ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

«ΥΛΟΠΟΙΗΣΗ ΛΟΓΙΣΜΙΚΟΥ ΑΡΑΧΝΗΣ ΓΙΑ ΑΝΑΖΗΤΗΣΗ ΠΟΛΙΤΙΣΤΙΚΟΥ ΠΕΡΙΕΧΟΜΕΝΟΥ ΣΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ»

ΣΤΕΦΑΝΟΣ ΦΙΛΙΟΣ
Α.Μ. 1312004140

ΕΠΙΒΛΕΠΟΝΤΕΣ ΚΑΘΗΓΗΤΕΣ
ΓΑΒΑΛΑΣ ΔΑΜΙΑΝΟΣ, ΤΣΕΚΟΥΡΑΣ ΓΙΩΡΓΟΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

ΤΜΗΜΑ ΠΟΛΙΤΙΣΜΙΚΗΣ ΤΕΧΝΟΛΟΓΙΑΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΣ

Ιούνιος 2008

Πίνακας Περιεχομένων

Πίνακας Εικόνων

Υλοποίηση λογισμικού αράχνης για αναζήτηση πολιτιστικού περιεχομένου στον παγκόσμιο ιστό

Στέφανος Φίλιος

Επιβλέποντες πτυχιακής: Γαβαλάς Δαμιανός, Τσεκούρας Γιώργος

Περίληψη

Το World Wide Web είναι η πιο διαδεδομένη και η πιο γρήγορα αναπτυσσόμενη, υπηρεσία του διαδικτύου. Οι χρήστες του έχουν πρόσβαση σε δισεκατομμύρια ιστοσελίδες, που περιλαμβάνουν πλούσιες πληροφορίες για μια ευρεία γκάμα θεμάτων. Αυτός ο τεράστιος όγκος πληροφοριών σε συνδυασμό με την αναρχία που επικρατεί στο διαδίκτυο, δημιούργησαν την επιτακτική ανάγκη για την ύπαρξη ενός μηχανισμού που θα έδινε μια πιο δομημένη μορφή, οργανώνοντας την διάσπαρτη πληροφορία. Αυτό τον ρόλο, ανέλαβαν οι μηχανές αναζήτησης, επιτρέποντας τον χρήστη να αναζητάει και να βρίσκει τις ιστοσελίδες με τα θέματα που τον ενδιαφέρουν. Υπάρχουν δυο είδη μηχανών, ανάλογα με τον αν αναζητούν πληροφορία για μια συγκεκριμένη θεματική ενότητα, ή όχι. Έτσι τις διαχωρίζουμε σε στοχευμένες μηχανές αναζητησης και σε γενικού σκοπού.

Στην εργασία αυτή, γίνεται η προσπάθεια παρουσίασης όλου του θεωρητικού υπόβαθρου που είναι απαραίτητο, για την όσο είναι το δυνατόν, καλύτερη κατανόηση της λειτουργίας και του τρόπου κατασκευής των μηχανών αναζήτησης. Ο κύριος στόχος της ήταν η κατασκευή ενός στοχευμένου λογισμικού αράχνης για πολιτιστικό περιεχόμενο. Τα λογισμικά αράχνης στην ουσία, είναι οι ανιχνευτές των μηχανών αναζήτησης. Ο βασικός τους στόχος, είναι να εντοπίζουν και να ανακτούν ιστοσελίδες στο διαδίκτυο και εν συνεχεία, να τις μεταβιβάζουν για αποθήκευση στο ευρετήριο της υπηρεσίας αναζήτησης. Προτού δημιουργήσουμε την αράχνη, επειδή θέλαμε να γνωρίζουμε και να προσδιορίζουμε εκ των προτέρων τις ιστοσελίδες στις οποίες θα αναζητήσει

πληροφορία, φτιάξαμε ένα πρόγραμμα αυτοματοποιημένης δημιουργίας ιστοσελίδων.

