παράμετροι λειτουργίας του *Mudserver* αρχικοποιούνται μέσω της μεθόδου *ReadServerProperties* και εν συνεχεία αποθηκεύονται στο *Hashtable ServerProp.* Το αποτέλεσμα της μεθόδου *ReadServerProperties,* χρησιμοποιείται στο σώμα της μεθόδου *StartServer,* η οποία είναι υπεύθυνη για την αρχικοποίηση του εξυπηρετητή σε δύο στάδια, πρώτον με την δημιουργία ενός *ServerSocket* [A1] αντικειμένου το οποίο καταλαμβάνει μια δικτυακή πόρτα για την εξυπηρέτηση των πελατών, και δεύτερον χρησιμοποιώντας την μέθοδο *accept,* της κλάσης *ServerSocket,* για την αναμονή των επερχομένων συνδέσεων.



Σχήμα 5.15 Το πλαίσιο λειτουργίας του εξυπηρετητή στο κόμβο ΕΠ.

Η μέθοδος *accept* επιστρέφει ένα στιγμιότυπο της κλάσης *Socket* [A1] το οποίο αποτελεί και την σύνδεση του εξυπηρετητή με τον πελάτη. Το στιγμιότυπο αυτό χρησιμοποιείται σαν όρισμα για την δημιουργία ενός αντικειμένου της κλάσης *MudClient,* που εκτελείται σαν ανεξάρτητη διεργασία - μέσω της υλοποίησης της διεπαφής *Runnable* [A2] - για την εξυπηρέτηση της αίτησης του πελάτη.

5.8.2 Λειτουργικότητα του Εξυπηρετητή της Υπηρεσίας Εικονικής Πραγματικότητας

Η ανεξάρτητη διεργασία *MudClient,* η οποία εξυπηρετεί τους πελάτες, όπως φαίνεται και στο σχήμα 5.16, εκτελεί μια σειρά από ενέργειες, όπως :

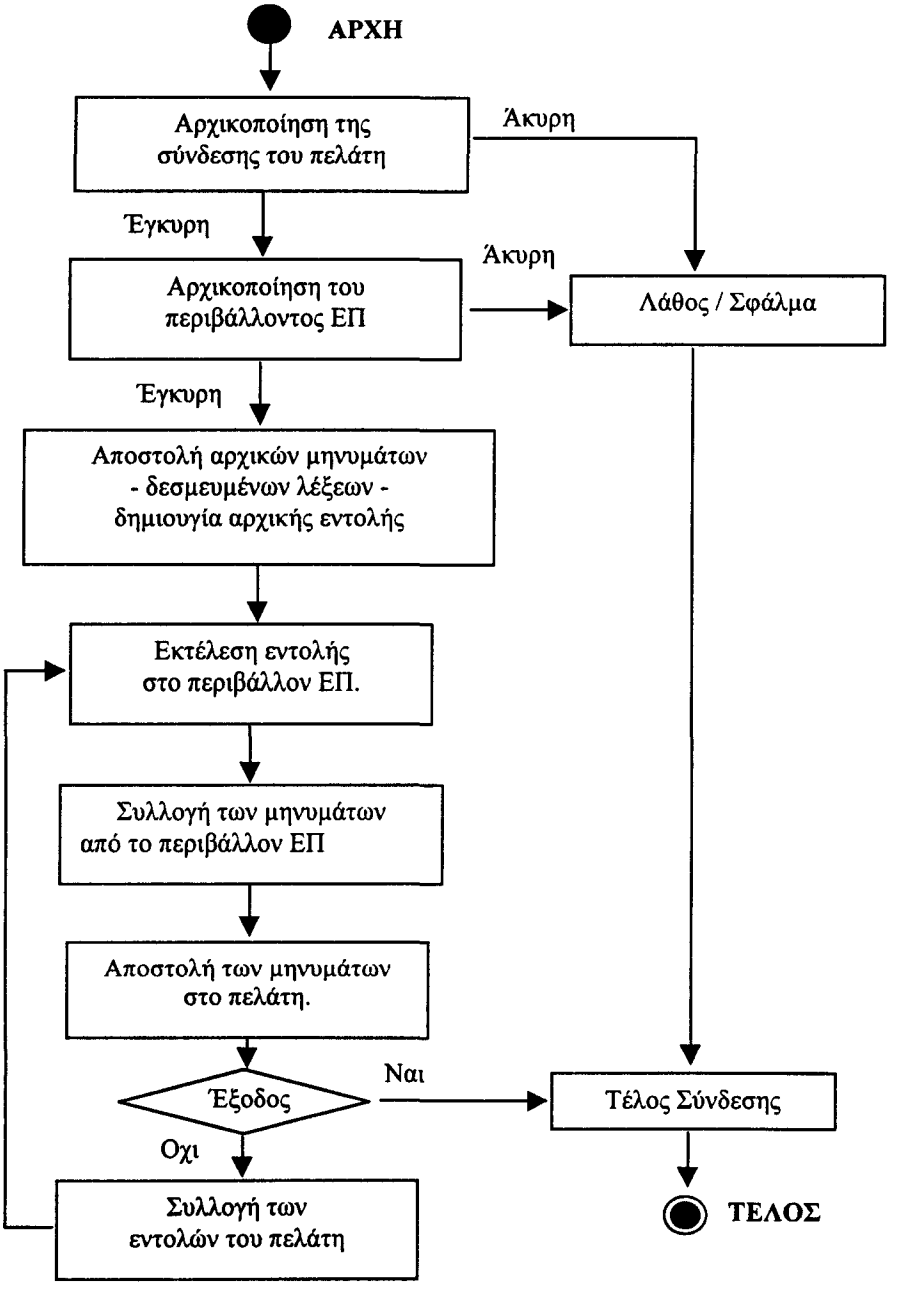
α) Αρχικοποιεί τα κανάλια αποστολής και λήψης δεδομένων για την επικοινωνία με τον πελάτη.

β) Αρχικοποιεί το περιβάλλον ΕΠ. Και στις δύο προηγούμενες ενέργειες, αν υπάρχει κάποιο σφάλμα, τότε ενημερώνεται ο πελάτης με το ανάλογο μήνυμα και φυσικά τερματίζεται η σύνδεση.

γ) Στην περίπτωση που δεν υπάρχει κάποιο σφάλμα, τότε γίνεται αποστολή δύο μηνυμάτων, από τον εξυπηρετητή προς τον πελάτη. Γενικά, οι πληροφορίες που αποστέλλονται προς τον πελάτη αφορούν, είτε μηνύματα από το περιβάλλον της ΕΠ τα οποία δημιουργούνται σαν συνέπεια των εντολών του χρήστη, είτε πληροφορίες του εξυπηρετητή προς το πρόγραμμα που χρησιμοποιεί ο πελάτης, για την αλληλεπίδραση του με τον κόσμο της ΕΠ.

Ο πελάτης κατά την ανάγνωση της απόκρισης του εξυπηρετητή, θεωρεί ότι όλα

τα μηνύματα ανήκουν στην πρώτη κατηγορία, δηλαδή προέρχονται από το περιβάλλον της ΕΠ, εκτός αν περικλείονται από τις ετικέτες <SERVER> και </SERVER>, όπως για παράδειγμα <SERVER:ΕΝΤΟΛΗ>ΠΛΗΡΟΦΟΡΙΕΣ ΕΝΤΟΛΗΣ</SERVER>. Στη δεύτερη περίπτωση, τα μηνύματα δεν επιδεικνύονται στον χρήστη, αλλά χρησιμοποιούνται εσωτερικά από το πρόγραμμα που χρησιμοποιεί ο χρήστης για την αλληλεπίδραση του με το κόμβο ΕΠ.



Σχήμα 5.16 *Το διάγραμμα δραστηριοτήτων του εξυπηρετητή στον κόμβο ΕΠ.*

Κατ' αυτόν τον τρόπο, δημιουργούμε ένα ανοικτό πρωτόκολλο επικοινωνίας, αφού υπάρχει η δυνατότητα μίξης μηνυμάτων από τον κόσμο ΕΠ και τον εξυπηρετητή.

Μετά την αποστολή των δύο μηνυμάτων - δες παρακάτω - είναι όλα έτοιμα για την επικοινωνία με τον χρήστη, γι' αυτόν τον λόγο ο εξυπηρετητής στέλνει μια εντολή αρχικοποίησης στο περιβάλλον ΕΠ.

δ) Η εκτέλεση μιας εντολής, δημιουργεί αλλαγές, οι οποίες εκφράζονται με νέα μηνύματα από το περιβάλλον ΕΠ προς το χρήστη. Όλες οι εντολές που θα εκτελεστούν, προέρχονται από τον χρήστη, εκτός από την εντολή αρχικοποίησης, η οποία προέρχεται από τον εξυπηρετητή και σκοπό έχει να ενημερώσει το περιβάλλον ΕΠ, ότι όλα είναι έτοιμα για να αρχίσει η επικοινωνία με το χρήστη. Το αποτέλεσμα της εντολής αρχικοποίησης, είναι κάποια μηνύματα εισαγωγικά προς τον χρήστη τα οποία τον καλωσορίζουν στο συγκεκριμένο χώρο.

ε) Όλα τα μηνύματα συλλέγονται από τον εξυπηρετητή και αποστέλλονται μέσω του καναλιού εξόδου, που χρησιμοποιείται για την επικοινωνία με τον χρήστη.

ζ) Ο εξυπηρετητής ελέγχει αν πρέπει να συνεχίσει την επικοινωνία με τον πελάτη, και αν ναι, τότε περιμένει τις εντολές του χρήστη και συνεχίζει την λειτουργία στο βήμα δ, διαφορετικά τερματίζει τη σύνδεση.

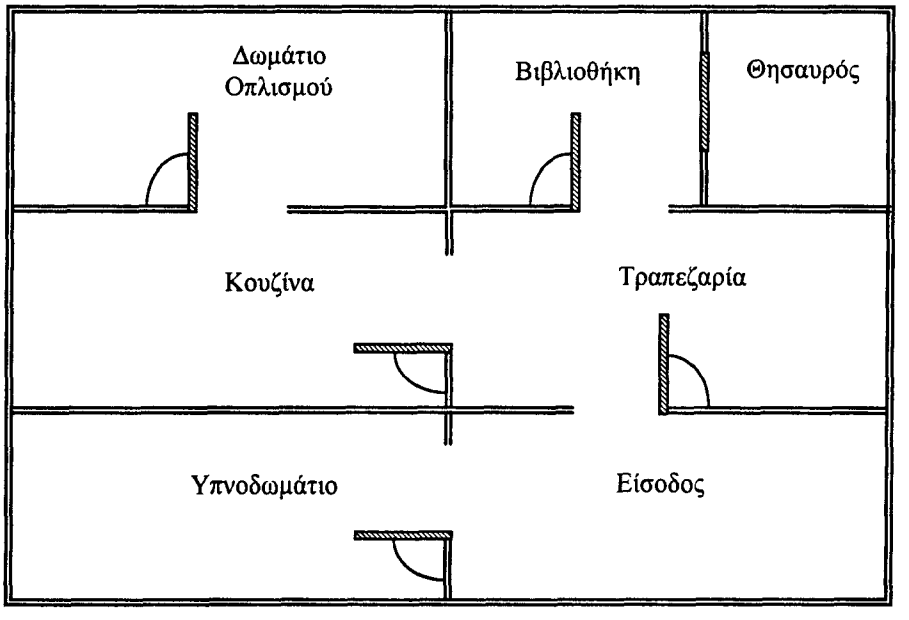
Ο ίδιος κύκλος, συνεχίζεται μέχρις ότου ο χρήστης να διακόψει την επικοινωνία του, είτε μένοντας ανενεργός για ένα προκαθορισμένο χρονικό διάστημα, ή μέχρι να τερματίσει ο εικονικός κόσμος την σύνδεση του χρήστη. Η τελευταία επιλογή είναι αναγκαία σε περιβάλλοντα διασκέδασης ή συνδιασκέψεων, όπου το ίδιο το περιβάλλον θα πρέπει να ενημερώσει τον εξυπηρετητή ότι θα πρέπει να διακόψει την σύνδεση με τον πελάτη.

5.8.3 Υλοποίηση του Κόσμου Εικονικής Πραγματικότητας

Στα πλαίσια της διατριβής, για την επίδειξη της λειτουργίας του ΠΠΕΠ, έχουμε δημιουργήσει και ένα εικονικό περιβάλλον περιπέτειας-διασκέδασης. Ο χρήστης μεταφέρεται σε ένα μεσαιωνικό κάστρο, με σκοπό να πάρει το χρυσάφι που βρίσκεται στο θησαυροφυλάκιο. Όμως, για να επιτύχει τον στόχο του, θα πρέπει να διασχίσει τα δωμάτια του κάστρου και να συλλέξει διάφορα αντικείμενα, που θα τον

βοηθήσουν να αντιμετωπίσει τον πολεμιστή, που προστατεύει το θησαυρό. Κατά την διάρκεια του παιχνιδιού, ο χρήστης έχει την δυνατότητα να απευθύνει εντολές σε φυσική γλώσσα και να αλληλεπιδρά με τα αντικείμενα του χώρου.

Η δράση, όπως φαίνεται και στο σχήμα 5.17, εκτυλίσσεται σε έναν κλειστό χώρο που αποτελείται από 7 δωμάτια. Ο χρήστης, μετά την επιτυχημένη σύνδεση του, τοποθετείται στο χώρο της εισόδου και έχει την δυνατότητα να απευθύνει εντολές. Η σύνταξη των εντολών είναι αρκετά απλή, αφού γίνεται χρήση της φυσικής γλώσσας, και πάντα υπάρχει η δυνατότητα της βοήθειας. Για παράδειγμα, η εντολή ΚΟΙΤΑΞΕ ή ΚΟΙΤΑΞΕ ΓΥΡΩ, ενημερώνει το χρήστη για το δωμάτιο που βρίσκεται τώρα, καθώς και για τα αντικείμενα που βρίσκονται στον συγκεκριμένο χώρο. Η εντολή ΠΗΓΑΙΝΕ ΣΤΗΝ ΚΟΥΖΙΝΑ, θα μεταφέρει τον χρήστη στην κουζίνα, αν φυσικά το δωμάτιο που βρίσκεται την τρέχουσα στιγμή συνδέεται με την κουζίνα, και δεν είναι κλειδωμένη η πόρτα που συνδέει τα δύο δωμάτια.



Σχήμα 5.17 *Η δράση εκτυλίσσεται σε ένα εικονικό χώρο επτά δωματίων.*

Το σύνολο των εντολών είναι πλήρες, για τους σκοπούς του παιχνιδιού, αφού επιτρέπει στο χρήστη την εξερεύνηση του χώρου, την μετακίνηση του από δωμάτιο σε δωμάτιο, την συλλογή και απόρριψη αντικειμένων, την αλληλεπίδραση του με τα διάφορα αντικείμενα που βρίσκονται στον χώρο. Ο χρήστης θα πρέπει να εξερευνήσει τον χώρο και να επιλέξει τα αντικείμενα-εργαλεία, τα οποία θα τον βοηθήσουν να επιτύχει τον στόχο του. Η κωδικοποίηση του χώρου στον οποίο εκτυλίσσεται η

δράση, η πλοκή, η ανάλυση των προτάσεων καθώς και οι κανόνες που διέπουν την επικοινωνία σε φυσική γλώσσα μεταξύ του χρήστη με το ΕΠ, έχουν γίνει με την βοήθεια της γλώσσας Prolog - PROgramming in LOGic - [Brat90, Cov97, Sho94, Ste94, Γ58, Γ59].

Η Prolog, διαφέρει από τις άλλες γλώσσες προγραμματισμού στο ότι δεν κωδικοποιούμε την μεθοδολογία επίλυσης ενός προβλήματος, αλλά ένα πρόγραμμα το οποίο αποτελείται από μια βάση δεδομένων από γεγονότα, καθώς και από κανόνες οι οποίοι περιγράφουν τις συσχετίσεις που αληθεύουν στα πλαίσια της εφαρμογής. Κατ' αυτόν τον τρόπο, ένας χρήστης δεν τρέχει το πρόγραμμα, αλλά απευθύνει μια ερώτηση την οποία το σύστημα θα προσπαθήσει να απαντήσει ερευνώντας την βάση των γεγονότων και συμπεραίνοντας επαγωγικά βάσει των κανόνων.

Η επιλογή και χρήση της γλώσσας Prolog έγινε λόγω των μοναδικών χαρακτηριστικών που διαθέτει και την κάνουν να ξεχωρίζει από τις άλλες γλώσσες προγραμματισμού. Παράδειγμα, οι "*λογικές μεταβλητές*", οι οποίες λειτουργούν όπως οι μαθηματικές μεταβλητές και μας επιτρέπουν να απευθύνουμε ερωτήσεις ή να ορίσουμε γενικευμένα γεγονότα. Επίσης η Prolog έχει έναν ισχυρό μηχανισμό ταυτοποίησης, χρησιμοποιεί σαν στρατηγική έρευνας την οπισθοδρόμηση (backtracking), με σκοπό να ερευνήσει για αποδείξεις. Τα παραπάνω χαρακτηριστικά, κάνουν αρκετά εύκολη υπόθεση την συγγραφή γραμματικής, για ανάλυση προτάσεων σε φυσική γλώσσα [Cov94, Gal91, Gaz89, Γ59.a], κάτι που είναι ιδιαίτερα σημαντικό στη συγκεκριμένη εφαρμογή.

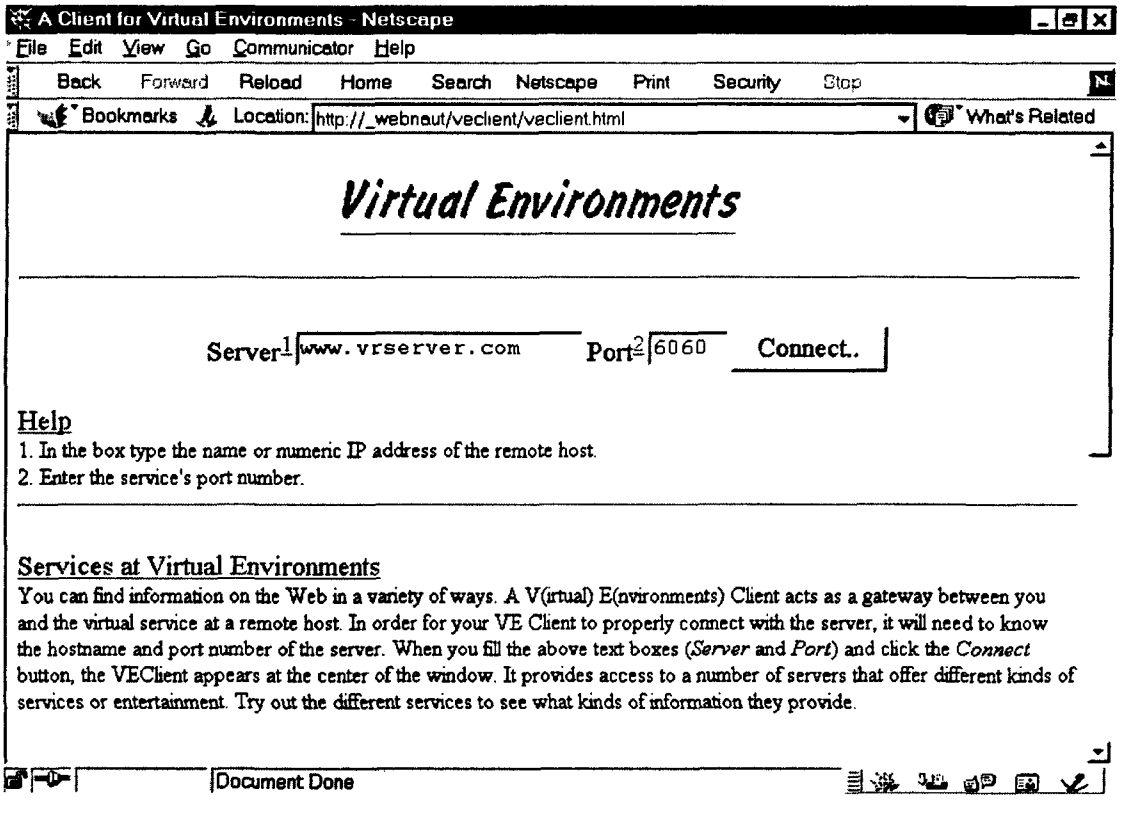
Για την διασύνδεση του εξυπηρετητή, που προσφέρει τις υπηρεσίες ΕΠ, με το πρόγραμμα Prolog, που υλοποιεί το περιβάλλον ΕΠ, έχουμε χρησιμοποιήσει το Logic Server της Amzi Prolog [Γ59.b]. Η συγκεκριμένη υλοποίηση μας προσφέρει δύο πλεονεκτήματα. Πρώτον, η μηχανή της Prolog είναι διαθέσιμη σε μορφή βιβλιοθήκης (dll), επιτρέποντας έτσι την εύκολη διασύνδεση της με άλλα προγραμματιστικά περιβάλλοντα, όπως για παράδειγμα της γλώσσας Java. Δεύτερον, επιτρέπει την πολυδιεργασία, δηλαδή να υπάρχουν ταυτόχρονα πολλά στιγμιότυπα της ιδίας μηχανής, κάτι που είναι επιθυμητό στο διαδίκτυο όταν ένας εξυπηρετητής θα πρέπει να προσφέρει ταυτόχρονα υπηρεσίες σε πολλούς πελάτες.

5.8.4 Η Διεπαφή του Πράκτορα Πύλη Εικονικής Πραγματικότητας

Η εμφάνιση της διεπαφής αλληλεπίδρασης του ΠΠΕΠ, για την σύνδεση ενός χρήστη

του συστήματος Webnaut με τον εξυπηρετητή που προσφέρει τις υπηρεσίες ΕΠ, μπορεί να γίνει είτε έμμεσα είτε άμεσα. Έμμεσα μπορεί να γίνει μέσω μιας HTML σελίδας, η οποία βρίσκεται αποθηκευμένη τοπικά στο ΠΕΔΚ. Στο σχήμα 5.18 απεικονίζεται η σελίδα ενεργοποίησης όπου ο χρήστης μπορεί να δηλώσει με ακρίβεια τα στοιχεία που απαιτούνται για την σύνδεση.

Άμεση σύνδεση μπορεί να γίνει κατά την διάρκεια της πλοήγησης ενός χρήστη στο διαδίκτυο. Πιο συγκεκριμένα, οι κόμβοι που θα προσφέρουν υπηρεσίες εικονικής πραγματικότητας, θα πρέπει να έχουν κατασκευάσει και έναν σύνδεσμο σε μια HTML σελίδα, όπου μέσω αυτού θα ενεργοποιείται η συγκεκριμένη υπηρεσία για τους χρήστες που κάνουν χρήση του προγράμματος Webnaut.



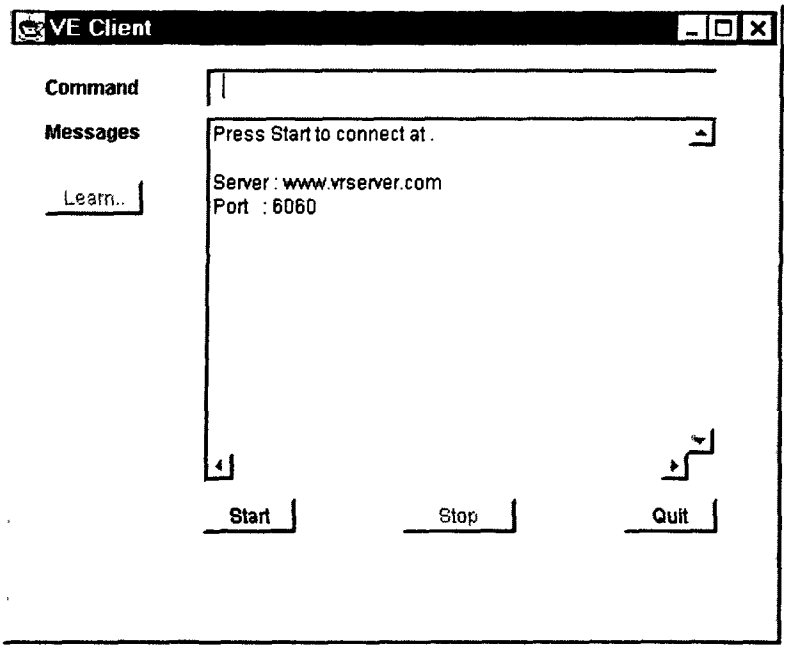
Σχήμα 5.18 Η εμφάνιση της διεπαφής του ΠΠΕΠ μπορεί να γίνει με την βοήθεια μιας HTML σελίδας η οποία βρίσκεται αποθηκευμένη στο ΠΕΔΚ.

Στην εν λόγω σελίδα εκτός από τη διαφήμιση - ενημέρωση για την υπηρεσία, τον τρόπο λειτουργίας της κ.λπ., θα πρέπει να υπάρχει και ένας σύνδεσμος στην εξής μορφή :

Webnaut Users : Click here to connect at the service

Σε αυτήν την περίπτωση, ο χρήστης δεν χρειάζεται να συμπληρώσει τα στοιχεία του εξυπηρετητή. Και στις δύο περιπτώσεις, η εμφάνιση της διεπαφής ενεργοποιείται μέσα από το πρόγραμμα πλοήγησης του χρήστη. Γι' αυτό το λόγο, πριν από την εμφάνιση της διεπαφής, είναι απαραίτητο να κλείσει η σύνδεση του προγράμματος πλοήγησης με το ΠΠΕΠ. Έτσι λοιπόν, ο ΠΠΕΠ στέλνει στην επικεφαλίδα της απόκρισης του, τον HTTP κωδικό 204 [Γ1]. Ο συγκεκριμένος κωδικός χρησιμοποιείται για να δείξει ότι η αίτηση έγινε κατανοητή, αλλά δεν υπάρχουν δεδομένα που πρέπει να στείλει ο εξυπηρετητής στον πελάτη, οπότε το πρόγραμμα πλοήγησης θα πρέπει να τερματίσει την σύνδεση.

Το σχήμα 5.19 απεικονίζει την διεπαφή αλληλεπίδρασης του χρήστη με τον εξυπηρετητή. Ο χρήστης έχει στην διάθεση του συνολικά τέσσερα κουμπιά εντολών και δύο κουτιά κειμένων. Με την βοήθεια των δύο κουμπιών *Start* και *Stop*, γίνεται η έναρξη και λήξη της σύνδεσης του χρήστη με τον απομακρυσμένο εξυπηρετητή. Η επαναλαμβανόμενη σύνδεση/αποσύνδεση του χρήστη με έναν εξυπηρετητή, έχει ιδιαίτερη αξία σε εικονικά περιβάλλοντα διασκέδασης.



Σχήμα 5.19 Η διεπαφή αλληλεπίδρασης του χρήστη με το περιβάλλον ΕΠ.

Με τα άλλα δύο κουμπιά *Learn* και *Quit*, γίνεται αντίστοιχα η τροποποίηση του προφίλ του χρήστη καθώς και ο τερματισμός της εφαρμογής. Το πλήκτρο *Learn* ενεργοποιείται με το πέρας της σύνδεσης με τον εξυπηρετητή. Έτσι λοιπόν, είναι

στην ευχέρεια του χρήστη αν θα ενημερώσει τον Πράκτορα Εκμάθησης ότι ενδιαφέρεται για σχετικές πληροφορίες με τα θέματα που έλαβαν μέρος κατά την διάρκεια της σύνδεσης.

Τα δύο κουτιά κειμένου *Messages* και *Command*, σκοπό έχουν αντίστοιχα, την εμφάνιση των μηνυμάτων του εξυπηρετητή προς τον πελάτη και την συγγραφή των εντολών του χρήστη που θα αποσταλούν προς τον εξυπηρετητή.

Η διεπαφή αλληλεπίδρασης είναι αρκετά λιτή ώστε να προσφέρει δύο σημαντικά πλεονεκτήματα. Πρώτον, δεν απαιτείται από τον χρήστη να διαθέτει επιπλέον προγράμματα π.χ. βιβλιοθήκες γραφικών ή ένα ισχυρό υπολογιστικό σύστημα επεξεργασίας οπτικοακουστικού υλικού ή/και τρισδιάστατων γραφικών. Δεύτερον, η διεπαφή στην παρούσα μορφή της μπορεί να χρησιμοποιηθεί και για συνδιάσκεψη.

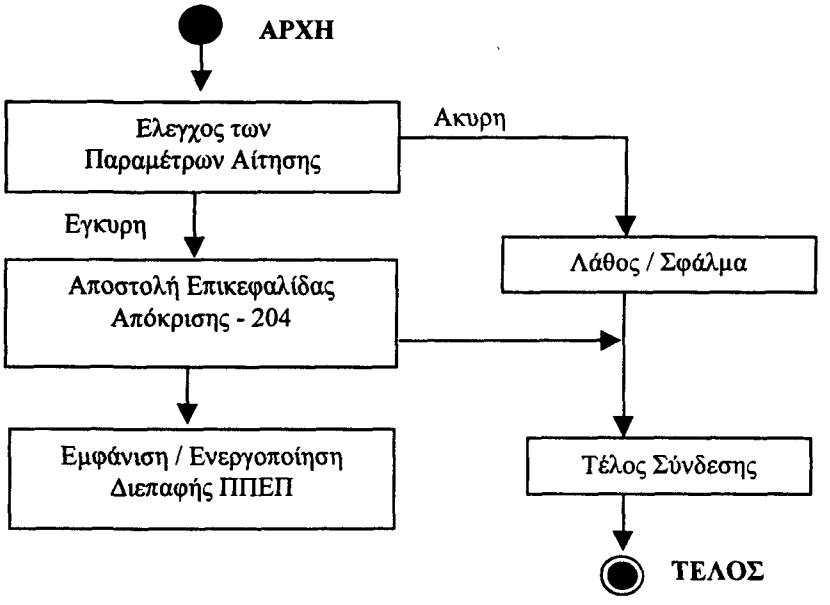
5.8.5 Λεπτομερής Σχεδιασμός του Πράκτορα Πύλη Εικονικής Πραγματικότητας

Ο ΠΠΕΠ αποτελείται από τέσσερις κλάσεις : α) την κλάση *VRGateway* [B33] β) την κλάση *SymAction* [B34], γ) την κλάση *SymWindow* [B35], και δ) κλάση *MessageDialog* [B36]. Το σχήμα 5.20 παρουσιάζει το διάγραμμα δραστηριοτήτων του ΠΠΕΠ. Η ανεξάρτητη διεργασία *VRGateway* κάνει δύο ενέργειες : α) ελέγχει αν είναι έγκυρη η αίτηση του χρήστη. Αν υπάρχει κάποιο σφάλμα ή έλλειψη στοιχείων, τότε ενημερώνεται ο χρήστης με το ανάλογο μήνυμα. β) Στην περίπτωση που δεν υπάρχει κάποιο σφάλμα, τότε τερματίζει την σύνδεση με το πρόγραμμα πλοήγησης και εν συνεχεία δημιουργεί μια διεπαφή για την επικοινωνία του χρήστη με τον εξυπηρετητή, ο οποίος προσφέρει το περιβάλλον της εικονικής πραγματικότητας.

Οι κλάσεις *SymAction* και *SymWindow* είναι εσωτερικές κλάσεις της *VRGateway*, και σκοπό έχουν να παραλαμβάνουν τα μηνύματα-συμβάντα, από το γραφικό περιβάλλον του λειτουργικού συστήματος και να τα ανακατευθύνουν προς την εφαρμογή. Η κλάση *MessageDialog*, χρησιμοποιείται για την δημιουργία παραθύρων διαλόγου για την ενημέρωση του χρήστη, σχετικά με το αποτέλεσμα διαφόρων ενεργειών, όπως για παράδειγμα, την αποτυχημένη προσπάθεια σύνδεσης του ΠΠΕΠ με τον απομακρυσμένο εξυπηρετητή.

Για την σύνδεση του χρήστη με τον εξυπηρετητή, που προσφέρει την υπηρεσία της εικονικής πραγματικότητας, θα πρέπει να είναι γνωστό το όνομα ή η IP διεύθυνση του εξυπηρετητή, καθώς επίσης και η δικτυακή πόρτα όπου "*τρέχει*" η υπηρεσία. Αφού ο χρήστης έχει στην διάθεση του αυτά τα στοιχεία, θα πρέπει να

χρησιμοποιήσει μια διεπαφή, η οποία θα του επιτρέπει να αλληλεπιδράσει με όλα τα "*αντικείμενα*" της υπηρεσίας.



Σχήμα 5.20 *Το διάγραμμα δραστηριοτήτων του ΠΠΕΠ*

Το παράδειγμα αλληλεπίδρασης που έχουμε υλοποιήσει, μας επιτρέπει να γενικεύσουμε τις όποιες παρατηρήσεις μας και σε άλλους πιο σύνθετες χώρους, για παράδειγμα, σε περιβάλλοντα συνδιασκέψεων, όπου πολλοί χρήστες μέσω μιας διεπαφής συνδέονται σε έναν κεντρικό εξυπηρετητή, ο οποίος απλά *μοιράζει* τις πληροφορίες σε όλους τους χρήστες.

Μετά την επιτυχημένη σύνδεση του πελάτη με τον εξυπηρετητή, μπορεί να ανταλλάξουν μεταξύ τους διάφορα μηνύματα. Χρησιμοποιούμε την λέξη μπορεί, επειδή σε καμία περίπτωση δεν είναι υποχρεωτική η αποστολή τέτοιων αρχικών μηνυμάτων. Σύμφωνα πάντα με το πρωτόκολλο επικοινωνίας, που έχουμε ορίσει για την επικοινωνία των δύο πλευρών, ο εξυπηρετητής μπορεί να στείλει οποιαδήποτε στιγμή μηνύματα-εντολές που χρησιμοποιούνται μόνο από την διεπαφή αλληλεπίδρασης αρκεί να τα έχει κλείσει, όπως είδαμε παραπάνω, μέσα στις ετικέτες <SERVER> και </SERVER>.

Τέτοια μηνύματα-εντολές μπορεί να έχουν σχέση και με τον κόσμο εικονικής πραγματικότητας που αλληλεπιδρά ο χρήστης, αφού σε μια πιο σύνθετη και πολύπλοκη διεπαφή, θα μπορούσε μέσα σε αυτές τις ετικέτες να περικλείονται εντολές, για να "*φορτωθεί*" ένα video ή ένα μουσικό κομμάτι από το διαδίκτυο, να αρχίσει η επίδειξη ενός video ή να μετακινηθούν σε νέες συντεταγμένες τα αντικείμενα ή οι χαρακτήρες που συμμετέχουν στον εικονικό κόσμο.

Στην παρούσα υλοποίηση, ο εξυπηρετητής αποστέλλει αρχικώς δύο μηνύματα

προς το πρόγραμμα αλληλεπίδρασης. Το πρώτο από αυτά, αναφέρεται στην επιτυχημένη αρχικοποίηση του κόσμου ΕΠ, ενώ το δεύτερο μήνυμα, που έχει την μορφή <SERVER:RESERVED WORDS>goto,go,to,look,at,..... </SERVER>, αναφέρεται στο σύνολο των δεσμευμένων λέξεων που χρησιμοποιούνται στο συγκεκριμένο περιβάλλον δράσης του χρήστη.

Μετά την παραλαβή του πρώτου μηνύματος, το πρόγραμμα θα προβεί σε ενέργειες που σχετίζονται με την αρχικοποίηση της επικοινωνίας π.χ. ενεργοποίηση και απενεργοποίηση στοιχείων στην διεπαφή, ενώ το δεύτερο μήνυμα και οι σχετικές με αυτό πληροφορίες, θα χρησιμοποιηθούν για να φιλτράρονται τα μηνύματα του χρήστη κατά την αλληλεπίδραση του με το περιβάλλον ΕΠ.

Το σύνολο των δεσμευμένων λέξεων μετά την ανάλυση τους, αποθηκεύεται για γρήγορη αναζήτηση στο *Hashtable ReserveWords* [B33]. Κάθε φορά που ο χρήστης πληκτρολογεί μια πρόταση στο αντίστοιχο κουτί κειμένου, αυτή η πρόταση, μετά την επιτυχημένη αποστολή της στον απομακρυσμένο εξυπηρετητή, αναλύεται στα δομικά στοιχεία της και εν συνεχεία, κάθε ένα από αυτά συγκρίνεται με τις λέξεις που υπάρχουν στο *Hashtable ReserveWords.* Όσες από τις λέξεις της πρότασης δεν ανήκουν στο σύνολο των δεσμευμένων λέξων, αποθηκεύονται στο *Hashtable UserCommands* [B33].

Με το πέρας της επικοινωνίας, δίνεται η δυνατότητα στο χρήστη να πατήσει το πλήκτρο *Learn,* και να χρησιμοποιήσει ο Πράκτορας Εκμάθησης το σύνολο των λέξεων που υπάρχουν στο *Hashtable UserCommands,* ώστε να βρει πληροφορίες στο διαδίκτυο, σχετικές με αυτές τις λέξεις. Όπως θα δούμε στην συνέχεια, οι συγκεκριμένες λέξεις, μαζί με τις λέξεις που εμπεριέχονται σε σελίδες του διαδικτύου που τις έχει προτείνει ο χρήστης σαν παραδείγματα των ενδιαφερόντων του, θα χρησιμοποιηθούν για την κατασκευή του προφίλ του χρήστη.

Το εικονικό περιβάλλον περιπέτειας που αναπτύξαμε, μπορεί εύκολα να χρησιμοποιηθεί σε μια εικονική βιβλιοθήκη, όπου τα διάφορα δωμάτια αντιστοιχούν σε διαφορετικά θέματα. Ο χρήστης έχει την δυνατότητα να μετακινείται στα δωμάτια, να αναζητά πληροφορίες, να απευθύνει ερωτήσεις είτε γενικά προς το σύστημα είτε προς τους άλλους χρήστες - πράκτορες ή ανθρώπους - της βιβλιοθήκης.

Σε αυτούς τους πληροφοριακούς χώρους, έχει ιδιαίτερη αξία ο διαχειριστής να παρέχει ένα σύνολο από δεσμευμένες λέξεις, οι οποίες δεν έχουν αξία για το χρήστη πέρα από τους σκοπούς της αλληλεπίδρασης του με το εικονικό περιβάλλον. Όμως αυτό το σύνολο των λέξεων θα διευκολύνει αρκετά την διαδικασία της εκμάθησης.

Όπως θα δούμε και στην συνέχεια, ακόμα και στην περίπτωση που ο εξυπηρετητής δεν παρέχει τις δεσμευμένες λέξεις, ο πράκτορας εκμάθησης χρησιμοποιεί ένα προκαθορισμένο σύνολο από κοινές λέξεις και υπάρχει πάντα η δυνατότητα να το εμπλουτίσει ο χρήστης.

Η φυσική γλώσσα, είτε μέσω ενός πληκτρολογίου είτε μέσω ενός μικροφώνου, θα χρησιμοποιείται στο μέλλον σαν το μόνο μέσο επικοινωνίας με τις διάφορες εφαρμογές. Επιλέξαμε να χρησιμοποιήσουμε την φυσική γλώσσα σαν το μέσο επικοινωνίας του χρήστη με την απομακρυσμένη εφαρμογή, με σκοπό να δείξουμε πως μπορούμε να *χτίσουμε* τέτοιου είδους εφαρμογές, οι οποίες ενοποιούνται σε ένα ολοκληρωμένο πακέτο υπηρεσιών διαδικτύου, και επιπλέον, πως οι εντολές - μηνύματα του χρήστη μπορούν να χρησιμοποιηθούν για την εκμάθηση των ενδιαφερόντων του.

5.9 Σχεδίαση του Πράκτορα Εκμάθησης

Ο Πράκτορας Εκμάθησης - ΠΕ, είναι ο πράκτορας λογισμικού ο οποίος συντονίζει και οργανώνει όλη την προσπάθεια του συστήματος, για την αναγνώριση και την εξυπηρέτηση των ενδιαφερόντων του χρήστη. Στην προσπάθεια αυτή συμμετέχει ο Πράκτορας Γενετικός Αλγόριθμος και γίνεται χρήση του Αναλυτή Κειμένων και του Ιστορικού Αρχείου.

5.9.1 Ο Αναλυτής Κειμένων

Ο Αναλυτής Κειμένων - ΑΝΚΕ, είναι ένα πολύ σημαντικό συστατικό του όλου συστήματος αφού είναι υπεύθυνο για την ανάλυση των σελίδων κειμένου που :

α) βρίσκει καθ' οδον το σύστημα στην προσπάθεια του να ανακαλύψει νέους ενδιαφέροντες διαδικτυακούς τόπους,

β) προέρχονται από την αλληλοεπίδραση του χρήστη με ένα διαδικτυακό τόπο εικονικής πραγματικότητας, και

γ) τις παρέχει ο χρήστης σαν παραδείγματα των ενδιαφερόντων του.

Την λειτουργία του ANKE την έχουμε χωρίσει στις παρακάτω φάσεις :

1) Αρχικά ένας λεκτικός αναλυτής αποσυνθέτει το κείμενο στα δομικά του στοιχεία με σκοπό να : α) συγκεντρώσει, εάν υπάρχουν, όλες τις ηλεκτρονικές διευθύνσεις που αναφέρονται στο κείμενο, β) απομακρύνει όλες τις δηλώσεις που αναφέρονται σε εκτελέσιμο κώδικα για τα προγράμματα πλοήγησης, όπως για παράδειγμα τον κώδικα σε γλώσσα JAVASCRIPT ή VBSCRIPT, γ) απομακρύνει όλες τις λέξεις που περικλείονται από τους χαρακτήρες < και >, όπως για παράδειγμα την λέξη </html>. Όλες οι λέξεις του κειμένου που δεν απομακρύνθηκαν στην πρώτη φάση και είναι μεγαλύτερες από ένα συγκεκριμένο αριθμό χαρακτήρων θα προωθηθούν προς επεξεργασία στην επόμενη φάση.

2) Εν συνεχεία, στις λέξεις του κειμένου εφαρμόζεται ο αλγόριθμος απομάκρυνσης λέξεων που παρουσιάζουν μεγάλη συχνότητα. όπως για παράδειγμα οι λέξεις "*for*", "*the*", "*and*" κ.λπ. Έπειτα θα απομακρυνθούν όλοι οι αριθμοί καθώς επίσης και όλες οι κοινές λέξεις τις οποίες συναντάμε αρκετές φορές σε κείμενα χωρίς να έχουν κάποιο ιδιαίτερο νόημα - όπως για παράδειγμα τα ονόματα των ημερών, τα ονόματα των μηνών ή άλλες λέξεις που συναντάμε αρκετές φορές στο διαδίκτυο όπως τις λέξεις "www" ή "ftp" κ.λπ. - οι οποίες περιέχονται σε ένα αρχείο που μπορεί να εμπλουτίσει και ο χρήστης και ονομάζεται *Αρχείο Κοινών Όρων.*

3) Τέλος, για την αύξηση της απόδοσης του συστήματος, στο κείμενο που έχει απομείνει μετά την εφαρμογή των παραπάνω φίλτρων, εφαρμόζουμε τον αλγόριθμο του Porter [Por80] ο οποίος αφαιρεί το πρόθεμα (π.χ. το πρόθεμα kilo ή milli ή micro από τις λέξεις kilometer ή millimeter ή micrometer αντίστοιχα ώστε να μείνει μόνο η λέξη meter) και εν συνεχεία τα παράγωγα από τις καταλήξεις των κλιτών λέξεων (π.χ. η κατάληξη της λέξης availability θα αντικατασταθεί σε available, δηλαδή η κατάληξη από bility θα μετατραπεί σε ble - και εν συνεχεία θα αφαιρεθεί ώστε να μείνει μόνο η λέξη avail)

Θα πρέπει να αναφέρουμε ότι η προσθήκη : α) της επιλογής για την ανάλυση ή όχι των αριθμών, καθώς επίσης και των λέξεων, που ο αριθμός των χαρακτήρων τους

είναι μικρότερος από ένα συγκεκριμένο όριο, και β) του *Αρχείου Κοινών Όρων*, έγιναν σε μεταγενέστερο στάδιο αφού πρώτα είδαμε την πληθώρα των αλφαριθμητικών που συνθέτουν μια τυχαία σελίδα αδόμητου κειμένου, όπως για παράδειγμα μια σελίδα στο διαδίκτυο.

Ο στόχος σε όλες αυτές τις προσθήκες ήταν διπλός : α) Η μείωση του αριθμού των προς επεξεργασία λέξεων και μάλιστα λέξεων άσχετων προς τα ενδιαφέροντα του χρήστη και β) Η ενεργός συμμετοχή του χρήστη για την βελτίωση της απόδοσης του συστήματος, που αυτό σημαίνει ότι το *Αρχείο Κοινών Όρων* θα πρέπει να εμπλουτίζεται και να τροποποιείται ανάλογα με το "*περιβάλλον δράσης*" του συστήματος.

5.9.2Λεπτομερής Σχεδιασμός του Αναλυτή Κειμένων

Ο ANKE αποτελείται από τρεις κλάσεις : α) την κλάση *TextServer* [B27], β) την κλάση *HtmlParser* [B28], γ) την κλάση *Porter* [B29]. Η κλάση *Porter* υλοποιεί τον αλγόριθμο της τρίτης φάσης και μέσω της μεθόδου *stripAffixes "καθαρίζει"* την λέξη από τα επισυναπτόμενα. Το μοναδικό στιγμιότυπο της κλάσης *TextServer* αρχικοποιείται στο σώμα της μεθόδου *StartModules* της κλάσης *ProxyServer.*

Όλες οι παράμετροι λειτουργίας του ANKE αρχικοποιούνται μέσω της μεθόδου *Initialize,* η οποία επιστρέφει το αποτέλεσμα της έγκυρης ή μη ενεργοποίησης του. Κατά την αρχικοποίηση οι παράμετροι λειτουργίας αποθηκεύονται στο στατικό *Hashtable,* με το όνομα *TextProp,* και τα περιεχόμενα του αρχείου *Αρχείο Κοινών Όρων* και του αρχείου με τις λέξεις με την μεγάλη συχνότητα, αποθηκεύονται αντίστοιχα στους στατικούς *Hashtables* με τα ονόματα *ExclLexico* και *StopWords.*

Η κλάση *HtmlParser* κληρονομεί την κλάση *TextServer* και υλοποιεί όλες τις μεθόδους για την ανάλυση του κειμένου είτε πρόκειται για ένα αρχείο σε έναν απομακρυσμένο διακομιστή είτε για ένα αρχείο στον τοπικό δίσκο. Πιο συγκεκριμένα, μετά την αρχικοποίηση ενός αντικειμένου της κλάσης με τις απαραίτητες πληροφορίες για τη διεύθυνση ή το μονοπάτι του αρχείου, εν συνεχεία η μέθοδος *Analyze* αναλαμβάνει την ανάλυση της πληροφορίας, όπως περιγράφεται στις τρεις φάσεις λειτουργίας του ANKE. Πρόσβαση στα αποτελέσματα της ανάλυσης έχουμε μέσω των δημοσίων μεθόδων [B28] *getSWords(),getSFreqs()* και *getLinks()* οι οποίες επιστρέφουν αντίστοιχα τις λέξεις του κειμένου, τις συχνότητες εμφάνισης τους και τέλος τις ηλεκτρονικές διευθύνσεις που υπάρχουν στο κείμενο.

5.9.3 Το Ιστορικό Αρχείο

Στο ΙΣτορικό ΑΡχείο - ΙΣΑΡ, αποθηκεύονται όλες οι ηλεκτρονικές διευθύνσεις που έχει προτείνει ο χρήστης ότι είναι αντιπροσωπευτικές των ενδιαφερόντων του. Το ιστορικό αρχείο είναι αρκετά χρήσιμο αφού αποτελεί την μνήμη του συστήματος μεταξύ των διαδοχικών ενεργοποιήσεων του. Επίσης αποφεύγονται οι καταστάσεις όπου το σύστημα προτείνει στον χρήστη κάτι που ο χρήστης έχει προτείνει στο παρελθόν και επιπλέον αποφεύγονται οι καταστάσεις να ακολουθήσει το σύστημα κάποια από τις ηλεκτρονικές διευθύνσεις που περιλαμβάνονται στην υπόδειξη του χρήστη.

Ο λόγος για την τελευταία συμπεριφορά έχει να κάνει με το ότι θα πρέπει το σύστημα να κερδίσει την εμπιστοσύνη του χρήστη και να μην προτείνει την ηλεκτρονική διεύθυνση που βρίσκεται στο αμέσως επόμενο επίπεδο από την υπόδειξη του χρήστη, ακόμα και αν αυτή είναι η πιο αντιπροσωπευτική και περιγραφική των ενδιαφερόντων του. Θα πρέπει το σύστημα να προτείνει καινούργιες ηλεκτρονικές διευθύνσεις ακόμα και αν αυτές είναι "*μικρότερης*" αξίας από αυτές που έχει προτείνει ο χρήστης.

Τέλος θα πρέπει να τονίσουμε ότι το ΙΣΑΡ δεν μεγαλώνει συνεχώς αλλά μέχρι ενός ορίου το οποίο περιγράφεται σαν μια από τις παραμέτρους του συστήματος. Εκτός από το προφανές πλεονέκτημα ότι ένας πολύ "*ενεργητικός*" χρήστης θα μπορούσε δυνητικά να αυξήσει υπερβολικά το ΙΣΑΡ από την άλλη πλευρά το σύστημα θα πρέπει να έχει δυναμική συμπεριφορά δηλαδή να "*ξεχνάει*" κάποιες ηλεκτρονικές διευθύνσεις οι οποίες είναι αρκετό καιρό μέσα στην μνήμη του και οι οποίες μπορεί να μην αποτελούν πλέον τα τρέχοντα ενδιαφέροντα του χρήστη.

5.9.4 Λεπτομερής Σχεδιασμός του Ιστορικού Αρχείου

Το ΙΣΑΡ αποτελείται από μία κλάση, την *HistoryServer* [B30] η οποία και υλοποιεί όλες τις απαραίτητες λειτουργίες. Όπως και σε αρκετές άλλες κλάσεις του συστήματος, το πρώτο αντικείμενο της κλάσης *HistoryServer* δημιουργείται στο σώμα της μεθόδου *StartModules* της κλάσης *ProxyServer*.

Όλες οι παράμετροι λειτουργίας του ΙΣΑΡ αρχικοποιούνται μέσω της μεθόδου *Initialize*, η οποία και επιστρέφει το αποτέλεσμα της έγκυρης ή μη ενεργοποίησης του πρώτου αντικειμένου της κλάσης. Κατά την αρχικοποίηση του πρώτου αντικειμένου, όλες οι παράμέτροι λειτουργίας αποθηκεύονται στον στατικό *Hashtable* με το όνομα

HistoryProp και όλες οι ηλεκτρονικές διευθύνσεις αποθηκεύονται στον επίσης στατικό *Hashtable* με το όνομα *HistoryItems*. Έτσι λοιπόν όλα τα στιγμιότυπα τις κλάσης "*μοιράζονται*" τα ίδια στοιχεία και επιπλέον όλα τα στιγμιότυπα *αποθηκεύουν* σε κοινόχρηστο αποθηκευτικό χώρο. Οι δημόσιες μέθοδοι *addEntry* και *contains* χρησιμοποιούνται αντίστοιχα, για την προσθήκη μια νέας ηλεκτρονικής διεύθυνσης και για την δοκιμή αν μια διεύθυνση είναι μέρος ή όχι του ΙΣΑΡ.

5.9.5 Ο Πράκτορας Γενετικός Αλγόριθμος

Ο Πράκτορας Γενετικός Αλγόριθμος (ΠΓΑ) θα χρησιμοποιήσει την πληροφορία που έχει συγκεντρώσει ο ΠΕ για να εξερευνήσει το διαδίκτυο και να συλλέξει νέα κείμενα που σχετίζονται με τα ενδιαφέροντα του χρήστη. υλοποίηση καθώς και ο τρόπος λειτουργίας του ΠΓΑ, ο οποίος χρησιμοποιεί τεχνικές γενετικής εξέλιξης περιγράφεται στο επόμενο κεφάλαιο.

5.10 Συμπεράσματα

Στο κεφάλαιο αυτό παρουσιάσαμε την αρχιτεκτονική του συστήματος *Webnaut* το οποίο αποτελείται από διαφορετικούς πράκτορες λογισμικού. Ειδικότερα, αναλύσαμε τις τεχνικές λεπτομέρειες που διέπουν την σχεδίαση, το πλαίσιο λειτουργίας, καθώς και την υλοποίηση των επιμέρους συστημάτων – πρακτόρων λογισμικού.

Το σύστημα Webnaut υποστηρίζει ένα χρήστη του διαδικτύου, καθ' όλη την διάρκεια της πλοήγησης του, σε μια σειρά από χρονοβόρες και κουραστικές εργασίες. Ο σχεδιασμός του συστήματος επιτρέπει την εύκολη ενσωμάτωση νέων λειτουργιών και υπηρεσιών, την συνεργασία του συστήματος με όλους τους φυλλομετρητές, την προσθήκη νέων πηγών πληροφόρησης, την άντληση πληροφοριών για τα ενδιαφέροντα του χρήστη από την αλληλεπίδραση του χρήστη με διάφορες υπηρεσίες του διαδικτύου.

Κεφάλαιο 6

Η ΛΕΙΤΟΥΡΓΙΑ ΤΗΣ ΠΡΟΣΑΡΜΟΖΟΜΕΝΗΣ ΕΚΜΑΘΗΣΗΣ ΤΩΝ ΕΝΔΙΑΦΕΡΟΝΤΩΝ ΤΟΥ ΧΡΗΣΤΗ

6.1 Εισαγωγή

Η εκμάθηση των ενδιαφερόντων ενός ή μιας ομάδας χρηστών είναι ένα αρκετά δύσκολο θέμα και εξαρτάται συνήθως από το περιβάλλον που δραστηριοποιείται ο πράκτορας. Η χρήση των κανόνων είναι μια θεωρητικά απλή και σίγουρη μέθοδος από την άποψη ότι οι άνθρωποι συνήθως βρίσκουν πιο εύκολο να περιγράψουν τις απαραίτητες ενέργειες που πρέπει να γίνουν σε ένα πρόβλημα με την μορφή κανόνων. Για παράδειγμα, ένας κανόνας για την διαχείριση του ηλεκτρονικού ταχυδρομείου ενός χρήστη, θα μπορούσε να ήταν ο εξής :

```
IF    MESSAGE.Subject.Contains("CFP")        THEN
      MESSAGE.Move(Conferences)
END IF
```

Ο παραπάνω κανόνας, τοποθετεί στο φάκελο με το όνομα Conferences, όλα τα μηνύματα που έχουν στον τίτλο τους, το αλφαριθμητικό CFP (Call for Papers). Ένα τέτοιο σύστημα, θα μπορούσε να βοηθήσει τον χρήστη στην διαγραφή των μηνυμάτων που αφορούν συνέδρια των οποίων η καταληκτική ημερομηνία για την αποστολή εργασιών αναφέρεται σε προηγούμενη ημερομηνία, ή στην ανταπάντηση ενός μηνύματος ανάλογα με τον αποστολέα ή το θέμα κ.λπ.

Υπάρχουν αρκετά προγράμματα που βασίζονται στην τεχνολογία των πρακτόρων και κάνουν χρήση κανόνων για την κατηγοριοποίηση των μηνυμάτων [Coh95, Gro95]. Συνήθως αυτού του είδους τα προγράμματα, εκτός από την ευκολία περιγραφής του προβλήματος με την βοήθεια των κανόνων, προτάσσουν και ένα ακόμα επιχείρημα ότι δηλαδή, οι χρήστες μπορούν επίσης εύκολα να τροποποιήσουν τους κανόνες όταν φυσικά τα ενδιαφέροντα τους αλλάζουν με αρκετά γρήγορο ρυθμό. Όμως ένα τέτοιο σύστημα, όπως αυτό που περιγράψαμε παραπάνω, είναι προφανές ότι απευθύνεται σε χρήστες οι οποίοι θα πρέπει :

- Να έχουν καλή γνώση και τεχνική κατάρτιση του χώρου δραστηριοποίησης του πράκτορα. Για παράδειγμα θα πρέπει να γνωρίζουν τι είναι το ηλεκτρονικό ταχυδρομείο και την σχετική ορολογία.

- Να έχουν καλή γνώση προγραμματισμού ώστε χρησιμοποιώντας τα εργαλεία και τα αντικείμενα που παρέχει το εργαλείο ανάπτυξης του πράκτορα να δημιουργήσουν τους κατάλληλους κανόνες.

- Τέλος θα πρέπει να ελέγχουν τους κανόνες, όσον αφορά, α) την ορθή περιγραφή και κωδικοποίηση του προβλήματος και β) ότι ενεργοποιούνται με την σωστή σειρά και δεν υπάρχουν συγκρούσεις μεταξύ τους. Επιπλέον θα πρέπει να εμπλουτίζουν ή να τροποποιούν τους κανόνες για να εξυπηρετεί το σύστημα τυχόν αλλαγές στα ενδιαφέροντα του χρήστη.

Από τα παραπάνω είναι προφανές, ότι ο τελικός χρήστης έχει αρκετά ενεργό συμμετοχή στην εκμάθηση του συστήματος, κάτι που θα αποτρέψει ένα μεγάλο πλήθος χρηστών να χρησιμοποιήσουν πράκτορες που κάνουν χρήση κανόνων ή τουλάχιστον θα τους χρησιμοποιήσουν για περιορισμένο εύρος προβλημάτων. Βέβαια δεν είναι κατά ανάγκη κακή η ενεργός συμμετοχή του τελικού χρήστη ειδικά όταν πρόκειται για έμπειρο χρήστη με αρκετή γνώση γύρω από το πρόβλημα.

Επίσης μία άλλη προσέγγιση στην κατασκευή πρακτόρων χωρίς την άμεση συμμετοχή του τελικού χρήστη, στην γνώση του συστήματος, είναι με την χρήση μιας βάσης γνώσεως [Gao98, Lam96]. Σε αυτή την περίπτωση, ο πράκτορας έχει προκαθορισμένη γνώση σχετικά με την οργανωτική δομή, τις υπηρεσίες, τον τρόπο ανάκτησης και ανάλυσης των πληροφοριών από συγκεκριμένους κόμβους. Για παράδειγμα, η εφαρμογή WebKB [Cra00] είναι ένα ερευνητικό πρόγραμμα το οποίο έχει σαν σκοπό την αυτόματη δημιουργία μιας βάσης γνώσης από το παγκόσμιο ιστό.

Ο πράκτορας αρχικά εκπαιδεύεται στην εξαγωγή μερικών συγκεκριμένων τύπων πληροφορίας και στη συνέχεια διασχίζει αυτόνομα το δίκτυο με σκοπό να συγκεντρώσει και να εμπλουτίσει με νέες πληροφορίες τη βάση γνώσης. Ο μηχανικός γνώσης, κατά την φάση της εκπαίδευσης, καθορίζει : α) την *οντολογία*, δηλαδή τις κλάσεις των αντικειμένων της εφαρμογής (π.χ. πρόσωπο, καθηγητής, μαθητής κ.λπ.) την ιεραρχία τους καθώς και τις σχέσεις (π.χ. διδάσκει) που είναι σημαντικές στα πλαίσια της εφαρμογής, β) παραδείγματα από κείμενα που σχετίζονται με την

οντολογία. Μετά το στάδιο της εκπαίδευσης, ο πράκτορας έχει μάθει την διαδικασία εξαγωγής νέων στιγμιότυπων των κλάσεων καθώς και των σχέσεων τους, από τον παγκόσμιο ιστό.

Ένα σημαντικό πλεονέκτημα αυτής της προσέγγισης, είναι ότι ο τελικός χρήστης του συστήματος έχει ελάχιστη ή καθόλου συμμετοχή στην γνώση του συστήματος. Βέβαια από την άλλη πλευρά, τον φόρτο εργασίας για την δημιουργία και την εκπαίδευση του συστήματος τον επωμίζεται ο μηχανικός της γνώσης. Επιπλέον, η συγκεκριμένη προσέγγιση χρειάζεται αρκετή εμπειρία από το μηχανικό γνώσης και την ομάδα του, ώστε το σύστημα να είναι σε θέση να αντιλαμβάνεται και να προσαρμόζεται στις ιδιαιτερότητες των διαφορετικών χρηστών, καθώς και σε διαφορετικά επίπεδα γνώσης του πεδίου εφαρμογής.

Οι πράκτορες που κάνουν χρήση των τεχνικών από το χώρο της Μηχανικής Εκμάθησης αποκτούν αυτόματα όλη την γνώση που απαιτείται για να βοηθήσουν το χρήστη. Αυτή η προσέγγιση, έχει δύο πλεονεκτήματα :

α) απαιτείται ελάχιστη προσπάθεια από τον χρήστη ή τον μηχανικό γνώσης, που υλοποιεί τον πράκτορα

β) ο πράκτορας μπορεί να προσαρμόζεται στους διαφορετικούς χρήστες.

Βέβαια, σε αυτήν την προσέγγιση, εκτός από την εκμάθηση, αρκετά άλλα θέματα που σχετίζονται με αυτήν θα πρέπει να επιλυθούν κατά την σχεδίαση του πράκτορα, όπως για παράδειγμα, πότε αρχίζει και πότε τελειώνει η εκμάθηση του, πως ανανεώνει τις γνώσεις του, θέματα ασφάλειας και εμπιστοσύνης μεταξύ του χρήστη και του πράκτορα ή μεταξύ του πράκτορα και άλλων πρακτόρων.

6.2 Η Λειτουργία του Πράκτορα Εκμάθησης

Ο Πράκτορας Εκμάθησης - ΠΕ, είναι ένας πράκτορας λογισμικού ο οποίος κάνει χρήση των τεχνικών της Μηχανικής Εκμάθησης για να φέρει σε πέρας το δύσκολο έργο της εκμάθησης των ενδιαφερόντων του χρήστη και παράλληλα να ανακαλύψει και να προτείνει νέα κείμενα τα οποία θα συλλέξει από το διαδίκτυο.

Οπως είδαμε στο προηγούμενο κεφάλαιο, ο χρήστης έχει την δυνατότητα να πλοηγηθεί στο διαδίκτυο με την βοήθεια του συστήματος Webnaut αφού πρώτα

δηλώσει στο πρόγραμμα πλοήγησης να χρησιμοποιήσει σαν διακομιστή διαμεσολάβησης τον Πράκτορα Διακομιστή Διαμεσολάβησης. Μια από τις λειτουργίες του ΠΔΔ είναι και η εισαγωγή, αν φυσικά ο χρήστης το επιθυμεί, της ΠΚΕ σε κάθε σελίδα που χαρακτηρίζεται, από το διακομιστή που την εξυπηρετεί, σαν σελίδα κειμένου.

Στο σχήμα 6.1 εμφανίζεται η ΠΚΕ, οι λειτουργίες της οποίας ενεργοποιούνται με το πάτημα στο αντίστοιχο εικονίδιο. Έτσι λοιπόν, ο χρήστης κατά την διάρκεια της πλοήγησης του στο διαδίκτυο, έχει τη δυνατότητα να μαρκάρει την τρέχουσα σελίδα, που εμφανίζεται στο πρόγραμμα πλοήγησης, σαν ένα κομμάτι από τα ενδιαφέροντα του. Η ενημέρωση του συστήματος ότι η τρέχουσα σελίδα είναι σύμφωνη με τα ενδιαφέροντα του χρήστη γίνεται με το πάτημα ενός κουμπιού στην ΠΚΕ, όπως εμφανίζεται στο παραπάνω σχήμα. Πιστεύουμε ότι η συγκεκριμένη διαδικασία πρέπει να είναι απλή, φιλική και χωρίς περιττές ερωτήσεις για την προτίμηση του χρήστη.

Σχήμα 6.1 *Το κουμπί ενημέρωσης του συστήματος στην ΠΚΕ*

Ο ΠΕ θα αποθηκεύσει την διεύθυνση της προτίμησης του χρήστη σε ένα εσωτερικό πίνακα αλλά και στο Ιστορικό Αρχείο. Ένα αντίγραφο των διευθύνσεων του εσωτερικού πίνακα αποθηκεύονται μέχρι να τύχουν επεξεργασίας και σε ένα τοπικό αρχείο ώστε να διατηρούνται οι προτιμήσεις του χρήστη μεταξύ των διαφορετικών ενεργοποιήσεων του προγράμματος. Έτσι λοιπόν ένας έμπειρος χρήστης μπορεί να μεταφέρει αρκετά εύκολα όλες τις διευθύνσεις που τον ενδιαφέρουν στο συγκεκριμένο αρχείο και να τύχουν επεξεργασίας στην επόμενη ενεργοποίηση του συστήματος. Επίσης η συγκεκριμένη δυνατότητα σκοπό έχει την γρήγορη και εύκολη ενημέρωση των μελών μιας ομάδας για τις κοινού ενδιαφέροντος ηλεκτρονικές διευθύνσεις.

6.3 Το Προφίλ των Ενδιαφερόντων του Χρήστη

Το σύνολο των λέξεων που περιέχονται στα κείμενα που ενδιαφέρουν τον χρήστη, μετά την επεξεργασία τους από τον Αναλυτή Κειμένων, θεωρούνται από το σύστημα

σαν τα στοιχειώδη τμήματα που συνθέτουν το προφίλ του χρήστη. Η όλη διαδικασία ομοιάζει αρκετά με την επίλυση ενός γρίφου και αυτό γιατί γνωρίζουμε ότι ο χρήστης ενδιαφέρεται για τα συγκεκριμένα σύνολα λέξεων που περιλαμβάνει η κάθε προτίμηση του αλλά δεν γνωρίζουμε αν έχει συμμετοχή μια συγκεκριμένη λέξη και σε τι ποσοστό. Είναι αρκετά πολύπλοκη διαδικασία ακόμα για τον ίδιο τον χρήστη αν του ζητούσαμε να απομακρύνει από την κάθε σελίδα που τον ενδιαφέρει όλες τις λέξεις που θεωρεί ότι δεν είναι χρήσιμες για την περιγραφή των ενδιαφερόντων του, και εν συνεχεία να βαθμολογήσει τις εναπομείνασες λέξεις ανάλογα με το πόσο περιγραφικές είναι για τα ενδιαφέροντα του.

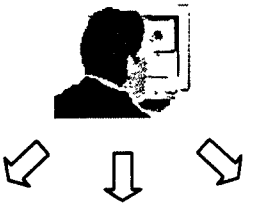
Θα πρέπει σε αυτό το σημείο να τονίσουμε ότι οι λέξεις που υπάρχουν στις προτιμήσεις του χρήστη έχουν αξία για μια συγκεκριμένη περίοδο του χρόνου. Πιο απλά, συγκεκριμένες λέξεις έχουν αξία ανάλογα με την γνώση ή εμπειρία που έχει ο χρήστης τη συγκεκριμένη χρονική στιγμή ενώ παίζει ρόλο και το συνολικό ενδιαφέρον του χρήστη για το θέμα που περιγράφουν αυτές οι λέξεις. Με το πέρασμα του χρόνου μπορεί το ενδιαφέρον για το θέμα συνολικά να μειωθεί ή αντίθετα να αυξηθεί η εμπειρία του χρήστη με αποτέλεσμα να επιζητεί πιο εξειδικευμένες γνώσεις που περιγράφονται και από διαφορετικές λέξεις. Άρα η όλη διαδικασία έχει να κάνει με την *προσαρμογή* του προφίλ με τα πραγματικά ενδιαφέροντα του χρήστη και είναι μια δυναμική διαδικασία αφού οι λέξεις θα πρέπει να σχετίζονται με κάποιο *βάρος* το οποίο θα υποδηλώνει το συνολικό ενδιαφέρον του χρήστη για τη συγκεκριμένη λέξη. Η αύξηση ή μείωση του βάρους θα πρέπει να αντανακλά την αύξηση ή την μείωση του ενδιαφέροντος του χρήστη.

Παράλληλα με την προσαρμογή, η *εξερεύνηση* του προφίλ για νέες θεματικές περιοχές ενδιαφερόντων, οι οποίες ίσως να μην έχουν υποπέσει στην αντίληψη του χρήστη, θα πρέπει να αποτελεί μέρος της όλης διαδικασίας. Πιο απλά, το σύστημα θα πρέπει να συνθέτει τα στοιχειώδη τμήματα του προφίλ του χρήστη με σκοπό την δημιουργία νέων ομάδων ενδιαφερόντων οι οποίες μπορεί να είναι αρεστές ή όχι στον τελικό αποδέκτη των υποδείξεων του συστήματος. Στην προκειμένη περίπτωση ο ΠΕ χρησιμοποιεί αυτόνομα το διαδίκτυο για την ανεύρεση νέων κειμένων, τα οποία θα περιέχουν τις ίδιες λέξεις με τις προτιμήσεις του χρήστη και μάλιστα σε συνδυασμούς τους με λογικούς τελεστές, ώστε να είναι δυνατόν ο πράκτορας να δημιουργήσει σύνθετες δομές αναπαράστασης των μακροχρόνιων ενδιαφερόντων του χρήστη.

6.4 Η Δημιουργία του Προφίλ του Χρήστη.

Για την επίτευξη των παραπάνω στόχων έχουμε δανειστεί από τον χώρο της ανάκτησης της πληροφορίας (Information Retrieval) το διανυσματικό μοντέλο [Sal71, Sal83] για την αναπαράσταση της πληροφορίας που περιέχεται στα κείμενα. Σύμφωνα με το διανυσματικό μοντέλο, όλες οι λέξεις που περιέχονται σε μια συλλογή κειμένων εξάγονται ώστε να δημιουργήσουν ένα μονοδιάστατο πίνακα D, όπου το κάθε στοιχείο του είναι μια λέξη di.

Έτσι λοιπόν το κάθε κείμενο στη συλλογή μπορεί να αναπαρασταθεί από έναν άλλο μονοδιάστατο πίνακα W, όπου το κάθε στοιχείο του, wi, είναι το βάρος της λέξης di για το συγκεκριμένο κείμενο. Το βάρος αναπαριστά την αξία της συγκεκριμένης λέξης για το συγκεκριμένο κείμενο.



Παράδειγμα 1 | Παράδειγμα 2 | Παράδειγμα n

Intelligent agents continuously perform three functions: perception of dynamic	*Genetic Algorithms differs from optimization techniques which maintain only one single point evaluation .....*	*MachineLearning Techniques Reinforcement learning is learning what to do...*

Πράκτορας Εκμάθησης

ΠΡΟΦΙΛ ΧΡΗΣΤΗ		
ΛΕΞΕΙΣ	**ΚΕΙΜΕΝΑ**	**ΣΥΧΝΟΤΗΤΑ**
Intelligent	10	20
Agents	8	18
....	....	....
Learning	....	....
Java	....	....
Genetic	....	....
....	....	....
Algorithms	....	....
Server	....	....
Reinforcement	....	....
....	....	....

***Σχήμα 6.2** Σχηματική αναπαράσταση της δημιουργίας του προφίλ ενός χρήστη*

Το βάρος μπορεί να έχει δυαδική τιμή δηλαδή 1 αν εμπεριέχεται η συγκεκριμένη λέξη

στο συγκεκριμένο κείμενο ή 0 αν δεν εμφανίζεται η λέξη. Ακόμα σαν βάρος μπορεί να χρησιμοποιηθεί η συχνότητα εμφάνισης της λέξης στο συγκεκριμένο κείμενο. Μια αρκετά γνωστή μέθοδος υπολογισμού του βάρους μιας λέξης w_i είναι η tfidf (term frequency inverse document frequency), η οποία δίδεται από την παρακάτω σχέση :

$$w_i = tf_i * \log(N/ N_i) \qquad \textbf{(6.1)}$$ όπου

w_i είναι το βάρος της λέξης i για ένα συγκεκριμένο κείμενο στην συλλογή
tf_i η συχνότητα εμφάνισης της λέξης i για το συγκεκριμένο κείμενο στην συλλογή
N ο συνολικός αριθμός των κειμένων της συλλογής
N_i ο συνολικός αριθμός των κειμένων στην συλλογή που εμπεριέχουν την λέξη i

Σε τυπικά συστήματα ανάκτησης πληροφορίας η ερώτηση του χρήστη αναπαριστάται κατά παρόμοιο τρόπο ώστε να δημιουργηθεί ο μονοδιάστατος πίνακας, Q. Η ομοιότητα μεταξύ της ερώτησης του χρήστη και ενός κειμένου ή μεταξύ δύο κειμένων μπορεί να υπολογιστεί από το εσωτερικό γινόμενο των αντίστοιχων πινάκων βάρους. Η μέτρηση αυτή είναι ίδια με το συνημίτονο της γωνίας μεταξύ δύο διανυσμάτων και γι' αυτό το λόγο ονομάζεται και *ομοιότητα συνημίτονου* και δίδεται από την παρακάτω σχέση :

$$sim(Q,D) = \frac{\sum_{k=1}^{N} w_{Qk} w_{Dk}}{\sqrt{\sum_{k=1}^{N} w_{Qk}^2 \sum_{k=1}^{N} w_{Dk}^2}} \qquad \textbf{(6.2)}$$ όπου

D, Q είναι αντίστοιχα τα δύο κείμενα που μετράμε την ομοιότητα τους μετά την κανονικοποίηση τους ώστε να έχουν την ίδια διάσταση.
N είναι ο αριθμός των λέξεων στην συλλογή των κειμένων
w_{QK} είναι το βάρος της λέξης κ για το κείμενο Q
w_{DK} είναι το βάρος της λέξης κ για το κείμενο D

Στην σχεδίαση μας, οι λέξεις με το μεγαλύτερο βάρος, από όλα τα κείμενα που παρέχει ο χρήστης σαν παραδείγματα, συγχωνεύονται σε ένα αρχείο το οποίο το ονομάζουμε *Λεξικό* και *το* θεωρούμε ως το προφίλ του χρήστη. Θα μπορούσαμε να θεωρήσουμε το προφίλ του χρήστη σαν ένα εικονικό κείμενο όπου περιέχει ανά πάσα

στιγμή όλες τις λέξεις που περιγράφουν τα ενδιαφέροντα του χρήστη για τη δεδομένη χρονική στιγμή.

Το προφίλ του χρήστη αναπαριστάται από ένα πίνακα Nx3, όπου N είναι ο αριθμός των λέξεων που περιέχονται στα κείμενα που ο χρήστης θεωρεί σαν αντιπροσωπευτικά των ενδιαφερόντων του. Η πρώτη στήλη του πίνακα περιέχει τις λέξεις, η δεύτερη στήλη περιέχει το συνολικό αριθμών κειμένων που περιέχουν τη συγκεκριμένη λέξη και η τελευταία στήλη περιέχει το άθροισμα των συχνοτήτων της λέξης σε όλα τα κείμενα που εμφανίζεται.

Πιστεύουμε ότι ο ανωτέρω τρόπος αναπαράστασης του *Λεξικού* μας επιτρέπει να συγκρίνουμε και να βαθμολογήσουμε διαφορετικούς τρόπους υπολογισμού του βάρους των λέξεων. Όλες οι λέξεις του *Λεξικού* είναι ταξινομημένες βάση του βάρους τους, το οποίο δίνεται από τον παρακάτω τύπο :

$$w_i = \frac{\left(\frac{tf_i}{tf_{\max}}\right)}{\sqrt{\sum_{j=1}^{N}\left(\frac{tf_j}{tf_{\max}}\right)^2}} \qquad \textbf{(6.3)}$$

όπου tfi είναι η συχνότητα της λέξης i σε όλα κείμενα που εμφανίζεται (τρίτη στήλη στον πίνακα), tfmax είναι η μεγαλύτερη συχνοτητα που εμφανίζεται στο *Λεξικό*, και N είναι ο αριθμός των λέξεων του *Λεξικού*. Με σκοπό να αποφύγουμε προβλήματα υπερφόρτωσης της μνήμης, έχουμε περιορίσει στις 1000 τον αριθμό των λέξεων που αποθηκεύονται στο *Λεξικό*. Βέβαια ο αριθμός αυτός αποτελεί μία από τις παραμέτρους του συστήματος η οποία καθορίζεται από το χρήστη του συστήματος.

Ο υπολογισμός του βάρους των λέξεων με το τύπο tfidf έχει υλοποιηθεί αλλά δεν χρησιμοποιείται γιατί το *Λεξικό* θεωρείται σαν ένα εικονικό κείμενο που περιέχει μόνο τις λέξεις που ενδιαφέρουν τον χρήστη. Βέβαια το tfidf είναι αρκετά δημοφιλές σε παρόμοιες εφαρμογές, αλλά θα πρέπει να υπάρχει μια συλλογή από κείμενα για να το εφαρμόσουμε. Ο χρήστης μπορεί να αλλάξει τις ρυθμίσεις του συστήματος και να χρησιμοποιήσει το tfidf αλλά σε αυτή την περίπτωση θα πρέπει να δημιουργήσει μια συλλογή από κείμενα δηλαδή να παρέχει ικανό αριθμό παραδειγμάτων, πριν αρχίσει ο Πράκτορας Εκμάθησης την διαδικασία εξερεύνησης για νέα πληροφορία.

Έτσι λοιπόν το βάρος που εμείς χρησιμοποιούμε δεν εξαρτάται από τον αριθμό των κειμένων που περιέχει το *Λεξικό* και κατ' αυτό τον τρόπο ο Πράκτορας Εκμάθησης μπορεί να φανεί χρήσιμος στον χρήστη ακόμα και από το πρώτο κείμενο

που θα παρέχει στο σύστημα σαν παράδειγμα. Από την άλλη πλευρά, το βάρος της κάθε λέξης εξαρτάται μόνο από την συχνότητα εμφάνισης της και έτσι είναι αρκετά εύκολο να μειώσουμε ή να αυξήσουμε αυτό το βάρος, σύμφωνα με την βαθμολόγηση του χρήστη. Τέλος ο ΠΕ είναι υπεύθυνος για την συλλογή και την παρουσίαση όλων των νέων πληροφοριών πού έχει συλλέξει ο ΠΓΑ. Η θετική ή αρνητική βαθμολογία των αποτελεσμάτων θα χρησιμοποιηθεί για την τροποποίηση των βαρών των λέξεων του *Λεξικού*.

6.5 Λεπτομερής Σχεδιασμός του Προφίλ του Χρήστη

Η κλάση *VDict* [B26] υλοποιεί όλες τις απαραίτητες μεθόδους για τη διαχείριση του προφίλ ενός χρήστη. Το πρώτο στιγμιότυπο της κλάσης αρχικοποιείται από την μέθοδο *InitDictionary* του ΠΕ [B17]. Όλες οι λέξεις, το σύνολο των κειμένων που τις περιέχει, καθώς επίσης και οι συχνότητες εμφάνισης τους αποθηκεύονται σε τρεις στατικούς *Vectors* [A3], με τα ονόματα *Words*, *Docs* και *Freqs* [B26]. Η μέθοδος *Initialize* επιστρέφει μια τιμή *Αληθής* ή *Ψευδής* ανάλογα με την ορθή αρχικοποίηση της κλάσης, πρακτικά την ανάγνωση ενός τοπικού αρχείου όπου σε αυτό είναι αποθηκευμένο το προφίλ του χρήστη. Η δημόσια μέθοδος *Initialize* θα καλέσει με τη σειρά της την τοπική μέθοδο *ReadDictionary* η οποία σε περίπτωση σφάλματος θα δημιουργήσει μια παρέκκλιση στην ροή του προγράμματος.

Το αρχείο του προφίλ είναι ένα αρχείο κειμένου. Όλες οι γραμμές, εκτός από την πρώτη, περιέχουν μια λέξη και δύο ακέραιους αριθμούς. Ο πρώτος αριθμός αντιστοιχεί στο σύνολο της συχνότητας εμφάνισης της λέξης και ο δεύτερος στο σύνολο των κειμένων που περιέχουν τη συγκεκριμένη λέξη. Η μέθοδος *AddElement* χρησιμοποιείται για να γεμίσει τους τρεις *Vectors- Words*, *Docs* και *Freqs-* με τα περιεχόμενα της κάθε σειράς του αρχείου. Η πρώτη σειρά του αρχείου περιέχει το συνολικό αριθμό των κειμένων που έχει αξιολογήσει ο χρήστης καθ' όλη την διάρκεια λειτουργίας του προγράμματος.

Οι αλλαγές που συμβαίνουν στο προφίλ κατά την διάρκεια της εκτέλεσης του προγράμματος αποθηκεύονται στο τοπικό αρχείο, με την βοήθεια της δημοσίας μεθόδου *SaveDictionary*, η οποία επαναδημιουργεί το τοπικό αρχείο και αποθηκεύει το συνολικό αριθμών των κειμένων καθώς και τα περιεχόμενα των τριών *Vectors*.

Οι μέθοδοι *AddEntries*, και *ModifyDictionary* επιφέρουν αλλαγές στο προφίλ του χρήστη. Η ανάλυση ενός κειμένου που προτείνει ο χρήστης σαν παράδειγμα των

ενδιαφερόντων του, καταλήγει σε δύο *Hashtable,* έναν με τις λέξεις του κειμένου και ένα με τις αντίστοιχες συχνότητες εμφάνισης τους στο συγκεκριμένο κείμενο - δες λειτουργία του Αναλυτή Κειμένου - οι οποίοι αποτελούν τα ορίσματα της μεθόδου *AddEntries*. Όταν πρόκειται για μια λέξη που δεν υπάρχει στο *Λεξικό* τότε θα δημιουργηθεί μια νέα καταχώρηση στο προφίλ του χρήστη, που θα περιέχει την συγκεκριμένη λέξη, την συχνότητα εμφάνισης της και την μονάδα που δηλώνει τον αριθμό των κειμένων που την εμπεριέχουν. Στην αντίθετη περίπτωση, δεν θα δημιουργηθεί νέα καταχώρηση αλλά θα αυξηθεί η συνολική συχνότητα εμφάνισης της λέξης κατά τον αριθμό που εμφανίζεται στο κείμενο και επίσης θα αυξηθεί κατά ένα ο αριθμός των κειμένων που την περιέχουν.

Η μέθοδος *ModifyDictionary* επιφέρει αλλαγές στο προφίλ του χρήστη μετά την αξιολόγηση του - όπως θα δούμε παρακάτω - σε μία από τις υποδείξεις του συστήματος. Και αυτή η μέθοδος δέχεται σαν ορίσματα δύο *Hashtables* που περιέχουν τις λέξεις και τις συχνότητες εμφάνισης τους στο κείμενο της υπόδειξης αλλά επιπλέον υπάρχει και η βαθμολογία του χρήστη. Η μέθοδος *CalculateWeights* χρησιμοποιεί την σχέση 6.3 για να γεμίσει τον πίνακα *Weights* με τα βάρη των λέξεων που εμπεριέχονται τη συγκεκριμένη στιγμή στο προφίλ του χρήστη. Η τοπική μέθοδος *SortPopulation* έχει επικουρικό ρόλο αφού διευκολύνει όλους τους υπολογισμούς ταξινομώντας τους *Vectors,* που αποτελούν το προφίλ του χρήστη, βάση του βάρους της κάθε λέξης.

Η μέθοδος *Similarity* δέχεται σαν όρισμα ένα κείμενο που έχει κωδικοποιηθεί σε δύο *Hashtables,* έναν με τις λέξεις και ένα με τις συχνότητες εμφάνισης τους, και στην συνέχεια υπολογίζει με την βοήθεια της σχέσης 6.2 την ομοιότητα του κειμένου με το προφίλ του χρήστη. Η τιμή που επιστρέφει η *Similarity* είναι ένας πραγματικός αριθμός από 0 έως 1. Τέλος η μέθοδος *DecreaseDictionary* υλοποιεί την λειτουργία που περιγράψαμε παραπάνω σχετικά με τον περιορισμό των λέξεων που είναι αποθηκευμένες στο προφίλ του χρήστη.

6.6 Σχεδίαση του Πράκτορα Γενετικός Αλγόριθμος

Ο Πράκτορας Γενετικός Αλγόριθμος - ΠΓΑ, θα χρησιμοποιήσει την πληροφορία που έχει συγκεντρώσει ο Πράκτορας Εκμάθησης για να εξερευνήσει το διαδίκτυο και να συλλέξει νέα κείμενα που σχετίζονται με τα ενδιαφέροντα του χρήστη. Η υλοποίηση καθώς και ο τρόπος λειτουργίας του ΠΓΑ, ο οποίος χρησιμοποιεί εξελικτικές τεχνικές

περιγράφεται στην επόμενη ενότητα.

6.6.1 Οι Γενετικοί Αλγόριθμοι

Οι Γενετικοί Αλγόριθμοι - ΓΑ, όπως τους περιέγραψε ο Holland [Hol75], είναι υπολογιστικά μοντέλα επίλυσης προβλημάτων τα οποία μιμούνται τις διαδικασίες της βιολογικής εξέλιξης τις οποίες διέπουν οι αρχές της φυσικής επιλογής και της επικράτησης του ισχυρότερου. Οι ΓΑ έχουν βρει πεδίο εφαρμογής σε προβλήματα εύρεσης και βελτιστοποίησης καθώς επίσης και ταξινόμησης.

Οι άλλες τεχνικές επίλυσης των ανωτέρω προβλημάτων συντηρούν για εκτίμηση μόνο ένα σημείο στο χώρο έρευνας και βάση της τιμής του σημείου αυτού οδηγούν την έρευνα στο χώρο που ορίζει το συγκεκριμένο πρόβλημα. Σε αντίθεση, οι ΓΑ διατηρούν ένα σύνολο από σημεία στο χώρο έρευνας, το οποίο ονομάζεται πληθυσμός και επιπλέον λειτουργούν σε κύκλους που ονομάζονται γενεές. Κατά την διάρκεια της κάθε γενεάς εκτελούνται οι ακόλουθες εργασίες :

1. Κάθε μέλος του πληθυσμού αποτιμάται και του αποδίδεται μια τιμή η οποία και αντιπροσωπεύει την αξία του, στα πλαίσια του προβλήματος.

2. Μερικά μέλη του πληθυσμού επιλέγονται για αναπαραγωγή.

3. Στα μέλη αυτά εφαρμόζονται γενετικοί μηχανισμοί οι οποίοι και θα δημιουργήσουν νέα μέλη στον πληθυσμό και ταυτόχρονα νέες λύσεις για το πρόβλημα.

Στο πίνακα 6.1 φαίνεται ο κύκλος λειτουργίας του Γενετικού Αλγορίθμου. Η λειτουργία ενός ΓΑ αρχίζει με την δημιουργία ενός αρχικού πληθυσμού, ο οποίος αποτελείται από τυχαία δημιουργημένες δομές, όπου η κάθε μια αναπαριστά για το πρόβλημα που επιλύεται μια υποψήφια λύση. Η επιλογή μιας αποτελεσματικής αναπαράστασης, η οποία θα κωδικοποιεί επιτυχώς τις τιμές των μεταβλητών του προβλήματος στο σώμα της κάθε δομής του πληθυσμού, είναι ένα από τα πιο βασικά στοιχεία για την επίλυση ενός προβλήματος με την χρήση των ΓΑ.

Μια αρκετή συνηθισμένη αναπαράσταση είναι η δυαδική, όπου το κάθε άτομο του πληθυσμού αναπαριστάται από ένα σταθερού μήκους αλφαριθμητικό όπου οι

χαρακτήρες του αποτελούνται από τα στοιχεία 0 και 1 [Hol75, Jong75].

Πίνακας 6.1 *Ο Γενετικός Αλγόριθμος*

```
Procedure Genetic Algorithm
begin
t = 1;
initialize Population(t);
evaluate Population(t);
while (not termination condition) do
begin
        t = t + 1;
        create Population(t) from Population(t-1);
        recombine members of Population(t)
        evaluate Population(t);
end;
end.
```

Εν συνεχεία αποτιμάται η αξία του κάθε μέλους του πληθυσμού σύμφωνα με την συνάρτηση ποιότητας. Αυτή η συνάρτηση εξαρτάται από το προς επίλυση πρόβλημα και μπορεί να είναι αρκετά απλή. Για παράδειγμα, σε ένα πρόβλημα βελτιστοποίησης μιας μαθηματικής συνάρτησης μιας μεταβλητής, η συνάρτηση ποιότητας μπορεί να είναι η μετατροπή της δυαδικής αναπαράστασης του κάθε μέλους του πληθυσμού στην δεκαδική μορφή. Όμως σε ένα πρόβλημα αναγνώρισης εικόνας, η συνάρτηση ποιότητας είναι μια αρκετά σύνθετη και περίπλοκη σχέση.

Η επιλογή των ατόμων που θα αποτελέσουν τον νέο πληθυσμό, ο οποίος θα χρησιμοποιηθεί για τη δημιουργία νέων μελών, συνήθως γίνεται βάση μιας συνάρτησης πιθανότητας η οποία λαμβάνει υπόψη της την αξία του κάθε ατόμου. Υπάρχουν αρκετές μεθοδολογίες επιλογής των νέων μελών αλλά όλες μοιράζονται την ίδια αρχή της *προτεραιότητας των καλύτερων μελών* [Gol91]. Στο νέο πληθυσμό που θα προκύψει μετά την επιλογή θα εφαρμοστούν οι γενετικοί μηχανισμοί με σκοπό τη δημιουργία νέων μελών. Η διασταύρωση και η μετάλλαξη αποτελούν τους πιο συνηθισμένους γενετικούς μηχανισμούς. Η εφαρμογή των γενετικών μηχανισμών σκοπό έχει από τον ήδη υπάρχοντα πληθυσμό να δημιουργηθούν απόγονοι οι οποίοι θα *είναι καλύτεροι* από τους προγόνους τους.

Η διασταύρωση υλοποιείται με την επιλογή δύο μελών του πληθυσμού και την

ανταλλαγή μεταξύ τους, όλου του υλικού πριν και μετά από μια τυχαία θέση στο σώμα της συμβολοσειράς που αναπαριστά μια λύση στο χώρο που ορίζει το πρόβλημα. Η μετάλλαξη υλοποιείται με την επιλογή ενός μέλους του πληθυσμού και την αλλαγή της τιμής, σε μια τυχαία θέση στο σώμα της συμβολοσειράς του μέλους. Η διασταύρωση στους γενετικούς αλγορίθμους είναι ο κύριος μηχανισμός εκτίμησης πολλών διαφορετικών λύσεων στο χώρο έρευνας που ορίζει το προς επίλυση πρόβλημα, ενώ η μετάλλαξη χρησιμοποιείται κυρίως για εισαγωγή μικρών αλλαγών στο *γενετικό υλικό* ενός πληθυσμού και την δοκιμή της αξίας αυτών των αλλαγών.

Η ανωτέρω λειτουργία, που αναφέρεται και σαν γενετικός κύκλος, επαναλαμβάνεται μέχρι την επίτευξη κάποιου κριτηρίου. Οι γενετικοί αλγόριθμοι έχουν χρησιμοποιηθεί με επιτυχία στην επίλυση αρκετά δύσκολων πραγματικών προβλημάτων και μερικοί από τους λόγους που τους κάνουν ελκυστικούς, είναι ότι : α) είναι απλή και εύκολη η υλοποίηση τους, β) είναι σχετικά εύκολη η κωδικοποίηση τους και η εισαγωγή και χρήση της γνώσης του πεδίου, γ) χρησιμοποιούν έναν πληθυσμό από σημεία κάτι που τους δίνει μια δυναμική και συνολική προσέγγιση στο πρόβλημα, ελαχιστοποιώντας την πιθανότητα παγίδευσης τους σε κάποια τοπική κορυφή του πεδίου έρευνας.

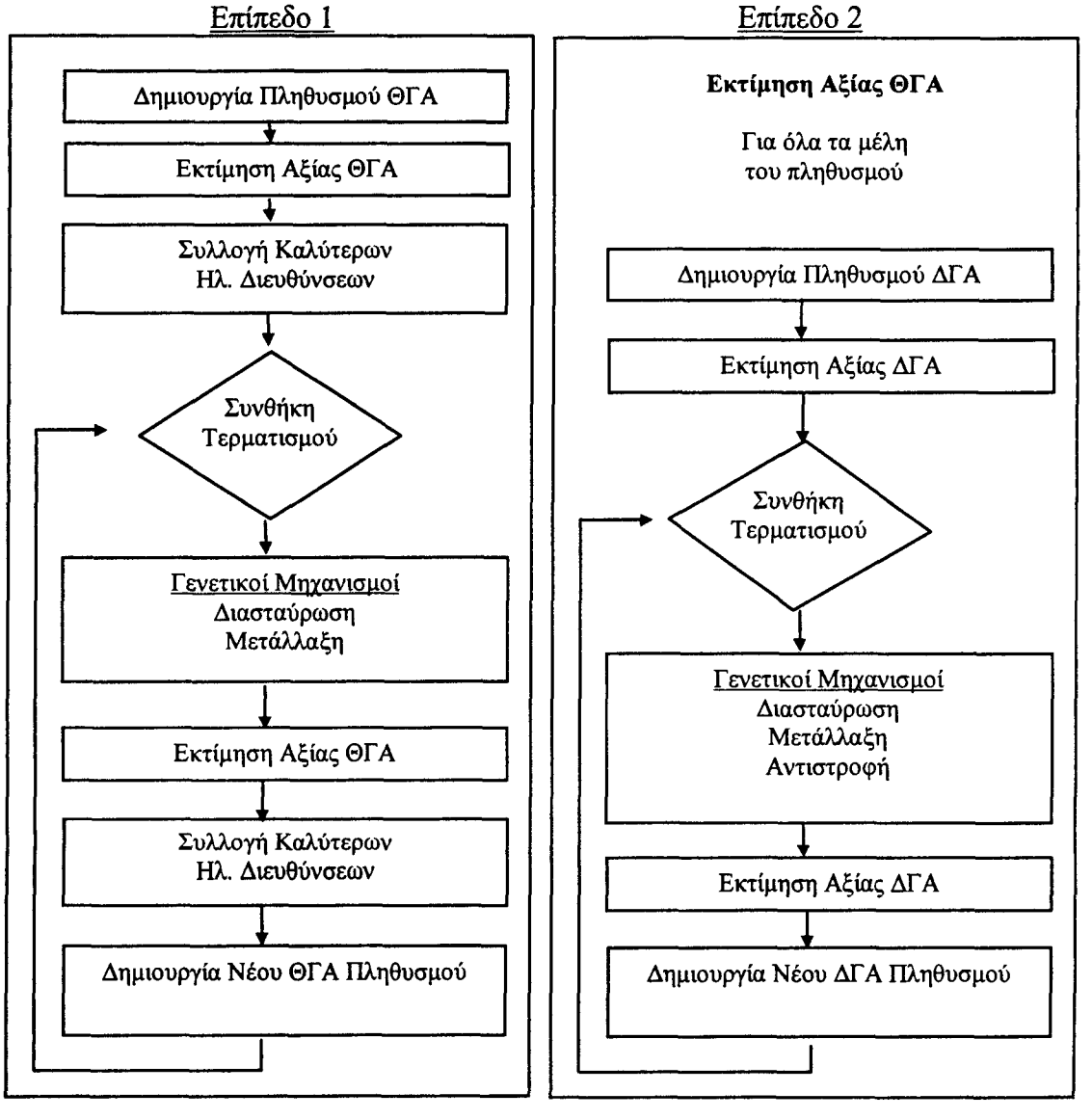
Λεπτομερής περιγραφή, σχόλια καθώς επίσης και διάφορα προβλήματα και η μεθοδολογία επίλυσης τους αντίστοιχα με την χρήση των ΓΑ, περιγράφεται από τους Goldberg [Gol89], David [Dav91] και Mitchell [Mit96]. Τέλος, η εργασία του Fogel [Fog97] περιγράφει με αρκετά συνοπτικό τρόπο τις ομοιότητες και τις διαφορές μεταξύ των διαφόρων εξελικτικών μεθόδων.

6.6.2 Λειτουργικότητα του Πράκτορα Γενετικός Αλγόριθμος

Ο ΠΓΑ χρησιμοποιεί τεχνικές που προέρχονται από το χώρο των γενετικών αλγορίθμων για να εξερευνήσει το πεδίο ενδιαφερόντων του χρήστη, να προσαρμοστεί σε αυτό και τέλος να προτείνει νέα θεματικά πεδία συνδυάζοντας την υπάρχουσα πληροφορία που είναι αποθηκευμένη στο προφίλ του χρήστη. Η όλη λειτουργία του ΠΓΑ, αναπαριστάται στο σχήμα 6.3, και είναι αρκετά σύνθετη αφού δύο γενετικοί αλγόριθμοι τρέχουν ταυτόχρονα με σκοπό την βέλτιστη αναπαράσταση των ενδιαφερόντων του χρήστη. Στην πραγματικότητα πρόκειται για έναν γενετικό αλγόριθμο δύο επιπέδων.

Στο πρώτο επίπεδο, ο Θεμελιώδης Γενετικός Αλγόριθμος - ΘΓΑ, δημιουργεί ένα

πληθυσμό από μέλη όπου το κάθε μέλος είναι ένα σύνολο από λέξεις. Η επιλογή των λέξεων που θα απαρτίζουν το κάθε μέλος του πληθυσμού γίνεται τυχαία μέσα από το *Λεξικό*. Ο μόνος περιορισμός που υπάρχει, είναι ότι η κάθε λέξη του συνόλου θα πρέπει να εμφανίζεται μόνο μια φορά. Ο αριθμός των μελών του πληθυσμού καθώς επίσης και ο αριθμός των λέξεων που θα περιλαμβάνει το κάθε μέλος είναι δύο από τις παραμέτρους του συστήματος και τις καθορίζει ο χρήστης του. Η δημιουργία του πληθυσμού του ΘΓΑ εμφανίζεται στο σχήμα 6.4.



***Σχήμα 6.3** Η λειτουργία του Πράκτορα Γενετικού Αλγορίθμου*

Όπως τονίσαμε και στις προηγούμενες ενότητες, όλες οι λέξεις που υπάρχουν στο λεξικό αποτελούν το προφίλ του χρήστη και κατ' επέκταση τον αρχικό χώρο έρευνας. Η λέξη αρχικός έχει να κάνει με το ότι ο χώρος που ορίζουν οι συγκεκριμένες λέξεις

θα αποτελέσει το πρώτο στάδιο έρευνας. Στην πραγματικότητα, δεν υπάρχει ένας συγκεκριμένος χώρος έρευνας αλλά πρόκειται για ένα ευμετάβλητο χώρο, όπου οι κορυφές αλλάζουν ύψη ανάλογα με την αξία που τους δίνει ο χρήστης του συστήματος και επιπλέον ολόκληρες περιοχές εμφανίζονται ή εξαφανίζονται ανάλογα με το ενδιαφέρον του χρήστη τη συγκεκριμένη χρονική περίοδο.

Σε αυτό το σημείο θα πρέπει να τονίσουμε ότι στη συγκεκριμένη εφαρμογή δεν υπάρχει λόγος να χρησιμοποιήσουμε άλλη κωδικοποίηση για την αναπαράσταση ενός μέλους του πληθυσμού. Θα μπορούσαμε αντί για τις λέξεις να χρησιμοποιούσαμε τον δείκτη στην αντίστοιχη θέση στο Λεξικό ή την κωδικοποίηση του ή δυαδικά αλφαριθμητικά, όπου το μήκος τους είναι ίσο με το μέγεθος του *Λεξικού*, και το 0 και το 1 στο σώμα της αναπαράστασης θα δηλώνουν την απουσία ή την παρουσία της αντίστοιχης λέξης. Πιστεύουμε ότι το αποτέλεσμα της χρήσης των παραπάνω και ίσως άλλων πιθανών αναπαραστάσεων θα είναι το ίδιο με μόνο μειονέκτημα την ταχύτητα της εφαρμογής.

Τυχαία σύνολα λέξεων

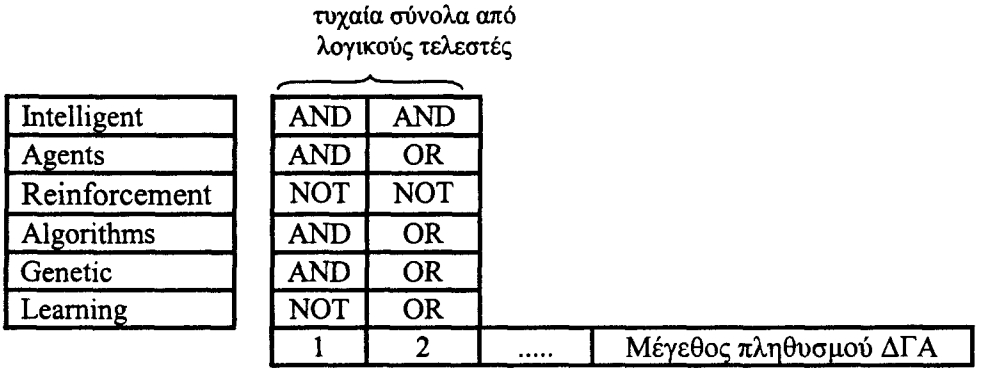
Intelligent	Server		
Agents	Java		
Reinforcement	Agents		
Algorithms	Genetic		
Genetic	Software		
Learning	Papers		
1	2	...	Μέγεθος πληθυσμού ΘΓΑ

ΠΡΟΦΙΛ ΧΡΗΣΤΗ		
ΛΕΞΕΙΣ	**ΚΕΙΜΕΝΑ**	**ΣΥΧΝΟΤΗΤΑ**
Intelligent	10	20
Agents	8	18
....	....	....
Learning	....	....
Java	....	....
Genetic	....	....
....	....	....
Algorithms	....	....
Server	....	....
Reinforcement	....	....
....	....	....

Σχήμα 6.4 *Η δημιουργία του Θεμελιώδους Γενετικού Αλγορίθμου*

Μετά την δημιουργία του αρχικού πληθυσμού, το επόμενο βήμα είναι να κάνουμε εκτίμηση της αξίας του κάθε μέλους. Σε αυτό το βήμα ένας δεύτερος γενετικός αλγόριθμος θα "*τρέξει*" για το κάθε μέλος του ΘΓΑ πληθυσμού. Ο Δευτερογενής Γενετικός Αλγόριθμος - ΔΓΑ, θα δημιουργήσει με την σειρά του, ένα πληθυσμό από δομές όπου η κάθε μια θα είναι ένα σύνολο από λογικούς τελεστές, επιλεγμένο τυχαία ανάμεσα από τους τελεστές "AND" "OR" and "NOT", σχήμα 6.5. Ο αριθμός των

μελών του ΔΓΑ πληθυσμού είναι μία από τις παραμέτρους που καθορίζει ο χρήστης του συστήματος. Ο αριθμός των λογικών τελεστών που θα περιλαμβάνει το σώμα του κάθε μέλους του ΔΓΑ, είναι ίσος με τον αριθμό των λέξεων που περιλαμβάνει το κάθε μέλος του ΘΓΑ.



Σχήμα 6.5 *Η δημιουργία του Δευτερογενούς Γενετικού Αλγορίθμου*

Μετά την δημιουργία και του ΔΓΑ, έχουμε ένα σύνολο από λέξεις οι οποίες εμφανίζονται στις προτιμήσεις του χρήστη και οι οποίες μπορεί να περιγράφουν μερικά από τα ενδιαφέροντα του, καθώς επίσης και έναν πληθυσμό από λογικούς τελεστές.

Ο συνδυασμός των λέξεων με τον πληθυσμό των λογικών τελεστών θα δημιουργήσει ένα πλήθος ερωτήσεων που θα εξυπηρετήσει ο Πράκτορας Μεταμηχανή Αναζήτησης Πληροφοριών. Οι δύο πρώτες ερωτήσεις που σχηματίζονται από τον συνδυασμό του πρώτου μέλους του ΘΓΑ με τα δύο πρώτα μέλη του ΔΓΑ είναι οι παρακάτω :

Α Ερώτηση	
AND	Intelligent, Agents, Algorithms, Genetic
OR	
NOT	Reinforcement, Learning

Β Ερώτηση	
AND	Intelligent
OR	Agents, Genetic, Algorithms, Learning
NOT	Reinforcement

Ο ΠΜΑΠ αφού παραλάβει τις ερωτήσεις από το ΠΓΑ θα τις προωθήσει με την σειρά του στις μηχανές αναζήτησης που χρησιμοποιούνται από την εφαρμογή. Οι ηλεκτρονικές διευθύνσεις των αποτελεσμάτων που θα συλλέξει ο ΠΜΑΠ θα επεξεργαστούν αν και μόνο αν δεν υπάρχουν στο Ιστορικό Αρχείο, όπου εκεί είναι

αποθηκευμένες οι διευθύνσεις πού έχει υποδείξει ο χρήστης σαν παραδείγματα των ενδιαφερόντων του. Ο λόγος για την συγκεκριμένη συμπεριφορά του συστήματος είναι ότι θα πρέπει να κερδίσουμε την εμπιστοσύνη του χρήστη και για αυτό δεν θα πρέπει να συμπεριλαμβάνονται στις υποδείξεις του συστήματος οι προτάσεις του χρήστη.

6.6.3 Λεπτομερής Σχεδιασμός του Πράκτορα Γενετικός Αλγόριθμος

Η κλάση *GeneticAlgorithm* [B21] που εκτελείται σαν ανεξάρτητη διεργασία - μέσω της υλοποίησης της διεπαφής *Runnable* [A2] πραγματοποιεί τις λειτουργίες του ΠΓΑ και αρχικοποιείται από την κλάση *LearningServer* [B17]. Οι προϋποθέσεις για την έναρξη λειτουργίας του ΠΓΑ, εκτός φυσικά από την συγκεκριμένη επιθυμία του χρήστη, είναι :

α) το *Λεξικό* να μην είναι άδειο, δηλαδή ο χρήστης είτε να έχει υποδείξει ορισμένα κείμενα σαν παραδείγματα.

β) να έχει συμπληρωθεί ο απαιτούμενος αριθμός κειμένων - ο συγκεκριμένος αριθμός των κειμένων αποτελεί μια από τις παραμέτρους του συστήματος που καθορίζει ο χρήστης και μπορεί να είναι μονάδα, δηλαδή από το πρώτο κείμενο.

γ) να μην υπάρχουν κείμενα είτε παραδείγματα του χρήστη είτε αξιολογήσεις του στις υποδείξεις του συστήματος που βρίσκονται σε αναμονή.

Αυτό σημαίνει ότι πριν αρχίσει η λειτουργία ή η επανενεργοποίηση του ΠΓΑ, όπως φαίνεται στο σχήμα 6.3, θα πρέπει να έχουν ενσωματωθεί στο προφίλ του χρήστη όλες οι προτάσεις του, είτε πρόκειται για νέα παραδείγματα είτε για τις αλληλοεπιδράσεις του με κόσμους εικονικής πραγματικότητας είτε για βαθμολογία του στις τρέχουσες υποδείξεις του συστήματος. Πιστεύουμε ότι η συμπεριφορά αυτή είναι επιθυμητή ειδικά σε περιπτώσεις όπου το σύστημα χρησιμοποιείται συνέχεια με αποτέλεσμα να υπάρχουν αρκετές αλληλοεπιδράσεις του χρήστη και κατά συνέπεια θα πρέπει το σύστημα να λαμβάνει υπόψη τις τρέχουσες απόψεις του χρήστη.

Η μέθοδος *ReadytoRun* ελέγχει τις παραπάνω συνθήκες εάν αληθεύουν και επιστρέφει μια τιμή *Αληθής* ή *Ψευδής* ανάλογα με το αποτέλεσμα της έρευνας. Το

αποτέλεσμα της μεθόδου *isDictionaryEmpty* [B21] επιστρέφει μια τιμή *Αληθής* ή *Ψευδής* ανάλογα με το αν το *Λεξικό* είναι άδειο ή όχι - πρακτικά αν ο χρήστης έχει υποδείξει κάποια κείμενα ή όχι. Εάν υπάρχουν νέες προτάσεις του χρήστη ή υπάρχουν βαθμολογίες του στις υποδείξεις του συστήματος θα επεξεργαστούν αντίστοιχα από τις μεθόδους *ProcessMessages* και *ProcessEvaluations*. Και οι δύο μέθοδοι θα χρησιμοποιήσουν τον ANKE για να συλλέξουν τις λέξεις και τις συχνότητες εμφάνισης τους στα αντίστοιχα κείμενα και στην συνέχεια και οι δύο θα τροποποιήσουν το *Λεξικό,* η μεν *ProcessMessages* με την βοήθεια της μεθόδου *AddEntries* [B26] ενώ η *ProcessEvaluations* με την μέθοδο *ModifyDictionary* [B26].

Όταν η μέθοδος *ReadytoRun* επιστρέψει μια τιμή *Αληθής,* τότε αρχίζει και η λειτουργία του ΠΓΑ με την μέθοδο *InitPGAPopulation*, η οποία θα δημιουργήσει έναν πληθυσμό από λέξεις. Μια προκαθορισμένη επανάληψη γεμίζει τις θέσεις του *Vector PGAPopulation* με *Vectors* από λέξεις οι οποίες είναι επιλεγμένες τυχαία από το προφίλ του χρήστη, με την βοήθεια της μεθόδου *randomWord*. Για την δημιουργία ενός συνόλου από λέξεις χρησιμοποιείται μια άλλη επανάληψη, η οποία αφού επιλέξει μια τυχαία λέξη από το *Λεξικό,* εν συνεχεία γίνεται έλεγχος αν η λέξη είναι ήδη μέλος του συνόλου. Στην περίπτωση που είναι μέλος τότε η λειτουργία επαναλαμβάνεται μέχρι του σημείου όπου το σύνολο να αποτελείται από διαφορετικές μεταξύ τους λέξεις. Η ίδια λέξη μπορεί να υπάρχει ή όχι στα άλλα σύνολα λέξεων αλλά εκείνο που μας ενδιαφέρει είναι να μην υπάρχει σε ένα σύνολο μια λέξη περισσότερες από μία φορές.

Με το πέρας της δημιουργίας του ΘΓΑ πληθυσμού εν συνεχεία θα γίνει εκτίμηση της αξίας του κάθε μέλους του. Η μέθοδος *PopulationFittness* με την βοήθεια ενός δεύτερου γενετικού αλγορίθμου θα αποδώσει μια τιμή η οποία αντιπροσωπεύει την αξία του κάθε μέλους. Στο σώμα της μεθόδου, η λειτουργία θα επαναληφθεί τόσες φορές όσα είναι τα μέλη του ΘΓΑ πληθυσμού, και το κάθε σύνολο από λέξεις θα αποτελέσει μια από τις παραμέτρους του δομητή της κλάσης *SGA* [B32]. Οι άλλες δύο παράμετροι είναι : α) το *Λεξικό* και β) ένα στιγμιότυπο της κλάσης *BadHistoryServer* [B20] η οποία έχει τις ίδιες μεθόδους και την ίδια λειτουργία με το ΙΣΑΡ, μόνο που εδώ αποθηκεύονται όλες οι διευθύνσεις των υποδείξεων του συστήματος που ο χρήστης τις έχει βαθμολογήσει αρνητικά, δηλαδή όσες δεν του άρεσαν. Στο σώμα της μεθόδου *runSGA* υλοποιείται όλη η λειτουργία του ΔΓΑ όπως αυτή απεικονίζεται στο σχήμα 6.3.

Αρχικά θα δημιουργηθεί ένας πληθυσμός από σύνολα λογικών τελεστών. Στο

σώμα της μεθόδου *InitSGAPopulation* μια προκαθορισμένη επανάληψη θα γεμίσει την κάθε θέση του *Vector Population* με ένα σύνολο από λογικούς τελεστές, όπως το AND, OR και NOT. Εν συνεχεία η μέθοδος *SGAPopulFittnes* θα υπολογίσει την αξία του κάθε μέλους του ΔΓΑ πληθυσμού. Έτσι λοιπόν το σύνολο των λέξεων μαζί με το κάθε σύνολο από λογικούς τελεστές δημιουργεί και μια διαφορετική ερώτηση. Όλες τις ερωτήσεις θα τις εξυπηρετήσει η μέθοδος *CreateQuery,* με την βοήθεια του ΠΜΑΠ. Η *CreateQuery* επιστρέφει ένα *Hashtable,* ο οποίος έχει σαν κλειδιά τα ονόματα των μηχανών αναζήτησης, που χρησιμοποιήθηκαν κατά τη διάρκεια της έρευνας, και σαν τιμές έχει *Vectors* που περιέχουν τα αποτελέσματα της αντίστοιχης μηχανής. Όμως η μια ερώτηση προς τις μηχανές αναζήτησης δεν δημιουργεί ένα και μοναδικό αποτέλεσμα αλλά ένα σύνολο από ηλεκτρονικές διευθύνσεις.

Όλα τα κείμενα που αναφέρονται στις διευθύνσεις των αποτελεσμάτων θα ανακτηθούν και εν συνεχεία η ομοιότητα του κάθε ενός θα συγκριθεί με το προφίλ του χρήστη. Η αξία του συνόλου των λογικών τελεστών που δημιούργησε την συγκεκριμένη ερώτηση είναι η μεγαλύτερη τιμή ομοιότητας των κειμένων - τα αποτελέσματα της ερώτησης - με το προφίλ του χρήστη. Αυτό πιστεύουμε ότι είναι *"δίκαιο"* γιατί κάθε μηχανή αναζήτησης χρησιμοποιεί την δική της μεθοδολογία βαθμολόγησης των κειμένων. Δεν έχει σημασία αν ένα *"καλό κείμενο"* βρίσκεται στην τελευταία θέση των αποτελεσμάτων γιατί ο σκοπός είναι να βρούμε καλούς συνδυασμούς λογικών τελεστών οι οποίοι με την σειρά τους θα φέρουν καλά αποτελέσματα.

Καθ'όλη την διάρκεια λειτουργίας του ΔΓΑ κάθε μέλος του πληθυσμού θα σχηματίσει μια ερωτήση, η οποία με την σειρά της θα αποφέρει αρκετά αποτελέσματα. Από όλα τα αποτελέσματα όλων των ερωτήσεων, οι διευθύνσεις των καλύτερων κειμένων, οι τιμές ομοιότητας τους με το προφίλ του χρήστη καθώς επίσης και οι λογικοί τελεστές που συμμετείχαν για το σχηματισμό της συγκεκριμένης ερώτησης αποθηκεύονται αντίστοιχα στους τρεις *Vectors FinalLinks, FinalVals* και *FinalOper* - το μέγεθος των οποίων είναι μια παράμετρος που καθορίζει ο χρήστης.

Η αξία του συνόλου των λέξεων που μετέχουν στην ερώτηση είναι ο μέσος όρος των τιμών ομοιότητας που εμπεριέχονται στο *Vector FinalVals.* Και αυτό πιστεύουμε ότι είναι *"σωστό"* γιατί μετράμε την συνολική αξία του συγκεκριμένου συνόλου των λέξεων για ένα αρκετά μεγάλο πλήθος ερωτήσεων που μπορούν να σχηματιστούν από τις συγκεκριμένες λέξεις.

Κατά την διάρκεια λειτουργίας του ΘΓΑ, τα σύνολα των λέξεων και τα σύνολα των λογικών τελεστών που μετέχουν στις ερωτήσεις, οι τιμές ομοιότητας με το προφίλ του χρήστη καθώς επίσης και οι ηλεκτρονικές διευθύνσεις των καλύτερων κειμένων αποθηκεύονται αντίστοιχα στους *Vectors* ΄ *FinalWords, FinalOpers, FinalVals* και *FinalUrls* [B21]. Με το πέρας της λειτουργίας του ΘΓΑ, τα περιεχόμενα αυτών των *Vectors* θα αποτελούν και τις υποδείξεις του συστήματος προς το χρήστη.

Πριν προχωρήσουμε στην περιγραφή των γενετικών μηχανισμών θα πρέπει να τονίσουμε ότι όλο αυτό το σύνολο των ερωτήσεων προς τις μηχανές αναζήτησης, η ανάκτηση των ηλεκτρονικών διευθύνσεων και εν συνεχεία των κειμένων, δημιουργεί μεγάλο φόρτο εργασίας στο τοπικό δίκτυο και μια γενικότερη καθυστέρηση στην εμφάνιση των υποδείξεων του συστήματος.

Επίσης, παρατηρήσαμε ότι κατά την διάρκεια της λειτουργίας του ΔΓΑ υπάρχει περίπτωση να σχηματιστούν όμοια σύνολα από λογικούς τελεστές, τα οποία φυσικά θα φέρουν τα ίδια αποτελέσματα ή διαφορετικοί συνδυασμοί λογικών τελεστών να φέρουν μερικά αποτελέσματα τα οποία έχουμε εκτιμήσει σε προηγούμενο βήμα.

Για την επίλυση των παραπάνω προβλημάτων αλλά και για την αύξηση της ταχύτητας του συστήματος, όλοι οι συνδυασμοί των λογικών τελεστών αποθηκεύονται στο *Hashtable MemOper* και οι διευθύνσεις που έχουμε ελέγξει αποθηκεύονται στο *Hashtable SGAMemory.*

6.6.4 Λεπτομερής Σχεδιασμός των Γενετικών Μηχανισμών

Μετά την εκτίμηση του κάθε μέλους του ΔΓΑ πληθυσμού θα εφαρμόσουμε σε όλο το πληθυσμό, τους γενετικούς μηχανισμούς Διασταύρωση, Μετάλλαξη και Αντιστροφή με σκοπό να διαφοροποιήσουμε τις ερωτήσεις που θα απευθύνουμε στο ΠΜΑΠ.

Η Διασταύρωση είναι ο κύριος μηχανισμός που χρησιμοποιείται στους γενετικούς αλγορίθμους γιατί μας επιτρέπει να εξετάζουμε πολλές λύσεις στο χώρο έρευνας του προβλήματος. Η λειτουργία του μηχανισμού είναι η επιλογή δύο μελών του πληθυσμού και η ανταλλαγή των λογικών τελεστών μεταξύ τους από ένα τυχαία επιλεγμένο σημείο της δομής μέχρι το τέλος της. Το σχήμα 6.6 παρουσιάζει δύο τυχαία μέλη (A,B) του ΔΓΑ πληθυσμού που διασταυρώνονται στην τρίτη θέση και δημιουργούνται δύο νέα μέλη A' και B'. Η ιδέα στο μηχανισμό της διασταύρωσης είναι να δημιουργηθούν νέα μέλη στον πληθυσμό με καλύτερες ιδιότητες και αυτό

επιτυγχάνεται αναδιατάσσοντας το υλικό που εμπεριέχεται στις δομές δύο παλαιών μελών.

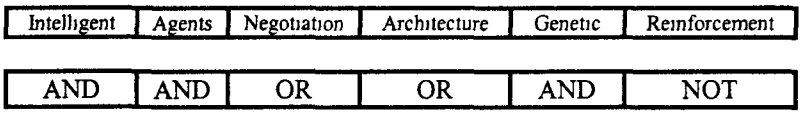
Ερώτηση A: AND (Genetic) OR (Agents, Negotiation, Architecture) NOT (Intelligent, Reinforcement)

A	NOT	OR	OR	OR	AND	NOT
B	AND	AND	OR	NOT	NOT	AND

Ερώτηση B: AND (Intelligent, Agents, Reinforcement) OR (Negotiation,) NOT (Architecture, Genetic)

A'	NOT	OR	OR	NOT	NOT	AND

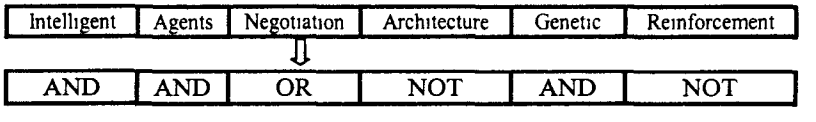
Ερώτηση A': AND (Reinforcement) OR (Agents, Negotiation) NOT (Intelligent,Architecture,Genetic)



Ερώτηση B': AND (Intelligent, Agents, Genetic) OR (Negotiation, Architecture) NOT (Reinforcement)

Σχήμα 6.6 *Η λειτουργία του μηχανισμού της Διασταύρωσης για το ΔΓΑ*

Η Μετάλλαξη είναι ο επόμενος γενετικός μηχανισμός ο οποίος εφαρμόζεται σε ένα μόνο μέλος του πληθυσμού. Πιο συγκεκριμένα, επιλέγεται ένα τυχαίο μέλος του ΔΓΑ πληθυσμού και στη συνέχεια σε ένα τυχαίο σημείο της δομής του διαφοροποιείται. Για παράδειγμα, αν ο λογικός τελεστής στο σημείο μετάλλαξης είναι OR τότε αυτός θα αλλάξει είτε σε AND ή NOT.



Ερώτηση A: AND(Intelligent, Agents,Genetic) OR(Negotiation) NOT (Reinforcement, Architecture)

A'	AND	AND	AND	NOT	AND	NOT

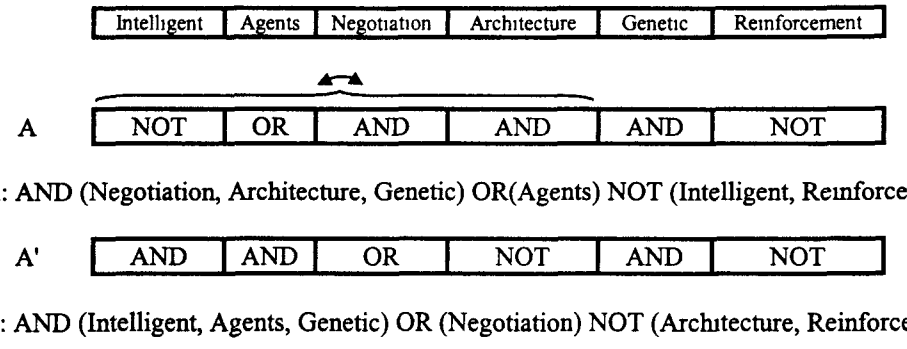
Ερώτηση A': AND(Intelligent, Agents,Genetic, Negotiation) NOT (Reinforcement, Architecture)

Σχήμα 6.7 *Η λειτουργία του μηχανισμού της Μετάλλαξης για το ΔΓΑ*

Η Μετάλλαξη θεωρείται βοηθητικός μηχανισμός και ο στόχος είναι η εισαγωγή μικρών αλλαγών στο πληθυσμό με σκοπό την ενσωμάτωση τους ή την απόρριψη τους. Επίσης ο συγκεκριμένος μηχανισμός επαναφέρει τμήμα της δομής που έχει

χαθεί ή εισάγει μια μικρή αλλαγή στη δομή, η οποία δεν μπορεί να επιτευχθεί από την λειτουργία του μηχανισμού της Διασταύρωσης

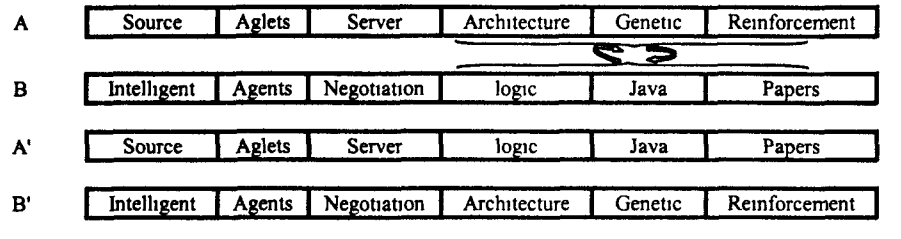
Για την δημιουργία νέων ερωτήσεων χρησιμοποιούμε επιπλέον τον γενετικό μηχανισμό της Αντιστροφής, ο οποίος εφαρμόζεται σε ένα μέλος του πληθυσμού και αντιστρέφει τη σειρά των λογικών τελεστών από την αρχή της δομής μέχρι ενός τυχαία επιλεγμένου σημείου. Το σχήμα 6.8 παρουσιάζει την λειτουργία του μηχανισμού της Αντιστροφής.



Σχήμα 6.8 *Η λειτουργία του μηχανισμού της Αντιστροφής για το ΔΓΑ*

Θα πρέπει να τονίσουμε ότι ο μηχανισμός της αντιστροφής δεν απαντάται στην φύση αλλά είναι ένας μηχανισμός αναδιάταξης της δομής ενός μέλους του πληθυσμού. Στην συγκεκριμένη εφαρμογή χρησιμοποιείται επειδή τα στοιχεία που αποτελούν μια συγκεκριμένη δομή δεν είναι ανεξάρτητα της θέσης τους μέσα στην δομή. Οπότε, είναι ένας επιπλέον μηχανισμός σχηματισμού νέων ερωτήσεων.

Οι μέθοδοι *SGACrossOver*, *SGAMutation* και *SGAInversion* υλοποιούν αντίστοιχα τους μηχανισμούς της Διασταύρωσης, της Μετάλλαξης και τέλος της Αντιστροφής. Το ποσοστό επί του συνόλου του πληθυσμού που θα εφαρμοστούν οι συγκεκριμένοι γενετικοί μηχανισμοί είναι παράμετροι του συστήματος τις οποίες ελέγχει ο χρήστης.



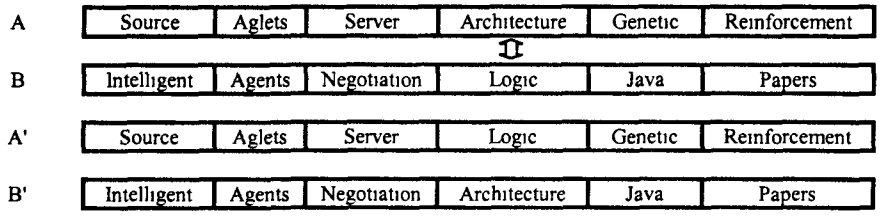
Σχήμα 6.9 *Η λειτουργία του μηχανισμού της Διασταύρωσης για το ΘΓΑ*

Με το πέρας της εφαρμογής των γενετικών μηχανισμών θα γίνει εκτίμηση της αξίας

όλων των μελών του ΔΓΑ πληθυσμού, όπως περιγράφεται παραπάνω. Αφού όλα τα μέλη έχουν μια τιμή που αντιπροσωπεύει την αξία τους, στην συνέχεια θα επιλεγούν τα καλύτερα μέλη του πληθυσμού ώστε να δημιουργήσουν ένα νέο πληθυσμό, ο οποίος με την σειρά του θα δώσει καλύτερους απογόνους.

Η μέθοδος *CreateNewSGAPopulation* υλοποιεί την λειτουργία επιλογής των καλύτερων μελών του πληθυσμού. Η όλη διαδικασία διέπεται από την αρχή της *αξίας των καλύτερων μελών* και ομοιάζει με μεγάλη ρουλέτα όπου η κάθε θέση καταλαμβάνεται από την αντίστοιχη αξία ενός μέλους του πληθυσμού. Όσο μεγαλύτερη αξία έχει ένα μέλος τόσο μεγαλύτερο χώρο καταλαμβάνει στην ρουλέτα και κατά συνέπεια έχει μεγαλύτερη πιθανότητα να επιλεγεί για να συμμετέχει στο νέο πληθυσμό.

Ο ίδιος κύκλος λειτουργίας με τους γενετικούς μηχανισμούς, την εκτίμηση του πληθυσμού και την δημιουργία νέου πληθυσμού με την συμμετοχή των καλύτερων μελών θα συνεχιστεί για μερικές προκαθορισμένες επαναλήψεις. Ο αριθμός των επαναλήψεων καθορίζεται από το χρήστη και το πέρας τους θεωρείται συνθήκη τερματισμού για το ΔΓΑ.



Σχήμα 6.10 *Η λειτουργία του μηχανισμού της Μετάλλαξης για το ΘΓΑ*

Με το πέρας της εκτίμησης των μελών του ΘΓΑ οι ίδιοι γενετικοί μηχανισμοί εκτός από την Αντιστροφή θα εφαρμοστούν ελαφρώς διαφοροποιημένοι στα σύνολα των λέξεων. Ο μηχανισμός της διασταύρωσης θα επιλέξει τυχαία δύο μέλη του ΘΓΑ πληθυσμού και θα ανταλλάξει όλες τις λέξεις μεταξύ τους από ένα τυχαία επιλεγμένο σημείο της δομής μέχρι το τέλος της, αν και μόνο αν καμία από τις λέξεις που αντιμετατίθενται δεν εμπεριέχεται περισσότερες από μια φορές σε κάποιο από τα δύο σύνολα. Στο σχήμα 6.9 απεικονίζεται ο μηχανισμός της Διασταύρωσης για δύο μέλη του ΘΓΑ με το σημείο διασταύρωσης να βρίσκεται στην τρίτη θέση της δομής.

Για το μηχανισμό της Μετάλλαξης έχουμε αναπτύξει δύο διαφορετικούς αλγορίθμους. Ο πρώτος επιλέγει δύο μέλη και ανταλλάσσει τις λέξεις τους σε ένα τυχαία επιλεγμένο σημείο, αν και μόνο αν καμία από τις λέξεις δεν εμφανίζεται

περισσότερες από μια φορές σε κάποιο σύνολο. Το σχήμα 6.10 απεικονίζει την συγκεκριμένη υλοποίηση του μηχανισμού της Μετάλλαξης. Ο δεύτερος αλγόριθμος, επιλέγει τυχαία ένα μέλος του πληθυσμού και διαγράφει μια λέξη από το σύνολο του και στην θέση της εισάγει μια νέα λέξη από το *Λεξικό* διατηρώντας τον ανωτέρω περιορισμό.

Με το πέρας της εφαρμογής των γενετικών μηχανισμών θα γίνει εκτίμηση της αξίας όλων των μελών του ΘΓΑ πληθυσμού. Κατά παρόμοιο τρόπο με το ΔΓΑ πληθυσμό, θα επιλεγούν από το ΘΓΑ τα καλύτερα μέλη του πληθυσμού, ώστε να δημιουργήσουν ένα νέο πληθυσμό από σύνολα λέξεων, τα οποία με την σειρά τους θα δώσει καλύτερους συνδυασμούς λέξεων. Οι μέθοδοι *PGACrossOver*, *PGAMutation* και *CreateNewPopulation* [B21] υλοποιούν για το ΘΓΑ αντίστοιχα τους μηχανισμούς της Διασταύρωσης, Μετάλλαξης και δημιουργίας νέου πληθυσμού.

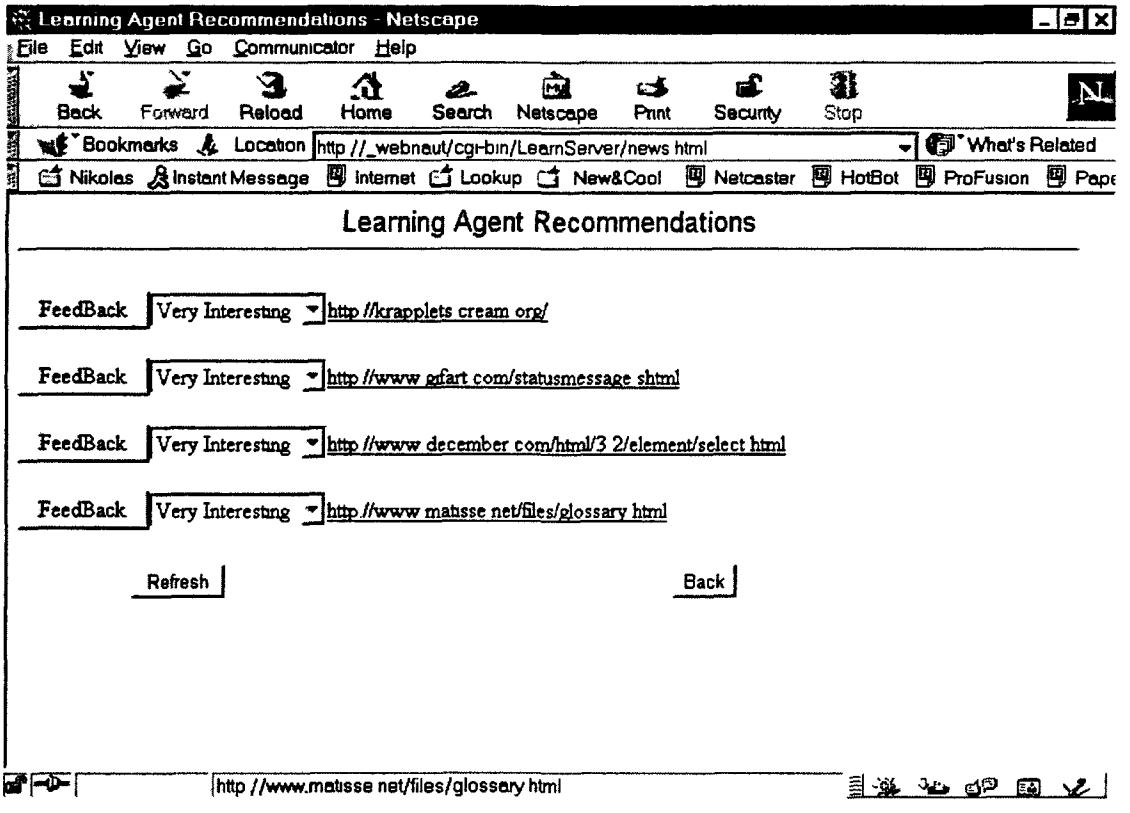
Ο ίδιος κύκλος λειτουργίας με τους γενετικούς μηχανισμούς, την εκτίμηση των μελών και την δημιουργία νέου πληθυσμού με την συμμετοχή των καλύτερων μελών θα συνεχιστεί για μερικές προκαθορισμένες επαναλήψεις, ο αριθμός των οποίων καθορίζεται από το χρήστη. Με το πέρας των επαναλήψεων ο ΠΓΑ θα έχει γεμίσει τους *Vectors FinalWords, FinalOpers, FinalVals* και *FinalUrls* [B21], τα περιεχόμενα των οποίων αποτελούν και τις υποδείξεις του συστήματος προς το χρήστη.

6.7 Η Ανάδραση του Χρήστη

Με το πέρας της ανωτέρω λειτουργίας και πριν ο ΠΓΑ τεθεί σε κατάσταση αναμονής περιμένοντας την ανάδραση του χρήστη στις υποδείξεις του, θα δημιουργήσει ένα στιγμιότυπο της κλάσης *ResultsGA* [B31] το οποίο αποτελεί και την πρόσβαση του ΠΕ στα αποτελέσματα της έρευνας του ΠΓΑ. Η μέθοδος *addEntries* θα δημιουργήσει αντίγραφα όλων των αποτελεσμάτων της έρευνας τα οποία και θα αποθηκεύσει στους στατικούς τοπικούς *Vectors Links, Words* και *Operands* [B31], τα περιεχόμενα των οποίων είναι αντίστοιχα οι ηλεκτρονικές διευθύνσεις, οι λέξεις και οι λογικοί τελεστές, των υποδείξεων του συστήματος.

Η κλάση *ResultsGA* υλοποιεί μεθόδους όπως την *saveFile*, για την αποθήκευση των αποτελεσμάτων σε τοπικό αρχείο με σκοπό την αποφυγή της καταστροφής των δεδομένων σε περίπτωση εξόδου από την εφαρμογή, και την *formatHtml*, η οποία μορφοποιεί και παρουσιάζει τα αποτελέσματα σε μορφή Html ώστε η κάθε υπόδειξη

να συνοδεύεται και από την βαθμολογία που επιτρέπεται να χρησιμοποιήσει ο χρήστης για να εκφράσει την άποψη του. Το σχήμα 6.11 εμφανίζει μερικές υποδείξεις του συστήματος όπως αυτές εμφανίζονται μορφοποιημένες σε γλώσσα Html στο πρόγραμμα πλοήγησης του χρήστη.

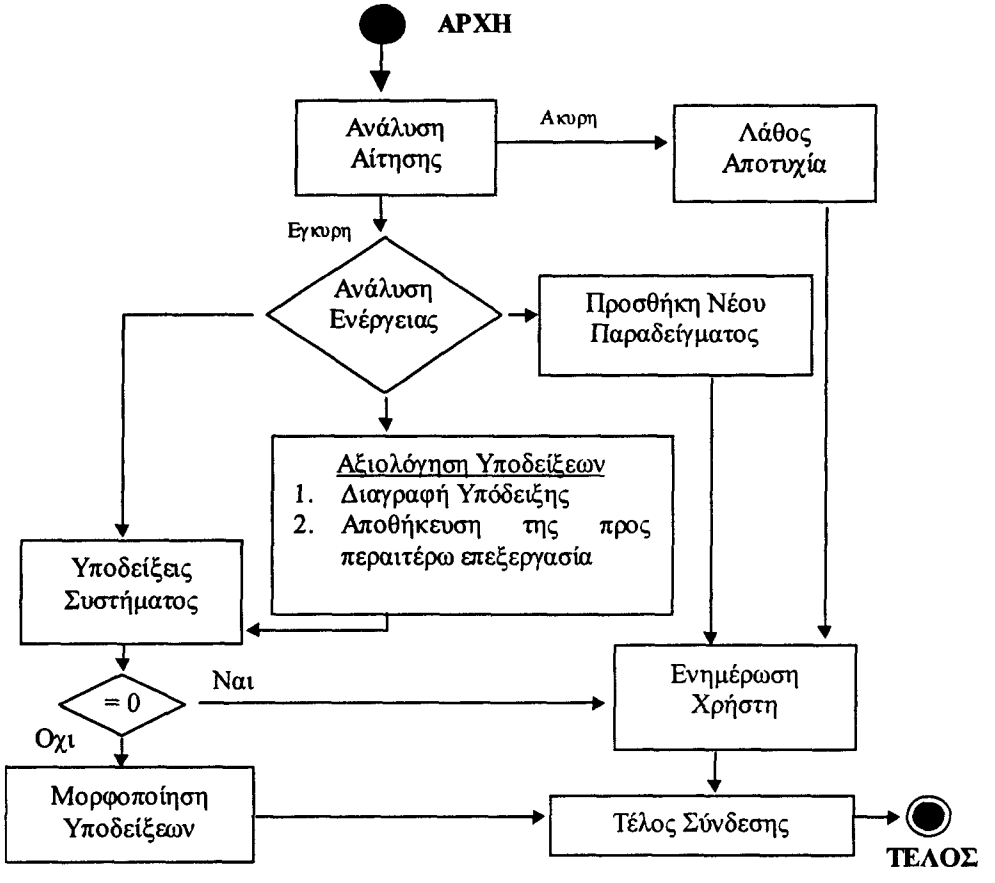


Σχήμα 6.11 *Η διεπαφή του ΠΕ για την εμφάνιση των υποδείξεων του συστήματος.*

Ο χρήστης του συστήματος μπορεί οποιαδήποτε στιγμή να ζητήσει τις υποδείξεις του συστήματος. Η συγκεκριμένη λειτουργία ενεργοποιείται με ένα πάτημα στο αντίστοιχο εικονίδιο της ΠΚΕ. Στο σχήμα 6.12 εμφανίζεται το διάγραμμα δραστηριοτήτων του ΠΕ κατά την αλληλοεπίδραση του με το χρήστη. Η κάθε αλληλοεπίδραση του χρήστη με το ΠΕ εξυπηρετείται από την μέθοδο *userFeedBack*, η οποία ενεργοποιείται από το ΠΔΔ, έχοντας πλήρη πρόσβαση στην αίτηση και στην απόκριση του χρήστη.

Μέσα στο σώμα της μεθόδου γίνεται έλεγχος της αίτησης του χρήστη και αν πρόκειται για την ενημέρωση του συστήματος με ένα ακόμα παράδειγμα των ενδιαφερόντων του χρήστη τότε η μέθοδος *addEntry* [B17] το αποθηκεύει στο τοπικό στατικό *Vector Messages*. Επιπλέον στο σώμα της μεθόδου υλοποιείται η λειτουργία της αποθήκευσης σε ένα τοπικό αρχείο, για την αποφυγή της καταστροφής των

δεδομένων λόγου εξόδου από το σύστημα, όλων των παραδειγμάτων που προτείνει ο χρήστης.



Σχήμα 6.12 *Το διάγραμμα δραστηριοτήτων του ΠΕ κατά την αλληλεπίδραση με το χρήστη*

Αφού ολοκληρωθεί η εργασία της αποθήκευσης στην συνέχεια ο χρήστης ανακατευθύνεται σε μια τοπική στατική σελίδα, η οποία εξυπηρετείται από το ΠΕΔΚ. Με το πάτημα στο κουμπί ενημέρωσης που βρίσκεται στην ΠΚΕ (δες σχήμα 6.1) δημιουργείται ένα νέο αυτοαναδυόμενο παράθυρο το οποίο εμφανίζει την απόκριση του εξυπηρετητή. Έτσι και ο χρήστης ενημερώνεται ότι η ενέργεια του έγινε αντιληπτή από το σύστημα αλλά και δεν χάνει την σελίδα που βρίσκεται στο ενδιαφέρον του.

Η μέθοδος *getEntry* [B17] χρησιμοποιείται από το ΠΓΑ για να διαβάσει ένα από τα παραδείγματα που έχει προτείνει ο χρήστης. Στο σώμα αυτής της μεθόδου γίνεται η αντίστροφη εργασία, δηλαδή, απομακρύνεται από το *Vector Messages* η διεύθυνση που είναι αποθηκευμένη στην πρώτη θέση και εν συνεχεία επαναγράφονται όλα τα υπόλοιπα δεδομένα στο προκαθορισμένο τοπικό αρχείο.

Αν η αίτηση του χρήστη αναφέρεται στην επίδειξη των υποδείξεων του

συστήματος τότε θα γίνει αρχικά έλεγχος αν υπάρχουν προτάσεις του συστήματος προς το χρήστη και αν αυτό είναι αληθές τότε θα μορφοποιηθούν σε γλώσσα Html και στην συνέχεια θα αποσταλούν στο χρήστη. Στην περίπτωση που δεν υπάρχουν υποδείξεις τότε θα γίνει έλεγχος αν ο ΠΓΑ είναι σε λειτουργία και αν αυτό είναι αληθές τότε ο χρήστης ενημερώνεται ότι η όλη λειτουργία βρίσκεται σε εξέλιξη. Διαφορετικά ενημερώνεται ότι ο ΠΓΑ δεν βρίσκεται σε λειτουργία και παρέχονται οδηγίες για την ενεργοποίηση του. Όλες οι οδηγίες προς το χρήστη όπως ακόμα και τα μηνύματα που αναφέρονται σε σφάλματα είναι στατικές σελίδες Html οι οποίες εξυπηρετούνται από το ΠΕΔΚ.

Όπως περιγράψαμε παραπάνω, με το πέρας της λειτουργίας του ΠΓΑ όλες οι διευθύνσεις που έχουν επιλεγεί σαν πιο σχετικές με την πληροφορία που είναι αποθηκευμένη στο *Λεξικό*, θα παρουσιαστούν στον χρήστη ο οποίος θα παρέχει την κρίση του σε μια βαθμολογημένη κλίμακα όπως εμφανίζεται στο σχήμα 6.11. Η κλίμακα της βαθμολογίας που έχει στην διάθεση του ο χρήστης εμφανίζεται στον παρακάτω πίνακα :

Πίνακας 6.2 *Η κλίμακα βαθμολογίας στις υποδείξεις του συστήματος*

Βαθμολογία Χρήστη	**c**
Πολύ Ενδιαφέρον	2
Ενδιαφέρον	1
Αδιάφορο	0
Άσχετο	-1
Πολύ Άσχετο	-2

Όπως φαίνεται και στο σχήμα 6.11 δίπλα από την κάθε υπόδειξη υπάρχει ένα κουτί επιλογών που περιέχει τις βαθμολογίες του πίνακα 6.2. Ο χρήστης θα ελέγξει μία προς μία τις υποδείξεις του συστήματος και εν συνεχεία θα επιλέξει για το κάθε κείμενο την βαθμολογία που πιστεύει ότι το χαρακτηρίζει. Τέλος θα πατήσει το πλήκτρο *Feedback* και θα αποστείλει την βαθμολογία του για την συγκεκριμένη υπόδειξη.

Ο ΠΕ αφού παραλάβει την αξιολόγηση του χρήστη στην συνέχεια θα καλέσει την μέθοδο *addEntry* της κλάσης *EvaluationGA* [B18] για να αποθηκεύσει την βαθμολογία του. Η συγκεκριμένη κλάση χρησιμοποιείται σαν *"μέσο επικοινωνίας"* μεταξύ του ΠΕ και του ΠΓΑ ώστε να συλλέγονται οι βαθμολογίες του χρήστη μέχρι να τις επεξεργαστεί ο ΠΓΑ. Το πρώτο στιγμιότυπο της κλάσης *EvaluationGA* αρχικοποιείται στο σώμα της μεθόδου *Initialize* της κλάσης *LearningServer* [B17].

Η κάθε βαθμολογία του χρήστη, η αντίστοιχη ηλεκτρονική διεύθυνση, οι λέξεις που σχημάτισαν την ερώτηση για την συγκεκριμένη υπόδειξη καθώς επίσης και οι λογικοί τελεστές της καταλαμβάνουν τις θέσεις των αντίστοιχων στατικών *Vectors Score, Links, Words* και *Operands* κλάσης *EvaluationGA.*

Πίνακας 6.3 *Η κλίμακα βαθμολογίας για τις λέξεις που μετέχουν στην ερώτηση*

Βαθμολογία Χρήστη	**c**		
	AND	**OR**	**NOT**
Πολύ Ενδιαφέρον	2	1	-2
Ενδιαφέρον	1	1	-1
Αδιάφορο	0	0	0
Άσχετο	-1	-1	0
Πολύ Άσχετο	-2	-1	0

Όπως και οι άλλες κλάσεις που διαχειρίζονται χρήσιμα δεδομένα έτσι και η *EvaluationGA* υλοποιεί την μέθοδο *SaveFile* για την αποθήκευση των περιεχομένων των *Vectors* σε ένα προκαθορισμένο αρχείο και αντίστοιχα την μέθοδο *openfile* για την ενημέρωση των *Vectors* κατά την έναρξη της εφαρμογής. Ο ΠΓΑ θα καλέσει την μέθοδο *size* [B18]η οποία επιστρέφει το αριθμό των στοιχείων που είναι αποθηκευμένα στο *Vector Links.* Στο σώμα της μεθόδου *ProcessEvaluations* [B21], οι βοηθητικές μέθοδοι *getLinkAt, getWordsAt, getScoreAt, getOperandsAt* [B18] προσφέρουν πρόσβαση σε μια συγκεκριμένη θέση στους αντίστοιχους *Vectors.*

Ο ΠΓΑ αρχικά θα ανακτήσει με την βοήθεια ANKE, σε δύο *Hashtables,* τις λέξεις που περιέχονται στο κείμενο που δείχνει η διεύθυνση της υπόδειξης καθώς και τις συχνότητες εμφάνισης τους. Στην συνέχεια, ο ΠΓΑ θα ενημερώσει το *Λεξικό* για την αξιολόγηση του χρήστη στην συγκεκριμένη υπόδειξη. Το έργο αυτό επιτελείται στο σώμα της μεθόδου *ModifyDictionary* [B26] η οποία θα τροποποιήσει τις συχνότητες εμφάνισης των λέξεων που είναι κοινές ανάμεσα στο λεξικό και στην υπόδειξη σύμφωνα με την παρακάτω σχέση :

$$Tf_D = Tf_D + ((\frac{c}{100}) * Tf_R) \qquad \textbf{(6.4)}$$ όπου

TF_D - είναι η συχνότητα εμφάνισης της λέξης στο Λεξικό

TF_R - είναι η συχνότητα εμφάνισης της λέξης στην υπόδειξη του συστήματος

c - είναι η βαθμολογία του χρήστη, από -2 έως 2, όπως αυτή εμφανίζεται στον πίνακα 6.2

Ο ΠΓΑ χρησιμοποιεί γνωστές μηχανές αναζήτησης για την συλλογή νέας πληροφορίας από το διαδίκτυο και γνωρίζει ποίος συγκεκριμένος συνδυασμός λέξεων και λογικών τελεστών έφερε σαν αποτέλεσμα την τρέχουσα υπόδειξη του συστήματος. Ετσι λοιπόν κατά την αξιολόγηση εφαρμόζουμε τον ανωτέρω τύπο, όχι μόνο στις κοινές λέξεις όπως είπαμε παραπάνω, αλλά επιπλέον και στις λέξεις που συνθέτουν την συγκεκριμένη ερώτηση. Στην περίπτωση αυτή, το c παίρνει τιμή όπως εμφανίζεται στον πίνακα 6.3.Κατ' αυτόν τον τρόπο μειώνουμε περισσότερο την αξία των συγκεκριμένων λέξεων με σκοπό την γρήγορη προσαρμογή του προφίλ του χρήστη στις ανάγκες του και στα πραγματικά ενδιαφέροντα του.

6.8 Συμπεράσματα

Στο κεφάλαιο αυτό παρουσιάσαμε την λειτουργία της προσαρμοζόμενης εκμάθησης των ενδιαφερόντων του χρήστη. Περιγράψαμε αναλυτικά την μεθοδολογία της δημιουργίας του προφίλ του χρήστη και εν συνεχεία την λειτουργία της εκμάθησης και προσαρμογής στα ενδιαφέροντα του χρήστη, με την χρήση ενός γενετικού αλγορίθμου δύο επιπέδων.

Το σύστημα Webnaut χρησιμοποιώντας την ανωτέρω μεθοδολογία επιτυγχάνει να δημιουργήσει σύνθετες δομές αναπαράστασης των μακροχρόνιων ενδιαφερόντων του χρήστη και να συλλέξει νέα κείμενα από το διαδίκτυο, τα οποία αποτελούν και τις υποδείξεις του συστήματος.

Τέλος, με την λειτουργία της ανάδρασης του χρήστη στις υποδείξεις του συστήματος, επιταχύνουμε την προσαρμογή του προφίλ του χρήστη στις ανάγκες και στα πραγματικά ενδιαφέροντα του.

Κεφάλαιο 7

ΑΞΙΟΛΟΓΗΣΗ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ

7.1 Εισαγωγή

Σε αυτό το κεφάλαιο θα παρουσιάσουμε τα πειραματικά αποτελέσματα τα οποία αποκομίσαμε με τη χρήση του συστήματος Webnaut. Τα συγκεκριμένα στοιχεία προέκυψαν κατά την διάρκεια δύο πειραμάτων, τα οποία είχαν σαν στόχο την συλλογή σελίδων του παγκοσμίου ιστού και εν συνεχεία το φιλτράρισμα των πληροφοριών, σύμφωνα με προκαθορισμένα προφίλ χρηστών.

Επίσης, τα συγκεκριμένα πειράματα σκοπό είχαν να διερευνήσουμε δύο σημαντικές όψεις του συστήματος, πρώτον την ακρίβεια των υποδείξεων του και δεύτερον την ικανότητα του να προσαρμόζεται στα ενδιαφέροντα του χρήστη ακόμη και στην περίπτωση που δεν διαθέτει γι' αυτά, από την αρχή, αρκετές πληροφορίες. Και στις δύο περιπτώσεις θα βγάλουμε χρήσιμα συμπεράσματα για την επίδραση της ανάδρασης του χρήστη στις υποδείξεις του συστήματος.

7.2 Πρώτο Πείραμα - Αναζήτηση των Ενδιαφερόντων του Χρήστη

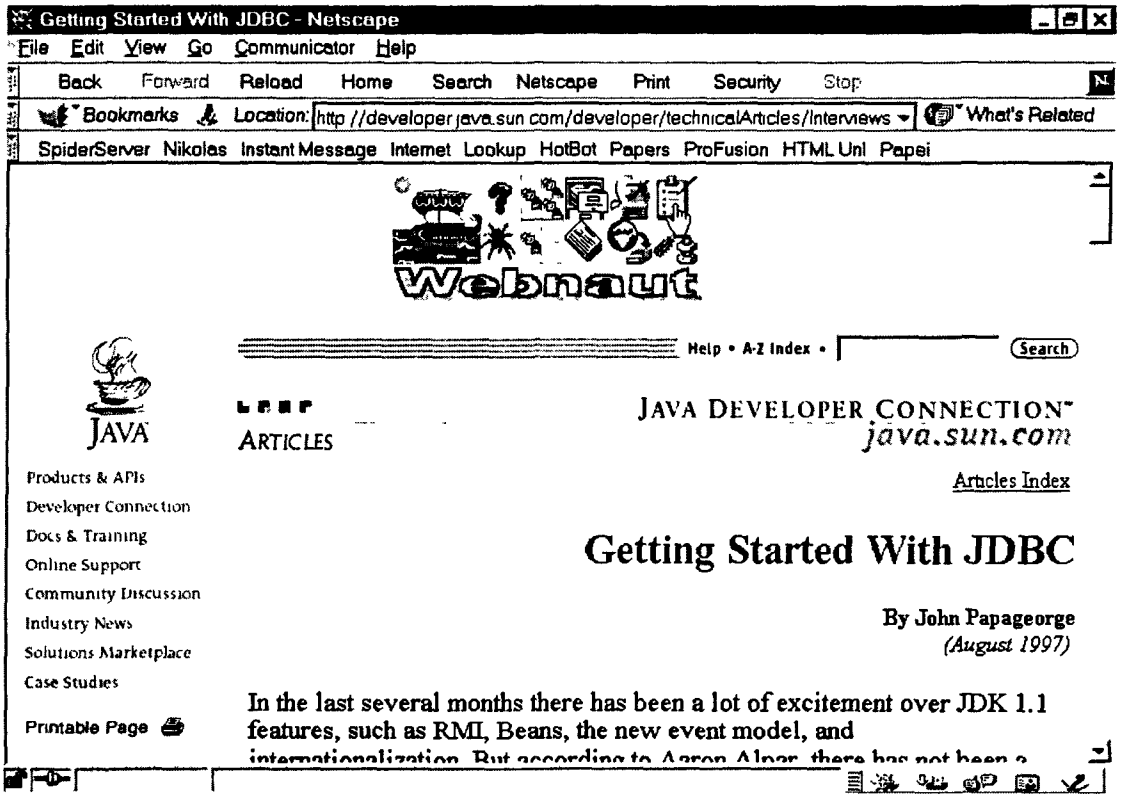
Ο σκοπός στο πρώτο πείραμα είναι να ελέγξουμε κατ' αρχήν αν το σύστημα είναι σε θέση να καταλάβει τα ενδιαφέροντα του χρήστη στην περίπτωση που αυτός ενδιαφέρεται για ένα και μόνο θέμα. Επίσης έχει ενδιαφέρον να μελετήσουμε πόσο γρήγορα προσαρμόζεται ο πράκτορας εκμάθησης στα ενδιαφέροντα του χρήστη. Πιό συγκεκριμένα, θεωρούμε ότι ένας χρήστης ενδιαφέρεται για το προγραμματισμό σε γλώσσα Java και έχει υποδείξει με την βοήθεια του συστήματος, ορισμένα παραδείγματα σχετικά με το συγκεκριμένο θέμα.

Ο προγραμματισμός με την βοήθεια της γλώσσας Java έχει πρακτική αξία ανάμεσα σε φοιτητές και προγραμματιστές, επαγγελματίες ή μή, που ενδιαφέρονται για τις γλώσσες προγραμματισμού και γενικότερα για τον αντικειμενοστραφή προγραμματισμό αλλά και για τις νέες τεχνολογίες στο διαδίκτυο. Θέλαμε το αντικείμενο της έρευνας να έχει και πρακτική αξία για το χρήστη ώστε να είναι

πρόθυμος καθ' όλη την διάρκεια του πειράματος.

7.2.1 Προετοιμασία του Πρώτου Πειράματος

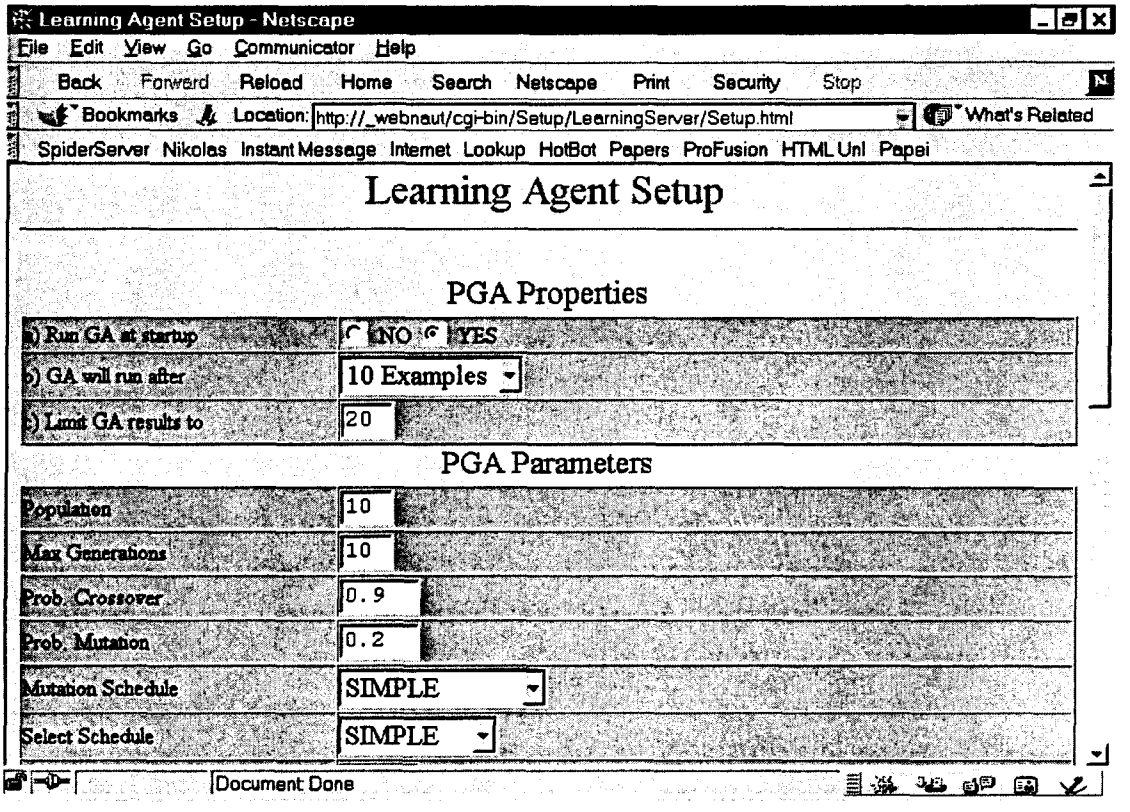
Πριν από την έναρξη του πειράματος επισκεφθήκαμε τρεις γνωστούς κόμβους του διαδικτύου - Java Developer [Γ46] , JavaWorld [Γ47], Developer.com [Γ48] - οι οποίοι περιέχουν εκπαιδευτικές σελίδες, σεμινάρια, συμβουλές σχετικές με την γλώσσα Java. Ξεχωρίσαμε αυτούς τους τρεις δικτυακούς τόπους λόγω του ότι τα θέματα που παρουσιάζουν και ο τρόπος με τον οποίο τα παρουσιάζουν θα τραβούσε το ενδιαφέρον του οποιουδήποτε ενδιαφέρετε για την γλώσσα Java.



Σχήμα 7.1 Μια εκπαιδευτική σελίδα στο κόμβο του Java Developer

Το σχήμα 7.1 παρουσιάζει την διαδικασία της συλλογής των ηλεκτρονικών διευθύνσεων με την βοήθεια του προγράμματος, που θα χρησιμοποιηθούν κατά την διάρκεια του πειράματος. Στο συγκεκριμένο σχήμα εμφανίζεται ο δικτυακός τόπος του Java Developer [Γ46] ο οποίος παρουσιάζει ένα άρθρο σχετικό με την χρήση της διεπαφής JDBC [Γ9.a] η οποία αποτελεί τον καθιερωμένο τρόπο για τη σύνδεση και αλληλοεπίδραση με μια βάση δεδομένων, πάντα με την βοήθεια της γλώσσας Java.

Πιστεύουμε ότι η συγκεκριμένη σελίδα και το θέμα που διαπραγματεύεται θα ενδιέφερε έναν οποιοδήποτε χρήστη είτε από πλευράς τεχνολογίας για τις δυνατότητες και τους περιορισμούς που παρέχει η συγκεκριμένη διεπαφή είτε από πλευράς προγραμματισμού, τόσο στο πρακτικό μέρος δηλαδή στη χρήση των εντολών όσο και στο θεωρητικό μέρος δηλαδή το στυλ, την παρουσίαση και την ανάπτυξη του κώδικα.



Σχήμα 7.2 Η διεπαφή των παραμέτρων λειτουργίας του Πράκτορα Εκμάθησης

Σε ανάλογο ύφος και τρόπο παρουσίασης είναι και οι υπόλοιπες σελίδες που επιλέχθηκαν για να σχηματίσουν το αρχικό προφίλ των χρηστών που θα μετέχουν στο πρώτο πείραμα. Επίσης, όλες οι σελίδες που θα χρησιμοποιηθούν σε αυτό το πείραμα είναι ένας συνδυασμός, από θεωρία και πράξη ώστε να προσομοιώσουμε καλύτερα τα πραγματικά ενδιαφέροντα ενός χρήστη.

Το σύνολο των ηλεκτρονικών διευθύνσεων που συγκεντρώσαμε και από τους τρεις δικτυακούς τόπους ήταν 40. Αυτές οι διευθύνσεις δεν επεξεργάστηκαν αλλά χωρίστηκαν με τυχαίο τρόπο σε 4 πακέτα, όπου το κάθε ένα από αυτά αριθμούσε 10 μοναδιαίες διευθύνσεις. Το κάθε ένα από αυτά τα τέσσερα πακέτα θα αποτελέσει το σύνολο των παραδειγμάτων που θα χρησιμοποιηθούν από τον Πράκτορα Εκμάθησης,

ώστε να σχηματίσει το αρχικό προφίλ για το κάθε ένα από τους 4 χρήστες που θα συμμετέχουν στο πείραμα.

Επίσης οι ρυθμίσεις του προγράμματος είναι ακριβώς οι ίδιες για τον καθένα χρήστη. Το σχήμα 7.2 παρουσιάζει την διεπαφή του προγράμματος Webnaut για την αλλαγή των παραμέτρων λειτουργίας του Πράκτορα Εκμάθησης. Στο συγκεκριμένο σχήμα εμφανίζεται το αρχικό όριο των παραδειγμάτων που θα πρέπει να έχει συγκεντρώσει ο χρήστης πριν τεθεί σε λειτουργία ο Πράκτορας Γενετικός Αλγόριθμος. Το όριο αυτό ανέρχεται στις 10 σελίδες, όσες ακριβώς είναι και οι ηλεκτρονικές διευθύνσεις που αποτελούν το κάθε πακέτο που έχουμε δημιουργήσει στο προηγούμενο στάδιο.

Πίνακας 7.1 *Οι παράμετροι λειτουργίας του Πράκτορα Γενετικού Αλγορίθμου*

	ΘΓΑ	**ΔΓΑ**
Μέγεθος Πληθυσμού	10	20
Μέγεθος Συνόλου (λέξεων και λογικών τελεστών)	6	6
Κύκλοι Λειτουργίας (Γενεές)	10	10
Ποσοστό Διασταύρωσης	0,9	0,9
Ποσοστό Μετάλλαξης	0,2	0,2
Ποσοστό Αντιστροφής	-	0,4

Στις παραμέτρους λειτουργίας περιλαμβάνεται και ο αριθμός των διευθύνσεων που θα εμφανίζεται στον χρήστη κατά το στάδιο της αξιολόγησης των υποδείξεων του συστήματος. Ο αριθμός αυτός περιορίζεται στις 20 ηλεκτρονικές διευθύνσεις και πιστεύουμε ότι είναι ικανοποιητικός τόσο από την πλευρά της εξαγωγής χρήσιμων συμπερασμάτων όσο και από την πλευρά της μη καταπόνησης του χρήστη κατά την διάρκεια της βαθμολόγησης.

Στον πίνακα 7.1 συνοψίζονται οι παράμετροι λειτουργίας των δύο γενετικών αλγορίθμων του ΠΓΑ. Ένας πληθυσμός από 10 μέλη για το ΘΓΑ θα δοκιμάσει αρκετούς συνδυασμούς λέξεων από το *Λεξικό*, το οποίο αποτελεί και το προφίλ του χρήστη. Εν συνεχεία, για το κάθε μέλος του ΘΓΑ θα σχηματιστεί ένας πληθυσμός από 20 μέλη, όπου το κάθε μέλος θα είναι ένα σύνολο από λογικούς τελεστές. Τα 20 σύνολα των λογικών τελεστών θα αποτελέσουν τον πληθυσμό του ΔΓΑ.

Πιστεύουμε ότι οι αριθμοί των δύο πληθυσμών είναι ικανοποιητικοί ώστε να δημιουργήσουν αρκετούς συνδυασμούς λέξεων με λογικούς τελεστές με αποτέλεσμα να έχει το σύστημα αρκετά καλή εικόνα του χώρου έρευνας των ενδιαφερόντων του

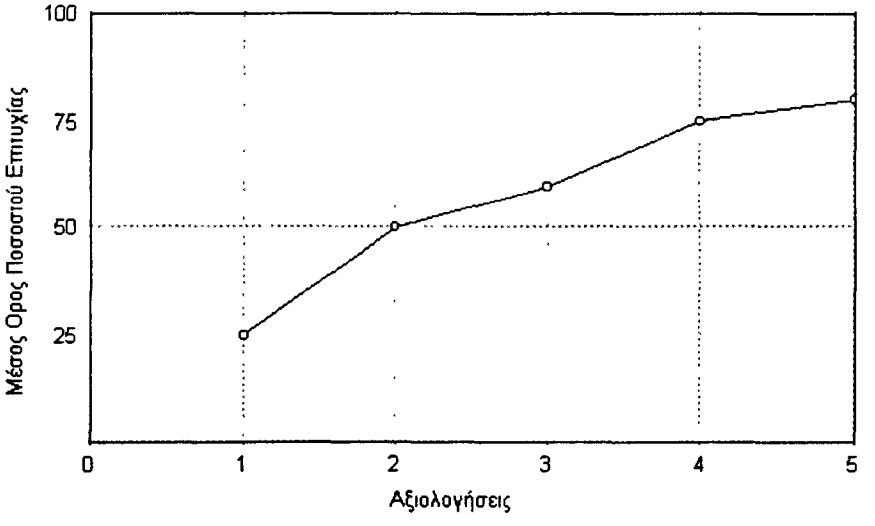
χρήστη.

Οι κύκλοι λειτουργίας του ΘΓΑ και ΔΓΑ είναι ίδιοι και έχουν οριστεί σε 10 ώστε να δώσουν τον απαραίτητο χρόνο στο σύστημα να ανακαλύψει το πεδίο ενδιαφερόντων του χρήστη. Τέλος στον πίνακα 7.1 εμφανίζονται τα ποσοστά επί του συνόλου του πληθυσμού της λειτουργίας των μηχανισμού της *Διασταύρωσης*, *Μετάλλαξης* και *Αντιστροφής*, αντίστοιχα για τον κάθε γενετικό αλγόριθμο.

7.2.2 Εκτέλεση του Πρώτου Πειράματος

Στο πρώτο πείραμα συμμετείχε μια ομάδα από 4 χρήστες οι οποίοι χρησιμοποίησαν το σύστημα Webnaut, ρυθμισμένο με τις ίδιες παραμέτρους για όλους τους χρήστες και το προφίλ του κάθε χρήστη δημιουργήθηκε αρχικά από ένα σύνολο από δέκα προεπιλεγμένες ηλεκτρονικές διευθύνσεις όπως αναφέραμε στην προηγούμενη ενότητα.

Η μόνη εργασία που είχαν να κάνουν οι χρήστες ήταν να παρέχουν βαθμολογία στις υποδείξεις του συστήματος μέσω της διεπαφής όπως αυτή εμφανίζεται στο σχήμα 6.11. Πιο συγκεκριμένα, με το πέρας του κύκλου λειτουργίας του ΘΓΑ, ο Πράκτορας Εκμάθησης έχει συγκεντρώσει μια λίστα από 20 ηλεκτρονικές διευθύνσεις οι οποίες αποτελούν και τις υποδείξεις του συστήματος προς τον χρήστη.



Σχήμα 7.3 Μέσος όρος του ποσοστού επιτυχίας των υποδείξεων του συστήματος στην κάθε αξιολόγηση.

Ο κάθε χρήστης επισκέπτεται ξεχωριστά την κάθε υπόδειξη και εν συνεχεία δίνει την

κρίση του για κάθε μια από αυτές μέσω της βαθμολογημένης κλίμακας η οποία περιλαμβάνει τις τιμές *Πολύ Ενδιαφέρον, Ενδιαφέρον, Αδιάφορο, Άσχετο, Πολύ Άσχετο.*

Ο Πράκτορας Εκμάθησης με το πέρας της κάθε αξιολόγησης και αφού ελάμβανε υπόψη του τις βαθμολογίες του χρήστη τροποποιώντας την αξία των λέξεων του προφίλ που μετείχαν στην κάθε υπόδειξη, εν συνεχεία άρχιζε την λειτουργία του ΠΓΑ με σκοπό να συγκεντρώσει νέες καλύτερες υποδείξεις. Θα πρέπει στο σημείο αυτό να κάνουμε δύο επισημάνσεις :

α) κατά την διάρκεια του πειράματος οι μόνες αλλαγές που συνέβαιναν στα προφίλ των χρηστών ήταν από τις βαθμολογίες τους στις υποδείξεις του συστήματος και όχι από τον εμπλουτισμό του προφίλ τους με νέες ηλεκτρονικές διευθύνσεις των πιθανών ενδιαφερόντων τους. Ο λόγος είναι ότι, από την εμπειρία μας γνωρίζουμε ότι αρκετοί χρήστες αλλάζουν τους σκοπούς τους κατά την διάρκεια της πλοήγησης καθώς ανακαλύπτουν νέα ενδιαφέροντα θέματα τα οποία δεν συμπεριλαμβάνονταν στις αρχικές τους προθέσεις.

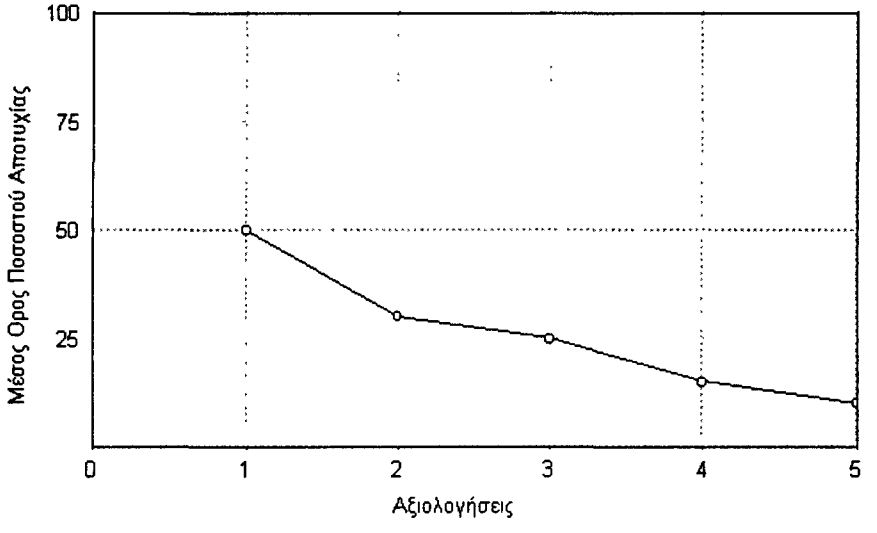
Μια τέτοια συμπεριφορά από την πλευρά των χρηστών κάνει την διαδικασία της εκμάθησης αρκετά δύσκολη και η προσαρμογή ενός έξυπνου συστήματος ίσως απαιτήσει περισσότερο χρόνο. Έτσι λοιπόν δεν επιτρεπόταν στους χρήστες κατά την διάρκεια του πειράματος να παρέχουν νέα παραδείγματα από σελίδες που τους ενδιαφέρουν.

β) Ένα ποσοστό των σελίδων που βαθμολόγησαν οι χρήστες τις έκριναν σαν *Αδιάφορες*, πράγμα που σημαίνει ότι ναι μεν η συγκεκριμένη υπόδειξη έχει σχέση με το θέμα άλλα δεν αξίζει να χαρακτηριστεί σαν αντιπροσωπευτικό δείγμα. Στο συγκεκριμένο πείραμα σαν επιτυχία θεωρούμε κάθε βαθμολογία του χρήστη η οποία έχει σαν τιμή έναν από τους δύο χαρακτηρισμούς *Πολύ Ενδιαφέρον* ή *Ενδιαφέρον* και αντίστοιχα θεωρούμε σαν αποτυχία τους δύο χαρακτηρισμούς *Άσχετο και Πολύ Άσχετο.*

Το σχήμα 7.3 παρουσιάζει το μέσο όρο του ποσοστού επιτυχίας για όλους τους χρήστες σε κάθε αξιολόγηση του συστήματος. Από την πρώτη αξιολόγηση φαίνεται ότι ο Πράκτορας Εκμάθησης έχει ικανοποιητική απόδοση, αφού κατά μέσο όρο 5 υποδείξεις του συστήματος κρίθηκαν από τους χρήστες σαν *Ενδιαφέρουσες* ή *Πολύ*

Ενδιαφέρουσες.

Στην δεύτερη αξιολόγηση το ποσοστό αυτό διπλασιάζεται για να φτάσει στο πέρας του πειράματος στο ποσοστό του 80% επί του συνόλου των υποδείξεων. Άρα τα συμπεράσματα από το πρώτο πείραμα είναι ότι ο ΠΕ έχει την ικανότητα να ανακαλύψει τα ενδιαφέροντα του χρήστη και η ανάδραση αυτού παίζει σημαντικό ρόλο στην όλη απόδοση του συστήματος. Εξ' άλλου και στο σχήμα 7.4 εμφανίζεται η μείωση του ποσοστού αποτυχίας του συστήματος καθώς περνά ο χρόνος και το πρόγραμμα αλλάζει την αξία των λέξεων που μετέχουν στο προφίλ του χρήστη.



Σχήμα 7.4 *Μέσος όρος του ποσοστού αποτυχίας των υποδείξεων του συστήματος στην κάθε αξιολόγηση.*

Πάντως ένα μικρό ποσοστό αποτυχίας υπάρχει και στο τέλος της πέμπτης αξιολόγησης. Πιστεύουμε ότι το ποσοστό αυτό οφείλεται σε δύο λόγους : α) ότι οι χρήστες είχαν σαν αντικειμενικό σκοπό να βαθμολογούν επιτυχώς μόνο τις υποδείξεις που είχαν μεγάλη σχέση με το θέμα της έρευνας, β) εκ κατασκευής δεν υπάρχει ένα κατώφλι υποδείξεων, δηλαδή, το σύστημα ναι μεν θα ψάξει για να εντοπίσει τις 20 καλύτερες σελίδες του διαδικτύου, οι οποίες είναι πιο σχετικές με το προφίλ του χρήστη, αλλά βασίζεται πάντα στα αποτελέσματα των μηχανών αναζήτησης. Έτσι λοιπόν το σύστημα πάντα θα υποδεικνύει ένα πλήθος διευθύνσεων ακόμα και στην περίπτωση που έχουν μικρή ομοιότητα με το προφίλ του χρήστη.

Πιστεύουμε ότι το μικρό ποσοστό αποτυχίας δεν είναι κατ' ανάγκη κακό από την άποψη ότι οι συγκεκριμένες υποδείξεις μπορεί να είχαν - πάντα κατά την γνώμη των χρηστών - μικρή σχέση με το θέμα της έρευνας αλλά είχαν μια μικρή η μεγάλη σχέση

με το προφίλ του χρήστη. Αυτό σημαίνει ότι αν μπορούσαμε να δούμε χωροταξικά το πεδίο έρευνας που δημιουργεί το προφίλ του κάθε χρήστη, το οποίο δημιουργήθηκε από τα αρχικά παραδείγματα, θα βλέπαμε ότι υπάρχει μια μεγάλη κορυφή η οποία είναι ο προγραμματισμός στην γλώσσα Java. Όμως υπάρχουν και άλλες κορυφές οι οποίες είναι μεν μικρότερες αλλά είναι και αυτές μέσα στο χώρο έρευνας. Έτσι λοιπόν το πρόγραμμα εντοπίζει και προτείνει και αυτά τα θέματα και μάλιστα σε αρκετές περιπτώσεις και συνδυασμό τους.

7.3 Δεύτερο Πείραμα - Προσαρμογή στα Ενδιαφέροντα του Χρήστη

Ο σκοπός στο δεύτερο πείραμα είναι να ελέγξουμε αν ο σύστημα είναι σε θέση να καταλάβει και να προσαρμοστεί στα ενδιαφέροντα του χρήστη στην περίπτωση που δεν έχει αρκετές πληροφορίες για αυτά στο αρχικό προφίλ. Πιο συγκεκριμένα, θεωρούμε ότι ένας χρήστης ενδιαφέρεται πάλι για τον προγραμματισμό σε γλώσσα Java και έχει υποδείξει, με την βοήθεια του συστήματος, ορισμένα παραδείγματα σχετικά με το συγκεκριμένο θέμα. Όμως το αρχικό προφίλ του χρήστη περιέχει και έναν αριθμό παραδειγμάτων από άλλα θέματα, όπως η μουσική και ο αθλητισμός.

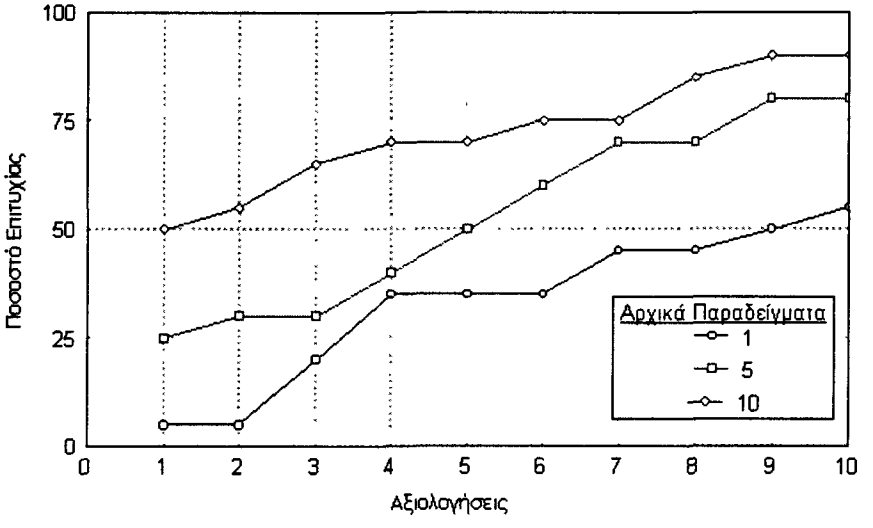
7.3.1 Προετοιμασία του Δεύτερου Πειράματος

Πριν από την έναρξη του πειράματος επισκεφθήκαμε επισκεφτήκαμε τρεις γνωστούς κόμβους του διαδικτύου - Rolling Stone Magazine [Γ49], MTV [Γ50], All Music Guide [Γ51] - οι οποίοι περιέχουν δισκογραφίες για διάφορα είδη μουσικής, βιογραφίες καλλιτεχνών και μουσικών συνόλων, κριτικές δίσκων κ.λπ. Συνολικά επιλέξαμε 15 μοναδιαίες διευθύνσεις οι οποίες σχετίζονται με την μουσική. Επιπλέον από τρεις άλλους κόμβους του διαδικτύου - USATODAY Sports [52], NBA [53], Sports.com [54] - συγκεντρώσαμε ακόμα 15 μοναδιαίες διευθύνσεις οι οποίες σχετίζονται με τον αθλητισμό. Ξεχωρίσαμε τους παραπάνω δικτυακούς τόπους όχι γιατί είναι οι καλύτεροι αλλά γιατί είναι αρκετά αντιπροσωπευτικοί στα συγκεκριμένα θέματα. Έτσι λοιπόν μερικές από τις 40 διευθύνσεις που είναι σχετικές με την Java και επιλέχτηκαν κατά την προετοιμασία του πρώτου πειράματος, μαζί με μερικές από τις σελίδες που επιλέχτηκαν με την παραπάνω διαδικασία και είναι σχετικές με την μουσική και τον αθλητισμό θα αποτελέσουν το αρχικό προφίλ για

τους χρήστες που θα συμμετάσχουν στο δεύτερο πείραμα.

7.3.2 Εκτέλεση του Δεύτερου Πειράματος

Στο δεύτερο πείραμα συμμετείχε μια άλλη ομάδα αποτελούμενη από 3 χρήστες. Ο σκοπός, οι παράμετροι και οι περιορισμοί ήταν ακριβώς οι ίδιοι με το πρώτο πείραμα. Η μόνη διαφοροποίηση ήταν ο αριθμός των αρχικών παραδειγμάτων που είχε το προφίλ του κάθε χρήστη και είχαν σχέση με το θέμα της έρευνας δηλαδή τον προγραμματισμό με την γλώσσα Java.



Σχήμα 7.5 *Το ποσοστό επιτυχίας των υποδείξεων του συστήματος στην κάθε αξιολόγηση και για το κάθε χρήστη.*

Ένα σύνολο από 15 σελίδες χρησιμοποιήθηκαν για την δημιουργία του αρχικού προφίλ, και από αυτές τις σελίδες σχετικές με το θέμα της έρευνας ήταν για τον πρώτο χρήστη μόνο μία σελίδα, για το δεύτερο χρήστη μόνο πέντε σελίδες και για τον τρίτο χρήστη ήταν οι δέκα σελίδες. Οι υπόλοιπες σελίδες - εκτός από τις σχετικές με το θέμα της έρευνας - για τον κάθε ένα χρήστη επιλέχτηκαν εξίσου από τα δύο άλλα θέματα με τυχαία σειρά. Έτσι λοιπόν οι 15 διευθύνσεις του αρχικού προφίλ για το δεύτερο χρήστη, μοιράστηκαν εξίσου στα τρία θέματα.

Το σχήμα 7.5 εμφανίζει τις επιτυχημένες υποδείξεις του ΠΕ για τον κάθε χρήστη σε κάθε αξιολόγηση. Είναι συναρπαστικό πόσο γρήγορα προσαρμόζεται ο ΠΕ στα ενδιαφέροντα του χρήστη ακόμα και με περιορισμένη αρχική πληροφορία. Το αρνητικό είναι ότι χρειάζεται πολλές αξιολογήσεις πριν να ανακαλύψει την περιοχή των ενδιαφερόντων του χρήστη.

Δεν πρέπει να ξεχνάμε πως ακόμα και για τον χρήστη με τη μια σχετική σελίδα είναι λίγο τεχνητή η κατάσταση λόγω του ότι, εάν κάποιος ενδιαφέρεται για ένα θέμα φυσικά θα παρέχει μεταξύ των αξιολογήσεων όσο το δυνατόν περισσότερα παραδείγματα σχετικά με το θέμα. Στην προκειμένη όμως περίπτωση αυτό δεν επιτρέπεται γιατί δεν θέλουμε να εμπλουτίσουμε το προφίλ με περισσότερη σχετική ορολογία αλλά να διαπιστώσουμε τη δυνατότητα προσαρμογής που έχει ο ΠΕ στα ενδιαφέροντα του χρήστη. Έτσι λοιπόν και σε αυτή την περίπτωση ο ΠΕ επιτυγχάνει ένα ποσοστό που κυμαίνεται γύρω στο 50% από την 7 έως την 10 αξιολόγηση.

Από το γράφημα φαίνεται επίσης, το πόσο σημαντικός είναι ο αριθμός των αρχικών παραδειγμάτων που είναι σχετικά με το θέμα της έρευνας. Αυτό φαίνεται ιδιαίτερα στο δεύτερο χρήστη με τα 5 σχετικά παραδείγματα, όπου διατηρεί μέχρι την τελική αξιολόγηση τη διαφορά του ποσοστού επιτυχίας από τον πρώτο χρήστη.

Το ποσοστό επιτυχίας του τρίτου χρήστη με τα 10 σχετικά παραδείγματα, μέχρι την 5 αξιολόγηση κυμαίνεται περίπου στα επίπεδα του μέσου όρου της ομάδας του πρώτου πειράματος. Η μικρή διαφορά που παρατηρείται οφείλεται στο ένα τρίτο των παραδειγμάτων τα οποία έχουν διαφορετική θεματολογία.

7.4 Συμπεράσματα

Στο κεφάλαιο αυτό, αξιολογούμε την απόδοση του συστήματος *Webnaut* κατά τη διάρκεια δύο πειραμάτων τα οποία είχαν σαν στόχο την συλλογή πληροφοριών από το διαδίκτυο, σύμφωνα με προκαθορισμένα προφίλ χρηστών.

Ο σκοπός στο πρώτο πείραμα ήταν να διερευνήσουμε την ακρίβεια των υποδείξεων του συστήματος ενώ στο δεύτερο πείραμα, ο σκοπός ήταν να ελέγξουμε την ικανότητα του συστήματος να προσαρμόζεται στα ενδιαφέροντα του χρήστη ακόμη και στην περίπτωση που δεν διαθέτει αρκετές πληροφορίες.

Και στις δύο περιπτώσεις η απόδοση του συστήματος ήταν ικανοποιητική και επιπλέον βγάλαμε χρήσιμα συμπεράσματα για το κατά πόσο επηρεάζει η ανάδραση του χρήστη τις υποδείξεις του συστήματος.

Κεφάλαιο 8

ΣΥΜΠΕΡΑΣΜΑΤΑ - ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ

8.1 Συμπεράσματα

Στην παρούσα διατριβή παρουσιάσαμε την ανάλυση, την σχεδίαση καθώς και την υλοποίηση του συστήματος *Webnaut*, το οποίο αναπτύξαμε με σκοπό να υποστηρίξει έναν χρήστη σε διάφορα θέματα κατά την διάρκεια της πλοήγησης του στο διαδίκτυο. Μια από τις εργασίες του πράκτορα ήταν να αντιμετωπίσει τον όγκο των πληροφοριών στον παγκόσμιο ιστό και να συλλέξει μόνο τις πληροφορίες που ήταν σχετικές με τα ενδιαφέροντα του χρήστη. Τα οφέλη από την συγκεκριμένη υλοποίηση τα οποία αποτελούν και την συνεισφορά της παρούσας διατριβής, είναι :

- Η ανάπτυξη μεθοδολογίας για την αναπαράσταση και εκμάθηση των ενδιαφερόντων ενός χρήστη, με την βοήθεια ενός γενετικού αλγόριθμου δύο επίπεδων.

Το πεδίο της εκμάθησης των ενδιαφερόντων ενός χρήστη του διαδικτύου, το αντιμετωπίσαμε σαν ένα πολυδιάστατο πρόβλημα βελτιστοποίησης. Οι λέξεις που βρίσκονται στα κείμενα, τα οποία ενδιαφέρουν τον χρήστη, συνθέτουν το προφίλ του και αποτελούν τον χώρο έρευνας του προβλήματος μας. Ο συγκεκριμένος χώρος είναι αρκετά δυναμικός, αφού τα ενδιαφέροντα καθώς και η γνώση του χρήστη μεταβάλλονται με το χρόνο. Ο σκοπός μας είναι να αξιολογήσουμε την κάθε λέξη που συμμετέχει στο προφίλ, ώστε να δημιουργήσουμε ένα σύνολο από λέξεις οι οποίες περιγράφουν τα ενδιαφέροντα του χρήστη, τη δεδομένη χρονική στιγμή.

Η επιλογή μας να χρησιμοποιήσουμε λέξεις και όχι σύνθετες φράσεις, έχει σαν πλεονέκτημα ότι η δομή της αναπαράστασης του προφίλ είναι γενική - καθολική, αφού δεν απαιτούνται γλωσσικοί πόροι, οι οποίοι κάνουν αδύνατη την χρήση του πράκτορα σε διαφορετικό γλωσσικό περιβάλλον ή πεδίο εφαρμογής. Επίσης η συγκεκριμένη αναπαράσταση είναι συμβατή με την δομή των μηχανών αναζήτησης, οι οποίες χρησιμοποιούν μόνο λέξεις για την δεικτοδότηση των κειμένων.

Ο πράκτορας λογισμικού, χρησιμοποιώντας τεχνικές από τον χώρο των γενετικών αλγορίθμων, καταφέρνει να εξερευνήσει και να προσαρμοστεί στο πεδίο των ενδιαφερόντων του χρήστη. Η λειτουργία του αλγορίθμου αναπτύσσεται σε δύο επίπεδα επιτυγχάνοντας την γρήγορη αξιολόγηση του προφίλ αλλά και την ανεύρεση νέων κειμένων στο διαδίκτυο, τα οποία αποτελούν και τις υποδείξεις του συστήματος προς τον χρήστη. Ο πράκτορας συνδυάζοντας τις λέξεις με λογικούς τελεστές, καταφέρνει να δημιουργήσει σύνθετες δομές αναπαράστασης των ενδιαφερόντων του χρήστη. Μια τέτοια λειτουργία είναι χρήσιμη τόσο στην συλλογή των κειμένων όσο και στην εξερεύνηση του προφίλ για νέες θεματικές περιοχές ενδιαφερόντων, οι οποίες ίσως να μην έχουν υποπέσει στην αντίληψη του χρήστη.

- Η ανάπτυξη μεθοδολογίας για την σχεδίαση ευφυών συστημάτων τα οποία αποτελούνται από πράκτορες λογισμικού, οι οποίοι συνεργάζονται μεταξύ τους για την συλλογή πληροφοριών από το διαδίκτυο.

Η αξιολόγηση της μεθοδολογίας μπορεί να γίνει σε σχέση με τους αρχικούς στόχους ανάπτυξης του συστήματος. Πιο συγκεκριμένα, η απόφαση μας να στηρίξουμε όλο το σύστημα σε έναν τοπικό διακομιστή διαμεσολάβησης έχει τα εξής πλεονεκτήματα :

Όλη η λειτουργία του πράκτορα εκτυλίσσεται στον τοπικό υπολογιστή, οπότε δεν υπάρχουν θέματα ασφάλειας αφού δεν υπάρχει πιστοποίηση των στοιχείων του χρήστη, ούτε αποθήκευση των ευαίσθητων δεδομένων του, σε απομακρυσμένο εξυπηρετητή.

Υπάρχει απόλυτη συνεργασία του συστήματος με τα υπάρχοντα προγράμματα πλοήγησης στο διαδίκτυο αφού όλα συνεργάζονται με διακομιστή διαμεσολάβησης. Τα πλεονεκτήματα είναι : α) δεν αλλάζει σε τίποτα ο τρόπος πλοήγησης του χρήστη στο διαδίκτυο και β) υπάρχει η δυνατότητα υποστήριξης του χρήστη καθ' όλη την διάρκεια της πλοήγησης του.

Το μοντέλο ανάπτυξης του συστήματος μας δίνει την δυνατότητα προσθήκης νέων υπηρεσιών χωρίς να επηρεάζονται οι ήδη υπάρχουσες υπηρεσίες. Ο Πράκτορας Διακομιστής Διαμεσολάβησης (ΠΔΑ), αφού παραλάβει την αίτηση από τον χρήστη, θα μεταφέρει τον έλεγχο της διαδικασίας στην νέα υπηρεσία.

Όλοι οι πράκτορες λογισμικού εκτελούν ανεξάρτητα τις εργασίες τους και όλοι έχουν την δυνατότητα να επικοινωνούν με τον χρήστη, αφήνοντας τα μηνύματα τους στον Πράκτορα Μαυροπίνακα.

Ο κάθε πράκτορας λογισμικού έχει την δυνατότητα αποθήκευσης δεδομένων στον πράκτορα που είναι υπεύθυνος για την εκμάθηση των ενδιαφερόντων του χρήστη. Έτσι κατά την κατασκευή του προφίλ υπάρχουν στοιχεία από την αλληλεπίδραση του χρήστη με διάφορες υπηρεσίες.

- Προτείνουμε ένα μοντέλο αλληλεπίδρασης του χρήστη, κατά την διάρκεια της πλοήγησης του στο διαδίκτυο, με τους πράκτορες λογισμικού.

Στο προτεινόμενο μοντέλο αλληλεπίδρασης, ο χρήστης έχει σε όλα τα στάδια την πρωτοβουλία των κινήσεων, την ευχέρεια τροποποίησης των παραμέτρων του συστήματος, την δυνατότητα να θέσει εκτός λειτουργίας διάφορες υπηρεσίες, την άμεση ενεργοποίηση των υπηρεσιών μέσω της πινακίδας κεντρικού ελέγχου. Όλα τα παραπάνω χαρακτηριστικά αυξάνουν την εμπιστοσύνη του χρήστη προς το σύστημα.

Επίσης θα πρέπει να τονίσουμε ότι στο μοντέλο αλληλεπίδρασης που αναπτύξαμε, δεν χρειάζεται ο χρήστης να καθορίσει επακριβώς τα ενδιαφέροντα του ή να δημιουργήσει σύνολα από κείμενα, που ανήκουν σε συγκεκριμένες θεματικές κατηγορίες, με σκοπό να εκπαιδεύσει το σύστημα. Ο χρήστης την ώρα που επιθεωρεί μια σελίδα στο διαδίκτυο έχει την δυνατότητα να ενημερώσει το σύστημα ότι το συγκεκριμένο κείμενο ανήκει στα ενδιαφέροντα του.

Η αξιολόγηση των υποδείξεων του συστήματος γίνεται εξίσου εύκολα από τον χρήστη. Εκ πρώτης όψεως μπορεί να φαίνεται κουραστικό να δίνει ο χρήστης βαθμολογία στην κάθε υπόδειξη αλλά κατά την γνώμη μας δεν υπάρχει πιο άμεσος και σίγουρος τρόπος ενημέρωσης του συστήματος για το ποσοστό ικανοποίησης του χρήστη. Επιπλέον στην υλοποίηση μας, η ανάδραση του χρήστη έχει άμεσο και επωφελή αντίκτυπο στην ποιότητα των υποδείξεων του συστήματος, αφού γίνεται απευθείας προσαρμογή της αξίας συγκεκριμένων λέξεων που μετέχουν στο προφίλ του χρήστη και χρησιμοποιήθηκαν για την συγκεκριμένη υπόδειξη.

8.2 Μελλοντικές Επεκτάσεις

Σχετική δουλειά σε συστήματα συλλογής πληροφοριών από το διαδίκτυο ή από μεγάλες βάσεις δεδομένων έχει υποδείξει την αναγκαιότητα για καλύτερους αλγορίθμους εξαγωγής πληροφορίας, ειδικά όταν κάποιος προσπαθεί να συλλέξει πληροφορία από αδόμητο κείμενο με πολλές επαναλαμβανόμενες λέξεις, άσχετη πληροφορία, χωρίς καθιερωμένους όρους κ.λπ. Στα πλαίσια αυτά θα θέλαμε να αξιολογήσουμε την προσθήκη ενός εξειδικευμένου πράκτορα που σκοπό θα έχει να αναγνωρίζει τις συνώνυμες λέξεις και να αποδίδει την ίδια αξία σε αυτές.

Επίσης θέλουμε να αναπτύξουμε ένα πρωτόκολλο επικοινωνίας και ανταλλαγής πληροφοριών ανάμεσα σε συστήματα Webnaut. Πιο συγκεκριμένα, το κάθε σύστημα να τρέχει αυτόνομα στον υπολογιστή ενός χρήστη αλλά να υπάρχει η δυνατότητα επικοινωνίας με τους άλλους χρήστες με σκοπό την επιτάχυνση της διαδικασίας της εκμάθησης αφού θα μπορεί να γίνει ανταλλαγή πληροφοριών σε θέματα κοινού ενδιαφέροντος.

ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

ΑΝΚΕ	Αναλυτής Κειμένων
ΑΠ	Ανάκτηση Πληροφοριών
ΔΓΑ	Δευτερογενής Γενετικός Αλγόριθμος
ΔΚΑΠ	Διακτυακοί Κόμβοι Αναζήτησης Πληροφοριών
ΕΠ	Εικονική Πραγματικότητα
ΘΓΑ	Θεμελιώδης Γενετικός Αλγόριθμος
ΘΕΚ	Θεματικοί Κατάλογοι
ΙΣΑΡ	Ιστορικό Αρχείο
ΜΧΑ	Μηχανές Αναζήτησης
ΜΜΧΑ	ΜεταΜηχανές Αναζήτησης
ΠΓΑ	Πράκτορας Γενετικός Αλγόριθμος
ΠΔΔ	Πράκτορας Διακομιστής Διαμεσολάβησης
ΠΕ	Πράκτορας Εκμάθησης
ΠΕΔΚ	Πράκτορας Εξυπηρετητής Διαδικτυακού Κόμβου
ΠΚΕ	Πινακίδα Κεντρικού Ελέγχου
ΠΛ	Πράκτορες Λογισμικού
ΠΜΑΠ	Πράκτορας Μεταμηχανή Αναζήτησης Πληροφοριών
ΠΠΕΠ	Πράκτορας Πύλη Εικονικής Πραγματικότητας
ΠΦΔΤ	Πράκτορας Φιλοξενούμενων Δικτυακών Τόπων
ΦΠ	Φιλτράρισμα Πληροφοριών

ΠΑΡΑΡΤΗΜΑ Α

Το παράρτημα Α παρέχει αναφορά στις κλάσεις και τις διεπαφές της γλώσσας JAVA οι οποίες χρησιμοποιούνται από το σύστημα *WebNaut* και αναφέρονται στην διατριβή. Τα πακέτα που εμπεριέχουν τις συγκεκριμένες κλάσεις ή/και διεπαφές είναι αριθμημένα με ένα αύξων αριθμό και ταυτόχρονα παρέχεται και μια μικρή περιγραφή του πακέτου.
Η ηλεκτρονική διεύθυνση όλων των πακέτων της γλώσσας JAVA, στο δικτυακό τόπο της εταιρίας SUN είναι : http://java.sun.com/products/jdk/1.1/docs/api/packages.html

1. **java.net**
Το πακέτο java.net περιλαμβάνει κλάσεις και διεπαφές για την υλοποίηση εργασιών στο δίκτυο, όπως για παράδειγμα την σύνδεση ενός πελάτη με ένα εξυπηρετητή ή την δημιουργία ενός εξυπηρετητή.

ΚΛΑΣΕΙΣ

Socket	http://java.sun.com/products/jdk/1.1/docs/api/java.net.Socket.html
ServerSocket	http://java.sun.com/products/jdk/1.1/docs/api/java.net.ServerSocket.html
URL	http://java.sun.com/products/jdk/1.1/docs/api/java.net.URL.html

2. **java.lang**
Το πακέτο java.lang περιλαμβάνει τις βασικές κλάσεις και διεπαφές που αποτελούν τον κορμό της γλώσσας JAVA. Για παράδειγμα εμπεριέχει τις κλάσεις Integer, Double κ.λπ. αλλά και την κλάσση Object η οποία είναι μια γενική κλάσση και μάλιστα βρίσκεται στην κορυφή της ιεραρχίας της κληρονομικότητας.

ΔΙΕΠΑΦΕΣ

Runnable	http://java.sun.com/products/jdk/1.1/docs/api/java.lang.Runnable.html

ΚΛΑΣΕΙΣ

Thread	http://java.sun.com/products/jdk/1.1/docs/api/java. lang.Thread.html

3. **java.util**
Το πακέτο java.util περιλαμβάνει αρκετές βοηθητικές κλάσεις και διεπαφές όπως για παράδειγμα κλάσεις για την παραγωγή τυχαίων αριθμών, πίνακες αντικειμένων είτε σε μορφή Vector είτε Hashtable.

ΚΛΑΣΕΙΣ

Hashtable	http://java.sun.com/products/jdk/1.1/docs/api/java.util.Hashtable.html
Vector	http://java.sun.com/products/jdk/1.1/docs/api/java.util.Vector.html

ΠΑΡΑΡΤΗΜΑ Β

Το παράρτημα Β περιέχει όλες τις κλάσεις που έχουμε υλοποιήσει και αποτελούν την δομή του συστήματος WebNaut.

1. Κλάση **ProxyServer**

ProxyServer
+ Hashtable : ProxyProperties + String : ServerName + byte : [][]ToolBarText + int : ToolBarOption - ServerSocket : theServer - int : LocalPort
+Server() +Start() - boolean ReadProxyProperties() - boolean StartModules() - ShowMessage() +main()

2. Κλάση **ProxyClient**

ProxyClient
- Socket : ClientSocket
- HandleBrowser() - HandleClient() - changeToolbar() - markAsGood +run()

3. Κλάση **WWWServer**

WWWServer
+ Hashtable : WWWProperties + Hashtable : MimeTypes + String : ServerName + String : HTML_DEFAULT +File : PUBLIC_ERROR +File : PUBLIC_HTML
+Server() + boolean initialize() - boolean ReadWWWProperties()

4. Κλάση **WWWClient**

WWWClient
- HttpRequest : req - HttpResponse : res - Socket : ClientSocket
- boolean isBadMethod - sendDirectory(File ObjRequest) - sendFile(File ObjRequest) - sendHeader(File ObjRequest) +run()

5. Κλάση **HttpResponse**

HttpResponse
- URLConnection : conn - DataOutputStream : out
+ askFile(HttpRequest req) - close() + Socket connectToServer(String url) + Socket connectToServer(HttpRequest req) - HttpError(String error, boolean except) + Redirect(String location) + sendHtml(String page) + sendHtmlHeader() + sendHtmlParagraph(String paragraph) + write(byte[] line) + writeln(String line)

6. Κλάση **HttpRequest**

HttpRequest
- byte : [] CRLF - byte : [] buffer - Hashtable : Cookies - Hashtable : ServerTags - Hashtable : Reqtags - Vector : Header - String : HttpMethod - String : HttpURL - String : HttpVersion + int : bufferlen - InputStream : in
- analyzeHeader() - int byteIndexOf(byte[] line, int linelen, String pattern, int start, boolean checkJS) + boolean CGIProcess() - close() - copyBytes(DataOutputStream out) - String getLine() + String getParam(String key) + String getSrvParam(String key) + String getHttpVersion() + String getHttpURL() + String getHttpMethod() + String getCookie(String key) + Hashtable getTags() - int getContentLength() + boolean isTextHtml() - boolean isJavaScript(byte [] line, int start, int end) - boolean isValidHeader() - boolean ParseGet(String line) - ParseHeader(String line) - boolean ParsePost() - ParseCookie(String line) - int readNextLine() - boolean ScanUrl() + sendHeader(DataOutputStream out) + sendLine(DataOutputStream out, boolean flush) + serveClient(String Html, DataOutputStream out) + serveHeader(DataOutputStream out) + String urlDecode(String line)

7. Κλάση **SpiderServer**

SpiderServer
+ Hashtable : SpiderProperties + String : HeaderTxt + String : FooterTxt
+Server() + boolean initialize() - boolean ReadSpiderProperties()

8. Κλάση **SpiderClient**

SpiderClient
- HttpRequest : req - HttpResponse : res - Socket : ClientSocket
+ Hashtable getResults(Hashtable reqOptions, boolean Sck) - String FormatEngine(String EngineName, Vector Results, String Style) - FormatResults(Hashtable EnginesResults) - ServeClient() +run()

9. Κλάση **SpiderParser**

SpiderParser
- HttpRequest : req - HttpResponse : res - Socket : ClientSocket
+ run - parseRequest - Analyze() - Initialize() - int CheckText(String code, String type, int start) - int MatchNumber(String code, String type, int start) - int MatchString(String code, String type, int start) - rmvSpaces(int start)

10. Κλάση **SpiderEngines**

SpiderEngines
+ Hashtable : Parsers + Hashtable : Options + Vector : Names
+ boolean initialize()

11. Κλάση **SpiderLink**

SpiderLink
- String : Description - String : Hlink - String : No - String : Summary
+ String getDescription() + String getHLink () + String getNo () + String getSummary() + String formatAsHtml()

12. Κλάση **SiteMirror**

SiteMirror
+ Hashtable : ServerProperties + int : MODE
+ boolean initialize() - boolean ReadServerProperties() + handleRequest(Socket ClientSock) + handleRequest(Socket ClientSock, HttpRequest req, HttpResponse res)

13. Κλάση **SiteMirrorClient**

SiteMirrorClient
- HttpRequest : req - HttpResponse : res - Socket : ClientSocket - String : HtmlPage - String : LocalDir - int : depth - int : forward - String message - int : mode - int : stayin
+run()

14. Κλάση **SiteMirrorRobot**

SiteMirrorRobot
- String : BaseURL - String : RootDir - HyperLink : Root - ThreadGuard : Guard - int : NoOfGuards - Vector : UnProcLinks - Vector : VisitedLinks - boolean : searchall - boolean : inside - boolean : forward - int : depth - String : RootServer
- String GetContentType(HyperLink page) - ExtractLinks(HyperLink page) - ProcessANDSave(HyperLink page) - boolean ProcessThisLink(HyperLink temp) - boolean SameHostAsRoot(URL temp) - boolean isDirectory(String filename) +run()

15. Κλάση **HyperLink**

HyperLink
+ int : depth + URL : url

16. Κλάση **ThreadGuard**

ThreadGuard
- int : AllThreads - int : RunningThreads
+ IncreaseAvailable() + int IsAvailable()

17. Κλάση **LearningServer**

LearningServer
- int : DICTIONLIMIT + VDict : Dictionary + EvaluationGA : EvalVect + Hashtable : LearnServerProp + Vector : Messages + Hashtable : Qgrades - int : WaitDocs + boolean : waitEvaluation + GeneticAlgorithm : ga
+ ChangeState() - FillMessages(String filename) + boolean GaWaitsEvaluation() - InitDictionary - InitGA + Initialize() + String getEntry() - boolean ReadServerProperties() - boolean ReadQGrades(String filename) - addEntry(String location) + userFeedBack(HttpRequest xreq, HttpResponse xres) - formatPage(HttpRequest xreq, HttpResponse xres) + addVEEntry(String data) + String getVEEntries()

18. Κλάση **EvaluationGA**

EvaluationGA
+ Vector : Words + Vector : Score + Vector : Operands + Vector : Links
- openfile(String filename) + int size() + saveFile(String filename) + removeElementAt(int i) + String getWordsAt(int i) + String getScoreAt(int i) + String getOperandsAt(int i) + String getLinkAt(int i) + String getElementAt(int i) + addEntry(String line) + addEntry(String link, String word, String operand, String score)

19. Κλάση **BlackBoard**

BlackBoard
- Vector : BBMessages - Hashtable : BBProp - String : Header - String : Footer - String : sTag - String : eTag
- boolean ReadProperties() + boolean initialize() + addEntry(String Message) + String getMessages()

20. Κλάση **BadHistoryServer**

BadHistoryServer
- Hashtable : HistoryItems - Hashtable : HistoryProp - int : HistoryLimit
- boolean ReadProperties() + boolean initialize() + addEntry(String location) + boolean contains(String location) - TruncHistoryFile()

21. Κλάση **GeneticAlgorithm**

GeneticAlgorithm
- BadHistoryServer : BadUrls - Vector : FinalWords - Vector : FinalOpers - Vector : FinalUrls - Vector : FinalVals - int : Generation - int : MaxGeneration - Random : Genrand + boolean : waitEvaluation - float : ProbCross - float : ProbMut - Vector : PGAPopulation
- boolean ReadGaProperties() - InitPGAPopulation() - CreateNewPopulation() - PGACrossOver() - PGAMutation() - PopulationFittness - boolean ReadyToRun() - SortPopulation() - boolean isDictionaryEmpty - String randomWord() - ProcessMessages () - ProcessEvaluations() + startGA() + run()

22. Κλάση **URLDecoder**

URLDecoder
+ String decode(String s)

23. Κλάση **TagBuilder**

TagBuilder
- Vector : Tags - Hashtable : Htags - String : Expr - int : mode
+ Hashtable getHashTags() + Vector getVectTags() - AddTag(String line) - AnalyzeTags() - setExpression(String Expression)

24. Κλάση **TagText**

TagText
- String : [3] Pedia
+ String getTag() + String getCode() + String getType()

25. Κλάση **URLHandle**

URLHandle
- String : filename; - String : RootDir; - ThreadGuard : t;
URLHandle(String filename, String RootDir, ThreadGuard Guard) + run - SaveFile - CreateSubDirs(String filepath)

26. Κλάση **VDict**

Vdict
- Vector : Words - Vector : Freqs - Vector : Docs - float : [] Weights
VDict(String fileString) + boolean Initialize() - ReadDictionary() + SaveDictionary() + int size() - SortAllVectors(int option) + int containsWord(String word) + addEntries(Hashtable Swords, Hashtable SFreqs) + addElement(String line) + String getElementAt(int i) + CalculateWeights() + float Similarity(Hashtable SWords, Hashtable SFreqs) + ModifyDictionary(float score, Hashtable SWords, Hashtable SFreqs) - DecreaseDictionary(int option, float limit)

27. Κλάση **TextServer**

TextServer
+ Hashtable : TextProp + Hashtable : StopWords + Hashtable : ExclLexico + int WORDLEN + MAXLENDOC + boolean RESTRICTNUMBERS
+ boolean Initialize() - FillVector(Hashtable Results, String filename)

28. Κλάση **HtmlParser**

HtmlParser
- Vector : Links - Hashtable : Sfreqs - Hashtable : Swords - String : DocText
HtmlParser(String Filename, boolean isUrl) - ReadFromStream(DataInputStream in) + Hashtable getSFreqs() + Hashtable getSWords() + Vector getLinks() - boolean isNumber(String word) - WordFrequency(String token) - ParseText(String token) + Analyze()

29. Κλάση **Porter**

Η υλοποίηση της συγκεκριμένης κλάσσης - με μερικές μόνο τροποποιήσεις - είναι αντίγραφο της κλάσσης *Porter* που είναι διαθέσιμη στο διαδίκτυο :

Συγγραφέας : Φώτης Λαζαρίνις
Ηλ. Διεύθυνση : *http://csgrad.cs.vt.edu/~jxzhao/6704/project/code/porter.txt* ή *http://www.cs.brandeis.edu/~jamesp/cs112/porter.java*

Porter
+ String : [] prefixes + String : [] suffixes2 + String : [] suffixes3 + String : [] suffixes4
- String Clean (String str) - boolean hasSuffix (String word, String suffix, String stem) - boolean vowel(char ch, char prev) - int measure(String stem) - boolean containsVowel(String word) - boolean cvc(String str) - String step1(String str) - String step2(String str) - String step3(String str) - String step4(String str) - String step5(String str) - String stripPrefixes(String str) - String stripSuffixes(String str) + String stripAffixes(String str)

30. Κλάση **HistoryServer**

HistoryServer
- Hashtable : HistoryItems - Hashtable : HistoryProp - int : HistoryLimit
- boolean ReadProperties() + boolean initialize() + addEntry(String location) + boolean contains(String location) - TruncHistoryFile()

31. Κλάση **ResultsGA**

ResultsGA
- String : filename - Vector : Links - Vector : Words - Vector : Operands
ResultsGA(String xfilename) + int size() - openFile(String filename) + saveFile() + addEntries(Vector GALinks, Vector GAWords, Vector GAOpers) + removeElementAt(int i) + String formatHtml(String formAction, String marks, String retURL)

32. Κλάση **SGA**

SGA
- Vector : Population - Vector : PopulVals - Hashtable : MemOper - Hashtable : SGAMemory - Vector : FinalLinks - Vector : FinalVals - Vector : FinalOper - float : fittness
SGA(VDict Dict, BadHistoryServer BadHttps, Vector GenWords) + runSGA() - CreateNewSGAPopulation() - SGAInversion() - SGAMutation() - SGACrossOver() - InitSGAPopulation() - SGAPopulFittnes() - Hashtable CreateQuery(Vector Xoperands) + float getFittness()

33. Κλάση **VRGateway**

VRGateway
- Socket : srv, xclient - DataOutputStream : out - InputStream : in - boolean : isRunning, isConnected - Thread : thrMudClient - byte : [] buffer - int : port - URL : base - HttpResponse : res - HttpRequest : req - TextField : CommText - Label : label1, label2; - TextArea : MessText - Button StartButton, StopButton, QuitButton, LearnButton - Hashtable : ReserveWords, UserCommands
+ VRGateWay(Socket client, HttpRequest xreq, HttpResponse xres) + destroy() + init() - boolean InitServerConnection() - CloseConnection() - ParseServerCommand(String comm) - AnalysisOfInput(String Serverdata) - getReservedWords(String data) - ParseUserCommand(String data) - ReceiveData() - SendData(String data) + StartButton_Action(ActionEvent event) + StopButton_Action(ActionEvent event) + QuitButton_Action(ActionEvent event) + LearnButton_Action(ActionEvent event) + CommText_EnterHit(ActionEvent event) + run()

34. Κλάση **SymAction**

SymAction
+ actionPerformed(ActionEvent event)

35. Κλάση **SymWindow**

SymWindow
+ windowClosing(WindowEvent event)

36. Κλάση **MessageDialog**

MessageDialog
Button : yesButton Label : label1
+MessageDialog(Frame parent, boolean modal, String title, String message) + addNotify() + show() + yesButton_Clicked(ActionEvent event)

37. Κλάση **MudServer**

MudServer
+ Hashtable : ServerProp - ServerSocket : theServer - int : LocalPort
+ MudServer() + startServer() - boolean ReadServerProperties() + main(String[] args)

38. Κλάση **MudClient**

MudClient
- byte : [] buffer - LogicServer : ls - Socket : client - DataOutputStream : out - InputStream : in
+ SrvMudClient(Socket sck, String filename, int actTime) - InitConnection() - CloseSocketConnection() - InitEngine(String filename) - StopEngine() - boolean SendData(String Message) - String ReceiveData() + run()

Παράρτημα Γ

Στο παράρτημα Γ αναφέρουμε τις ηλεκτρονικές διευθύνσεις που χρησιμοποιούμε στα πλαίσια της διατριβής. Σε μερικές περιπτώσεις αναφέρονται περισσότερες από μία διευθύνσεις για ένα συγκεκριμένο θέμα ή η ίδια διεύθυνση αναφέρεται σε περισότερα του ενός θέματα.

[1] **HTTP**
version 1.0 http://www.ietf.org/rfc/rfc2616.txt
version 1.1 http://www.ietf.org/rfc/rfc1945.txt

[2] **GOPHER**
http://www.ietf.org/rfc/rfc1436.txt

[3] **FTP**
http://www.ietf.org/rfc/rfc959.txt

[4] **NETNEWS**
http://www.ietf.org/rfc/rfc977.txt
http://www.ietf.org/rfc/rfc850.txt

[5] **WAIS**
http://www.ietf.org/rfc/rfc1625.txt

[6] **MIME**
http://www.ietf.org/rfc/rfc822.txt

[7] **CGI**
http://www.w3.org/CGI/

[8] **URL**
http://www.ietf.org/rfc/rfc1738.txt

[9] **JAVA**
http://java.sun.com

a. **JDBC**
http://splash.javasoft.com/jdbc/

[10] **JAVASCRIPT**
http://home.netscape.com/eng/mozilla/3.0/handbook/javascript/index.html
http://msdn.microsoft.com/library/en-us/script56/html/js56jsoriJScript.asp

[11] **CONTENT TYPES**
http://www.ietf.org/rfc/rfc2045.txt
http://www.ietf.org/rfc/rfc2046.txt

[12] **HTML**
http://www.ietf.org/rfc/rfc1866.txt
http://www.w3.org/TR/html4

a. **FRAMES**
http://www.w3.org/TR/html4/present/frames.html

b. **FORMS**
http://www.w3.org/TR/html4/interact/forms.html

c. **TABLES**

http://www.ietf.org/rfc/rfc1942.txt

d. **META-TAGS**
http://www.w3.org/TR/html4/struct/global.html#edef-META

e : **IMAGE MAP**
http://www.w3.org/TR/html4/struct/objects.html#h-13.6

[13] **COOKIES**
http://www.ietf.org/rfc/rfc2109.txt

[14] **AltaVista**
http://www.altavista.com

[15] **Excite**
http://www.excite.com

[16] **Hotbot**
http://www.hotbot.com

[17] **Lycos**
http://www.lycos.com

[18] **Yahoo**
http://www.yahoo.com

[19] **DARPA**
http://www.arpa.mil/

[20] **TCP**
http://www.ietf.org/rfc/rfc793.txt

[21] **UDP**
http://www.ietf.org/rfc/rfc768.txt

[22] **IDC**
http://www.idc.com/utilites/search.htm

[23] **DNS**
http://www.ietf.org/rfc/rfc1034.txt

[24] **Netscape**
http://www.netscape.com

[25] **VBScript**
http://msdn.microsoft.com/library/en-us/script56/html/vtoriVBScript.asp

[26] **Microsoft Internet Explorer**
http://www.microsoft.com/ie

[27] **Java Server Pages**
http://java.sun.com/products/servlet/archive.html

[28] **NUA**
http://www.nua.ie/surveys/how_many_online/

[29] **ISO 8879:1986**
Information processing - Text and office systems - Standard Generalized Markup Language
http://www.iso.ch/cate/d16387.html

[30] **ActiveX**
http://www.microsoft.com/com/

[31] **Open Directory Project**
http://dmoz.org/

[32] **LookSmart**
http://www.looksmart.com/

[33] **MetaCrawler**
http://www.metacrawler.com

[34] **DogPile**
http://www.dogpile.com

[35] **Archie**
http://archie.emnet.co.uk/

[36] **Veronica**
gopher://gopher.floodgap.com:70/1/v2/help
gopher://veronica.scs.unr.edu/11/veronica

[37] **Κινηματογράφος**
http://movies.go.com/

[38] **Μουσική**
http://www.music.com/

[39] **Τέχνες**
http://www.wwar.com/
http://www.culturefinder.com/
http://www.sfmoma.org/artfinder/artfinder.html

[40] **Εύρεση Εργασίας**
http://www.jobseeker.com

[41] **Επιστήμη - Ερευνα - Εκπαίδευση**
http://www.euratin.net/
http://tantalum-e2.hgmp.mrc.ac.uk/hypermail/BIOTECHNIQUES/
http://www.cordis.lu/en/home.html

[42] **Internationalization of the Hypertext Markup Language**
http://www.ietf.org/rfc/rfc2070.txt

[43] **Internet Growth**
a : ***Internet Software Consortium***
http://www.isc.org/ds/WWW-200107/index.html

b : ***NetSizer.com***
http://www.netsizer.com/index.html

c : ***Matrix.Net***
http://www.matrix.net/index.html

d : ***NetCraft***
http://www.netcraft.com/

[44] **The Search Engine Standards Project**
http://www.searchenginewatch.com/standards/990204.html

[45] **A Standard for Robot Exclusion**
http://www.robotstxt.org/wc/norobots.html

[46] **Java Developer**
http://developer.java.sun.com

[47] **JavaWorld**
http://www.javaworld.com

[48] **Developer.com**
http://www.developer.com

[49] **Rolling Stone Magazine**
http://www.rollingstone.com/

[50] **MTV**
http://www.mtv.com

[51] **All Music Guide**
http://www.allmusic.com/

[52] **USATODAY Sports**
http://www.usatoday.com/sports/sfront.htm

[53] **NBA**
http://www.nba.com

[54] **Sports.com**
http://www.sports.com/en/

[55] **Sun Microsystems**
http://www.sun.com

a : **Jini**
http://java.sun.com/products/jini/whitepapers/architectureoverview.pdf

[56] **IBM**
http://www.ibm.com

a: **TSpaces**
www.almaden.ibm.com/cs/Tspaces

[57] **Microsoft**
http://www.microsoft.com

a : **Project Millennium**
http://research.microsoft.com/sn/Millennium

[58] **SICStus Prolog**
http://www.sics.se/isl/sicstuswww/site/documentation.html

[59] **Amzi Prolog**
http://www.amzi.com/

a: **Natural Language Processing**
http://www.amzi.com/AdventureInProlog/a15nlang.htm

b: **Logic Server**
http://www.amzi.com/articles/java_class.htm

Βιβλιογραφία

[Ack97] Ackerman M., Billsus D., Gaffney S., Hettich S., Khoo G., Kim D.J., Klefstad R., Lowe C., Ludeman A., Muramatsu J., Omori K., Pazzani M.J., Semler D., Starr B., Yap P., "Learning Probabilistic User Profiles: Applications for Finding Interesting Web Sites, Notifying Users of Relevant Changes to Web Pages, and Locating Grant Opportunities", AI Magazine, AAAI Press, vol 18, no 2, 1997, 47-56

[Arm95] Armstrong R., Freitag D., Joachims T., Mitchell T., "WebWatcher: A Learning Apprentice for the WWW", In Proceedings of AAAI Symp. Information Gathering from Heterogeneous, Distributed Environments, AAAI Tech. Report SS-95-08, Menlo Park, 1995, 6-12

[Atk99] Atkins E.M., Abdelzaher T.F., Shin K.G., Durfee E.H., "Planning and resource allocation for hard real-time, fault-tolerant plan execution", In Proceedings of the Third International Conference on Autonomous Agents (Agents'99), ACM Press, Seattle, WA, USA, 1999, 244-251

[Ayl00] Aylett R., Luck M., "Applying artificial intelligence to virtual reality: Intelligent virtual environments", Applied Artificial Intelligence Journal, vol 14, no 1, Taylor & Francis Group, UK, 2000, 3-32

[Bal97] Ball G., Ling D., Kurlander D., Miller J., Pugh D., Skelly T., Stankosky A., Thiel D., Dantzich M.V., Wax T., "Lifelike Computer Characters: the Persona project at Microsoft Research", in Software Agents, Jeffrey Bradshaw (ed), AAAI Press, 1997, 191-222

[Bald98] Baldi M., Picco G.P., "Evaluating the Tradeoffs of Mobile Code Design Paradigms in Network Management Applications", Proceedings of the 20th International Conference on Software Engineering, IEEE Computer Society Press, 1998, 146-155

[Balv95] Balabanovic M., Shoham Y., "Learning Information Retrieval Agents: Experiments with Automated Web Browsing", In Proceedings of AAAI Symp. Information Gathering from Heterogeneous, Distributed Environments, AAAI Tech. Report SS-95-08, Menlo Park, 1995, 13-18

[Balv97] Balabanovic M., "An Adaptive Web Page Recommendation Service", Proceedings of the First International Conference on Autonomous Agents (Agents'97), ACM Press, 1997, 378-385

[Balv98] Balabanovic M.,"Learning to Surf: Multiagent Systems for Adaptive Web Page Recommendation", PhD thesis, Department of Robotics and Computer Science, University of Stanford, Stanford, 1998

[Bat98] Bates J., Blumberg B., Goldberg A., Hayes-Roth B., Stern A., "Autonomous

agents in entertainment", In Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98), Sycara K.P., Wooldridge M. (eds), ACM Press, N.Y., 1998, 140-140

[Bel00] Bellman K.L., Landauer C., "Playing in the MUD: Virtual Worlds are Real Places", Applied Artificial Intelligence, Taylor & Francis, vol 14, no 1, 2000, 93-123

[Belk92] Belkin N.J., Croft W.B., "Information filtering and information retrieval: Two sides of the same coin.", Communications of the ACM, ACM Press, vol. 35, no. 12, 1992, 29–38

[Bha98] K. Bharat, A. Broder, "A technique for measuring the relative size and overlap of public Web search engines", Computer Networks, Elsevier Science, Volume 30, Issue 1-7, 1998, 379-388

[Bra98] Brandt C., Kyriakaki G., Lamotte W., Loling R., Maragoudakis Y., Mavraganis Y., Klaus M., Pappas N., "The SICMA Multimedia Server and Virtual Museum Application", Springer Verlag, Lecture Notes Computer Science 1425, 1998, 83-96

[Brat90] Bratko I., "PROLOG Programming for Artificial Intelligence", Addison Wesley, NY, 1990

[Bri98] S. Brin, L. Page, "The anatomy of a large-scale hypertextual Web search engine", Computer Networks, Elsevier Science, Volume 30, Issue 1-7, 1998, 107-117

[Brs99] Brusic I., Hassler V., Lugmayr W., "Mobile agents in the mobile telephone network management", First International Workshop on Mobile Agents for Telecommunication Applications (MATA'99), World Scientific Publishing Ltd., Ottawa, Canada, 1999, 229-242

[Bru91] Brustoloni J.C., "Autonomous Agents: Characterization and Requirements", Carnegie Mellon Technical Report CMU-CS-91-204, Pittsburgh: Carnegie Mellon University, 1991

[Bow01] Bowman D., Kruijff E., LaViola J., Poupyrev I. "An Introduction to 3-D User Interface Design", PRESENCE: Teleoperators and Virtual Environments, MIT Press, vol 10, no 1, 2001, 96-108

[Bye98] D. Byersa, "Full-text indexing of non-textual resources", Computer Networks, Elsevier Science, Volume 30, Issue 1-7, 1998, 141-148

[Bie99] Bieszczad A., Biswas P.K., Buga W., Malek M., H.Tan, "Management of Heterogeneous Networks with Intelligent Agents", Bell Labs technical Journal, Lucent Technologies Inc., vol 4 no 4, 1999, 109-135

[Bra99] Brazier F.M.T., Jonker C. M., Jungen F. J., Treur J., "Distributed scheduling to support a call centre: a co-operative multi-agent approach", Applied Artificial Intelligence Journal, vol. 13, Taylor & Francis Group, UK, 1999, 65-90

[Bra98] Brazier F.M.T., Cornelissen F., Gustavsson R., Jonker C.M., Lindeberg O., Polak B., Treur J., "Agents Negotiating for Load Balancing of Electricity Use", In Proceedings of the 18th International Conference on Distributed Computing Systems, ICDCS'98, IEEE Computer Society Press, 1998, 622-629

[Bro91] Brooks R.A., "Intelligence Without Reason", Proceedings of 12th Int. Joint Conf. on Artificial Intelligence, Sydney, Australia, Morgan Kaufmann, 1991, 569-595.

[Cab00] Cabri G., Leonardi L., Zambonelli F., "Agents for information retrieval: Issues of mobility and coordination", Journal of Systems Architecture, Elsevier Science B.V., vol 46 (2000) 1419-1433

[Cag97] Caglayan A., Harrison C. "Agent Sourcebook - A Computer Guide to Desktop Internet, and Intranet Agents", Wiley Computer Publishing, New York, NY, 1997

[Car93] Carlsson C., Hagasand O., "DIVE: a platform for multi-user virtual environments", Computers & Graphics, Elsevier Science, vol 17, no 6, 1993, 663-669

[Carr98] Carraro G.U., Edmark J.T., Ensor J.R., "Techniques for handling video in virtual environments", In Proceedings of the 25th annual conference on Computer Graphics, Orlando, USA, 1998, 353-360

[Cas96] Casselberry R., Baker D., Benett G., Calabria J., "Running a Perfect Intranet", Que Corporation, In, 1996

[Cast95] Castelfranchi C., "Guarantees for autonomy in cognitive agent architecture", In Intelligent Agents: Theories, Architectures, and Languages, Wooldridge, M. and Jennings, N. R., (eds), Lecture Notes in Artificial Intelligence, vol 890, Springer-Verlag: Heidelberg, 56–70

[Cha98] Chang S.F., "Content-based indexing and retrieval of visual information", IEEE Signal Processing Magazine, Vol 14, no 4, IEEE Computer Society Publications Office, 1997, 45-48.

[Che96] Cheong F., "Internet Agents: Spiders, Wanderers, Brokers, and Bots", New Riders, Indianapolis, 1996.

[Ches97] Chess D., Grosof B., Harrison C., Levine D., Parris C., Tsudik G., "Itinerant agents for mobile computing", In Huhns M.N. and Singh M.P. (eds), Readings in Agents, Morgan Kaufmann, San Francisco, California, 1997, 267-282.

[Chi91] Chin D. N., "Intelligent Interfaces As Agents", in Intelligent User Interfaces, J.W. Sullivan and S.W. Tyler (Eds.), Addison-Wesley, Reading, MA, 1991, 127-206

[Cho98] Cho J., Garcia-Molina H., Page L., "Efficient Crawling Through URL Ordering", Proceedings of the 7th International World Wide Web Conference, Brisbane, Australia, 1998, Elsevier Science, 161-172

[Chu99] Churchill E.F., Bly S.,"Virtual Environments at Work: ongoing use of MUDs in the Workplace", In Proceedings of the International Joint Conference on Work Activities Coordination and Collaboration, ACM Press, 1999, 99-108

[Coh96] Cohen W.W., "Learning rules that classify e-mail", In Proceedings of the AAAI Spring Symposium on Machine Learning in Information Access, AAAI Tech. Report SS-96-05, Stanford, 1996, 18-25

[Coo97] Coomans M.K.D., Timmermans H.J.P., "Towards a Taxonomy of Virtual Reality User Interfaces", In Proceedings of the IEEE Conference on Information Visualization (IV '97), IEEE Press, 1997, 279-285

[Cor01] Corradi A., Montanari R., Stefanelli C, "Security of mobile agents on the internet", Internet Research: Elecronice Network Applications and Policy, MCB University Press, vol 11, no 11, 2001, 84-95

[Cou99] Courtois M.P., Berry M.W., "Results Ranking in Web Search Engines", Online, Online Inc., vol 23, no 3, 1999

[Cov94] Covington M.A., "Natural Language Processing for Prolog Programmers", Prentice Hall Inc, New Jersey, 1994

[Cov97] Covington M.A.,"Prolog Programming in Depth", Prentice Hall Inc, New Jersey, 1997

[Cra00] Craven M., DiPasquo D., Freitag D., McCallum A., Mitchell T., Nigam K., Slattery S., "Learning to construct knowledge bases from the World Wide Web", Artificial Intelligence 118, Elsevier Science, 2000, 69–113

[Cre99] Cremonini M., Corradi A., Montanari R., Stefanelli C., "Mobile agents and security: Protocols for integrity", In the Proceedings of the Second IFIP WG 6.1 International Working Conference on Distributed Applications and Interoperable Systems (DAIS'99), Helsinki, 1999, 177-190

[Cuen99] Cuena J., Ossowski S., "Distributed models for decision support", Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence, Gerhard Weiss (ed), The MIT Press, Cambridge, Massachusetts, 1999, 459-504

[Cze99] Czernuszenko M., Sandin D., Johnson A., DeFanti T., "Modeling 3D Scenes

from Video", In The Visual Computer Journal, Special Issue on Real-time Virtual Worlds, Springer-Verlag, vol 15, no 7/8, 1999, 341-348

[Dal97] Dale J., "A Mobile Agent Architecture for Distributed Information Management", PhD thesis, Department of Electronics and Computer Science, University of Southampton, Southampton, 1997

[Dar98] Darnell R., "HTML Unleashed, Professional Reference Edition", Sams Publishing, 1998

[Dav91] Davis L. "Handbook of Genetic Algorithms", Van Nostrand Reinhold, NY, 1991

[Dec97] Decker K., Pannu A., Sycara K.P., Williamson M.,"Designing Behaviors for Information Agents", In Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, CA, USA, ACM Press, 1997, 404-412

[Don97] Donath J.S., "Inhabiting the virtual city: The design of social environments for electronic communities", Phd Thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology, 1997

[Doo97] Doorendos R.B., Etzioni O., Weld D.S., "A Scalable Comparison-Shopping Agent for the WWW", In Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, CA, USA, ACM Press, 1997, 39-48

[Dra99] Drashansky T., Houstis N. E., Ramakrishnan N., Rice R.J., "Networked Agents for Scientific Computing", Communications of the ACM, vol 42, no 3, ACM Press, 1999, 48-54

[Edw96] Edwards P.,Bayer D.,Green C. L.,Payne T. R., "Experience with learning agents which manage internet-based information", In Proceedings of the AAAI Stanford Spring Symposium on Machine Learning in Information Access, AAAI Press, 1996, 31-40

[Etz94] Etzioni O., Weld D., "A Softbot-Based Interface to the Internet", Communications of the ACM, ACM Press,vol 37, no 7, 1994, 72-79

[Fil98] R.E. Filman, S. Pant, "Searching the Internet", IEEE Internet Computing, IEEE Computer Society Publications Office, Vol 2, no 4, 1998, pp. 21-23

[Fis94] Fisher M., Wooldridge M., "Specifying and executing protocols for cooperative action", In Proceedings of International Working Conference on Cooperating Knowledge-Based Systems (CKBS), DAKE Centre,U.K., 1994, 295-306

[Fog97] Fogel, L.J. "A Retrospective View and Outlook on Evolutionary Algorithms", *Computational Intelligence: Theory and Applications, 5th Fuzzy Days,*

Springer-Verlag, Berlin, 1997, 337-342

[Fra96] Franklin S., Graesser Art, "Is it an Agent, or Just a Program?: A Taxonomy for Autonomous Agents", Lecture Notes in Computer Science, Vol. 1193, Springer, 1997, 21-35

[Gal91] Gal A., Lapalme G., Saint-Dizier P., Somers H., "Prolog for Natural Language Processing", Wiley, 1991

[Gar95] Garzotto F., Mainetti L., Paolini P., "Hypermedia Design Analysis, and Evaluation Issues", Communications of ACM, ACM PressVol 38, No 8, August 1995, 74-86

[Gao98] Gao X., Sterling L., "A methodology for building information agents", Web Technologies and Applications, Yang Y.,Li M., Ellis A., (eds), International Academic Publishers, China, 1998, 43-52

[Gaz89] Gazder G., Mellish C.S., "Natural Language Processing in Prolog", Addison-Wesley, 1989

[Gen94] Genesereth M.R., Ketchpel S.P.,"Software Agents", Communications of the ACM, ACM Press, vol 37, no 7, 1994, 48-53

[God99] Godireaux C., El-Guedj P.O., Revolta F., Nugues P.,"Ulysse: An Interactive Spoken Dialogue Interface to Navigate in Virtual Worlds", Virtual Worlds on the Internet, Vince J. and Earnshaw R. (eds), IEEE Computer Society Press, Chapter 4, 1999, 53-70

[Gol89] Goldberg E.D., "*Genetic algorithms in search, optimization and machine learning*", Addison-Wessley Publishing Company Inc., 1989

[Gol94] Goldberg E.D., "*Genetic and Evolutionary Algorithms Come of Age*", Communications of ACM, Vol. 37, No 3, 1994, 113-119

[Goo93] Goodwin R., "Formalizing properties of agents", Technical Report CMU–CS–93–159, School of Computer Science, Carnegie-Mellon University, Pittsburgh, PA, 1993.

[Gra00] Gray R., Kotz D., Cybenko G, Rus D., "Mobile Agents: Motivations and State-of-the-Art Systems", Technical Report TR2000-365, Dartmouth College, Computer Science, Hanover, 2000

[Gro95] Grosof B.N., Levine D.W., Chan H.Y., Parris C.J., Auerbach J.S., "Reusable Architecture for Embedding Rule-based Intelligence in Information Agents", IBM Research Report RC 20305, IBM, 1995

[Gud97] V.K. Gudivada, V.V. Raghavan, W.I. Grosky, R. Kasanagottu, "Information Retrieval on the World Wide Web", IEEE Internet Computing, IEEE Computer Society Publications Office, Vol 1, no 5, 1997, pp. 58-68

[Hay95] Hayes-Roth B., "An Architecture for Adaptive Intelligent Systems", Artificial Intelligence, Elsevier Science, vol. 72, no. 1-2, 1995 329-365

[Hay96] Hayes-Roth B., Larsson J.E., "A domain-specific software architecture for a class of intelligent patient monitoring agents", Journal of Experimental and Theoretical Artificial Intelligence, vol 8, no 2, Taylor & Francis Group, UK, 1996, 149-171

[Hel96] Helfman J.I., Isbell C. L., "Ishmail: Immediate identification of important information. Submitted to CHI-96, 1995.

[Hen99] M.R. Henzinger, A. Heydon, M. Mitzenmacher, M. Najork, "Measuring Index Quality Using Random Walks on the Web", Proceedings of the 8th International World Wide Web Conference, Toronto, Canada, 1999, Elsevier Science, 213-225

[Hoc98] R. Hock, "How to Do Field Searching in Web Search Engines: A Field Trip", Online, Online Inc.,Vol 22, no 3, 1998

[Hol75] Holland J.H., "*Adaption in natural and artificial systems*", Ann Arbor: The University of Michigan Press, 1975

[Huh93] Huhns N.M., "Networking Embedded Agents", IEEE Internet Computing, vol. 3, no 1, IEEE Computer Society Press, 1999, 91-93

[Joa97] Joachims T., Freitag D., Mitchell T.M., "Web Watcher: A Tour Guide for the World Wide Web", Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, IJCAI 97, Morgan Kaufmann, 1997, 770-777

[Jen98] Jennings N.R., Wooldridge M.J., "Applications of intelligent agents", Agent Technology: Foundations, Applications, and Markets, In Nicholas R. Jennings and Michael J. Wooldridge, (eds), Springer-Verlag: Heidelberg, Germany, 1998, 3-28

[Joh98] Johnson W.L.,Rickel J.,Stiles R.,Munro A.,"Integrating Pedagogical Agents into Virtual Environments", PRESENCE: Teleoperators and Virtual Environments, MIT Press, vol 7, no 6, 1998, 523-546

[Joh00] Johnson W.L., Rickel J.W., Lester J.C., "Animated Pedagogical Agents: Face-to-Face Interaction in Interactive Learning Environments", International Journal of Artificial Intelligence in Education, vol 11, IAIED Society, 2000, 47-78

[Jon95] Jonsson A., "A Dialogue Manager for Natural Language Interfaces" In

Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95, Morgan Kaufmann, Canada, 1995, 1405-1413

[Jong75] De Jong, K.A. "*An analysis of the behavior of a class of genetic adaptive system*", Doctoral Dissertation, University of Michigan, 1975

[Jos99] Joshi A., Singh P.M., "Multiagent Systems on the Net", Communications of the ACM, ACM Press, vol 42, no 3, 1999, 39-40

[Kah95] Kahn P., "Visual Cues for Local and Global Coherence in the WWW", Communications of ACM, ACM PressVol 38, No 8, August 1995, 67-71

[Kar97] Karjoth G., Lange D. B., Oshima M., "A Security Model for Aglets", IEEE Internet Computing, IEEE Computer Society Press, vol 1, no 4, 1997, 68-77

[Kle99] J.M. Kleinberg, R. Kumar, P. Raghavan, S. Rajagopalan, A.S. Tomkins, "The Web as a graph: measurements, models, and methods", Proc. 5th Annual Int. Conf. Computing and Combinatorics, Lecture Notes in Computer Science, Vol 1627, Springer Verlag, 1999, 1-17

[Klu01] Klusch M., "Information Agent Technology for the Internet: A Survey", Journal on Data and Knowledge Engineering, Special Issue on Intelligent Information Integration, Elsevier Science, vol 36, no 3, 2001, 337-372

[Koz93] Kozierok R., Maes P., "A learning interface agent for scheduling meetings", In Proceedings of the International Workshop on Intelligent User Interfaces, Gray W.D., Hefley W.E., Murray D. (eds), ACM Press, NY, USA, 1993, 81-88

[Kru97] Krulwich B., "LifeStyle Finder: Intelligent User Profiling", AI Magazine, AAAI Press vol. 18, no. 2, 1997, 37-45

[Lai88] Lai K.Y., Malone T. W., " Object lens: a "spreadsheet" for cooperative work", ACM Transactions on Office Information Systems, ACM Digital Library, vol. 6, no. 4, 1988, 115-124

[Lam96] LaMacchia B. A., "Internet Fish", Ph.D. Thesis, Department. of Electric Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 1996

[Lan99] Lange D. B., Oshima M., "Seven Good Reasons for Mobile Agents", Communications of the ACM, ACM Press, vol 42, no 3, 1999, 88-89

[Lar98] Larsson J.E., Hayes-Roth B., "Guardian: An intelligent autonomous agent for medical monitoring and diagnosis", IEEE Intelligent Systems & Their Applications, vol 13, no 1, IEEE Computer Society Press, 1998, 58-64

[Law98] S. Lawrence, C.L. Giles, "Searching the World Wide Web", Science, AAAS, Vol 280, no 4, 1998, 98-100

[Lev98] Levy R. M., "Web Programming in Guide", Software-Practise and Experience, John Wiley & Sons, Ltd, vol. 28, no. 15, December 1998, pp. 1581-1603

[Les97] J. Lester, Stone B., "Increasing Believability in Animated Pedagogical Agents", In Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, California, February, 1997, 16-21

[Lie97] Lieberman H., "Autonomous Interface Agents", In Proceedings of the CHI 97: Human Factors in Computing Systems, ACM Press, 1997 67-74

[Mac97] Macedonia M., "A Taxonomy for Networked Virtual Environments", IEEE Multimedia, IEEE Press, 1997

[Mae89] Maes P., "How to Do the Right Thing", Connection Science Journal, Vol. 1, No. 3, Taylor & Francis Group, 1989, 291-323

[Mae93] Maes P., Kozierok R., "Learning interface agents", In Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI-93), MIT Press, Cambridge, Mass., 1993, 459-465

[Mae94] Maes P., "Agents that Reduce Work and Information Overload", Communications of the ACM, vol 37, no 7, ACM Press, 1994, 31-40

[Mae95] Maes P., "Artificial Life Meets Entertainment: Life like Autonomous Agents", Communications of the ACM, vol. 38, no. 11, ACM Press, 1995, 108-114

[Mae99] Maes P., Guttman H.R., Moukas G.A, "Agents That Buy and Sell", Communications of the ACM, ACM Press, vol 42, no 3, 1999, 81-91

[Mal95] Malone T.W., Lai K.Y., Fry C., "Experiments with Oval: A Radically Tailorable Tool for Cooperative Work", ACM Transactions on Information Systems, ACM Digital Library, vol. 13, no. 1, 1995, 177-205

[Mit96] Mitchell, M., "*An Introduction to Genetic Algorithms*", MIT Press, Cambridge, MA, 1996

[Mik97] Miksch S., Cheng K., Hayes-Roth B., "An intelligent assistant for patient health care", In Proceedings of the First International Conference on Autonomous Agents (Agents'97), ACM Press, New York, 1997, 458-465

[Mul95] Muller J.P., Pischel M., Thiel M., "Modeling Reactive Behaviour in Vertically Layered Agent Architectures", In Jennings N.R. and Wooldridge M.J. (eds), Intelligent Agents, Springer-Verlag, Berlin, 1995, 261-276

[Nas94] Nass C., Steuer J., Tauber E., "Computers are Social Actors", In Proceedings of CHI'94 - Human Factors in Computing Systems, ACM Press, 1994, 72-78

[Oar97] Oard D.W., "The State of the Art in Text Filtering", User Modeling and User-Adapted Interaction, Kluwer Academic Publishers, vol. 7, no 3, 1997, 141–178

[Pan99] Panayiotopoulos T., Zacharis Z.N., Vosinakis S., "Intelligent Guidance in a Virtual University", Advances in Intelligent Systems : Concepts, Tools and Applications, Tzafestas (ed), Kluwer Academic Publishers, Netherlands, 1999, 109-119

[Pan01] Panayiotopoulos T., Zacharis Z.N., "Machine Learning and Intelligent Agents", Lecture Notes in Artificial Intelligence 2049, Springer-Verlag, 2001, 281-286

[Pand01] Pandzic I., Babski C., Capin T., Lee W., Magnenat-Thalmann N, Raupp-Musse S., Moccozet L., Seo H., Thalmann D., "Simulating Virtual Humans in Networked Virtual Environments", PRESENCE: Teleoperators and Virtual Environments, MIT Press, vol 10, no 6, 2001, 632 - 646

[Par00] Parunak H.V.D., "A practitioners review of industrial agent applications", Autonomous Agents and Multi-Agent Systems, Kluwer Academic Publishers, vol 3, no 4, 2000, 389-407

[Paz97] Pazzani M., Billsus D., "Learning and Revising User Profiles: The Identification of Interesting Web Sites", Machine Learning, Kluwer Academic Publishers, vol 27, no 3, 1997, 313-331

[Rac99] Rachlin J., Goodwin R., Murthy S., Akkiraju R., Wu F., Kumaran S., Das R., "A-Teams: An agent architecture for optimization and decision support", Proceedings of the 5th International Workshop on Intelligent Agents V : Agent Theories, Architectures, and Languages (ATAL-98), Springer-Verlag: Heidelberg, Germany, 1999, 261-276

[Rao95] Rao A.S., Georgeff M.P., "BDI-agents: from theory to practice", In Proceedings of the First International Conference on Multiagent Systems, The MIT Press: Cambridge, MA, San Francisco, 1995, 312-319

[Rei97] G.L. Rein, D.L. McCue, J.A. Slein, "A Case for Document Management Function on the WEB", Communications of the ACM, ACM Press, 1997, Vol. 40, No 9, pp. 81-89

[Rod97] Rodriguez J., Noriega P., Sierra C., Padget J., "A Java-based Electronic Auction House", In Proceedings of the Second International Conference on Practical Applications of Intelligent Agents and MultiAgent Technology (PAAM'97), Practical Application Company, UK, 1997, 207-224

[Rus95] Russell, S.J., Peter Norvig, "Artificial Intelligence: A Modern Approach", Englewood Cliffs, Prentice Hall, NJ, 1995

[Sal71] G. Salton, "The *SMART* Retrieval System ", Prentice Hall, Upper Sandle River, N.J. 1971

[Sal83] G. Salton, M. McGill, "Introduction to modern information retrieval", McGraw-Hill Inc, NY,1983)

[Sca95] B.J. Scales, E.C. Felt. "Diversity on the World Wide Web: using robots to search the web", Library Software Review, Meckler Publishing, Vol 14, no 3, 1995, 132-136.

[Sch00] Schooten B.V., "Process and agent based modelling techniques for dialogue systems and virtual environments", Faculty of Computer Science, University of Twente, Technical Report 00-04, The Netherlands, 2000

[Seg00] Segal R., Kephart J.O., " Incremental Learning in SwiftFile ", In Proceedings of the Seventh International Conference on Machine Learning (ICML 2000), Morgan Kaufmann, San Francisco,2000, 863-870

[Sho93] Shoham Y, "Agent-oriented programming", Artificial Intelligence, AAAI Press, vol 60, no 1, 1993, 51–92

[Sho94] Shoham Y., "Artificial Intelligence Techniques in Prolog", Morgan Kaufmann Publishers, 1994

[Smi97] Smith J.R., Chang S.F., "Visually searching the Web for content", IEEE Multimedia Magazine, Vol. 4, no 3, IEEE Computer Society Publications Office, 1997, 12-20

[Smit94] Smith D.C., Cypher A., Spohrer J., "KidSim: Programming Agents Without a Programming Language", Communications of the ACM, ACM Press, vol. 37, no. 7, 1994, 55-67

[Son98] W. Sonnereich, T. Macinta, "Web Developer.com Guide to Search Engines", John Willey and Sons Inc, NY, 1998

[Spe98] E. Spertus, "ParaSite: Mining the Structural Information on the World-Wide Web", PhD Thesis, Department of EECS, MIT, Cambridge, MA, 1998

[Sta98] Stanney K.M., R.R. Mourant, Kennedy R.S., "Human Factors Issues in Virtual Environments: A Review of the Literature", PRESENCE: Teleoperators and Virtual Environments, MIT Press, vol 7, no 4, 1998, 327- 351

[Ste94] Sterling L., Shapiro E., "The Art of Prolog : Advanced Programming

Techniques", Mit Press, Cambridge, MA., 1994

[Stev94] R. W. Stevens, "TCP/IP Illustrated, Volume 1 : The Protocols", Addison-Wesley Professional Computing Series, Reading, MA, USA 1994

[Stev96] R. W. Stevens, " TCP/IP Illustrated, Volume 3 : Tcp for Transactions, Http, Nntp, and the Unix Domain Protocols", Addison-Wesley Professional Computing Series, Reading, MA, USA 1996

[The98] Theilmann W., Rothermel K., "Domain Experts for Information Retrieval in the World Wide Web", Proc. of 2nd Int. Workshop on Cooperative Informative Agents, Lecture Notes in Computer Science, Volume 1435, Spinger, 1998, pp 216-227

[Vie99] Viegas F., Donath J., "Chat Circles", In Proceedings of Computer Human Interaction, ACM Press, USA, 1999, 9-16

[Whi98] White T., Bieszczad A., Pagurek B., Sugar G., Tran X., "Intelligent network modeling using mobile agents", In Proceedings of the IEEE Global Telecommunications Conference GLOBECOM '98, IEEE Computer Society Press, Sydney, Australia, 1998, 1082-1087

[Wil96] J. Williams, "Bots and Other Internet Beasties", Sams Publishing, 1996

[Woo95a] Wooldridge M., Jennings N. R., "Agent Theories, Architectures, and Languages: a Survey", Intelligent Agents, Berlin: Springer-Verlag, 1995, 1-22

[Woo95b] Wooldridge M., Jennings N. R., "Intelligent Agents: Theory and Practice", Knowledge Engineering Review, vol 10, no 2, 1995, 115-152

[Woo97] Wooldridge M., "Issues in agent-based software engineering", In Proceedings of the First International Workshop on Cooperative Information Agents, volume 1202 of LNAI, Springer-Verlag: Heidelberg, 1997, 1-18

[Wri95] G. R. Wright, R. W. Stevens, "TCP/IP Illustrated, Volume 2 : The Implementation", Addison-Wesley Professional Computing Series, Reading, MA, USA 1995

[Yan99] Yan T.W., Garcia-Molina H., "The SIFT information dissemination system", ACM Transactions on Database Systems, ACM Press, vol 24, no 4, 1999, 529-565

[Zac98] Zacharis Z.N., Panayiotopoulos T., "An experimental comparison among genetic algorithms and annealing simplex", In Proceedings of 16th International Conference on Applied Informatics, M.H. Hamza (ed), Garmisch-Partenkirhen, Germany,1998, 141-143

[Zac99] Zacharis Z.N., Panayiotopoulos T.,"A Learning Personalized Information Agent for the WWW", In Proceedings of Workshop on Machine Learning and Intelligent Agents (ACAI-99), Chania, Greece, 1999, 39-44

[Zac00] Zacharis Z.N., Panayiotopoulos T., "A Two Level Genetic Based Query Optimization for Searching the WWW", In Proceedings of 4th World Multiconference on Systemics, Cybernetics and Informatics (SCAI 2000), Florida, USA, 2000, 450-456

[Zac01a] Zacharis Z.N., Panayiotopoulos T., "Web Search Using a Genetic Algorithm", Internet Computing, IEEE Computer Press, vol 5, no 2, 2001, 18-26

[Zac01b] Zacharis Z.N., Panayiotopoulos T., "A metagenetic algorithm for information filtering and collection from the World Wide Web", Expert Systems, Blackwell Publishers, vol 18, no 2, 2001, 99 - 108

[Zac02] Zacharis Z.N., Panayiotopoulos T., "SpiderServer : The MetaSearch Engine of WebNaut", In Proceedings of the 2nd Hellenic Conference on Artificial Intelligence (SETN-02), Thessaloniki, Greece, 475-486