Το πρόγραμμα αυτό αντλεί λέξεις από δυο αρχεία κειμένου. Το πρώτο αρχείο περιέχει κοινότυπες λέξεις, ενώ το δεύτερο λέξεις πολιτιστικού περιεχομένου. Προγραμματίστηκε με τέτοιο τρόπο, ώστε ο χρήστης να μπορεί να ορίσει τον συγκεκριμένο αριθμό εγγράφων που επιθυμεί να δημιουργηθούν, την ποσότητα των λέξεων που θα περιέχει η κάθε ιστοσελίδα, καθώς και το συνολικό ποσοστό περιεχομένου σε πολιτιστικούς όρους, αντλώντας το πάντα από το πρώτο λεξικό. Μετά την δημιουργία των επιθυμητών τόσο σε αριθμό, όσο και σε περιεχόμενο ιστοσελίδων, ακολούθησε το «ανέβασμα» τους σε διακομιστή.

Για την αναζήτηση πολιτιστικής πληροφορίας σε όλες αυτές τις ιστοσελίδες, προγραμματίστηκε το λογισμικό της στοχευμένης αράχνης. Αρχικά είναι απαραίτητο να τοποθετηθεί (μέσω φόρμας) το αρχικό URL από το οποίο θα ξεκινήσει να αναζητά πληροφορία, καθώς και ο μέγιστος αριθμός ιστοσελίδων που θα επισκεφθεί, ακολουθώντας τους υπέρ-συνδέσμους τους. Η κύρια διεργασία της, είναι η καταγραφή και η σύγκριση των λέξεων που βρίσκει σε κάθε ιστοσελίδα, με τις λέξεις πολιτιστικού ενδιαφέροντος, που έχουμε ήδη φτιάξει και τοποθετήσει σε ένα txt αρχείο. Παρουσιάζοντας στο τέλος, με την ολοκλήρωση της αναζήτησης σε φθίνουσα σειρά, τα ποσοστά σχετικότητας της κάθε ιστοσελίδας σε σχέση με την θεματική που μας ενδιαφέρει.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω ιδιαιτέρως τους υπεύθυνους καθηγητές μου, κο. Γαβαλά Δαμιανό και κο. Τσεκούρα Γιώργο, για την καθοδήγηση και την πολύτιμη βοήθεια που μου προσέφεραν στην προσπάθεια ολοκλήρωσης αυτής της πτυχιακής. Τέλος, θα ήθελα να πω ένα πολύ μεγάλο ευχαριστώ στους γονείς μου και στον Θεό, γιατί χωρίς αυτούς στο πλευρό της ζωής μου, δεν θα είχα καταφέρει πραγματικά τίποτα.

1. Κεφάλαιο : Εισαγωγή

1.1 Διαδίκτυο

Το διαδίκτυο ή Internet όπως έχει επικρατήσει διεθνώς, είναι ένα παγκόσμιο δίκτυο υπολογιστών που αλληλοσυνδέονται μέσω της τηλεπικοινωνιακής υποδομής και επικοινωνούν μεταξύ τους, χρησιμοποιώντας ένα κοινό πρωτόκολλο επικοινωνίας, το TCP/IP (Transmission Control Protocol/ Internet Protocol).

Μετά το 1992, με την ραγδαία εξάπλωση που συνεχώς γνώριζε το διαδίκτυο, δεν άργησε και πολύ η καθιέρωση του ως ένα κορυφαίο μέσο επικοινωνίας. Υπάρχουν πολλοί λόγοι για την ραγδαία αυτή εξάπλωση του, κάποιοι από αυτούς είναι η ευκολία πρόσβασης, η ενσωμάτωση πολυμεσικών χαρακτηριστικών, η ποικιλία και το μέγεθος των πληροφοριών, η ευκολίες στην αναζήτηση τους, και τέλος το ότι η ανάπτυξη του γίνεται με τέτοιο τρόπο, ώστε να είναι όλο και πιο φιλικό όσο και προσιτό, στον χρήστη [1] [2].

Με την χρήση του διαδικτύου οι χρήστες μπορούν να έχουν πρόσβαση σε πληροφορίες που μέχρι πρότινος δεν είχαν την δυνατότητα, καθώς και να επικοινωνούν μεταξύ τους σε ελάχιστο χρόνο, ανεξάρτητα την απόσταση που τους χωρίζει. Το διαδίκτυο παρέχει πολλές υπηρεσίες στους χρήστες, όπως είναι το ηλεκτρονικό ταχυδρομείο, η μεταφορά αρχείων από έναν υπολογιστή σε έναν άλλον (με την χρήση του πρωτοκόλλου ftp), σύνδεση με έναν απομακρυσμένο υπολογιστή (telnet) με δυνατότητα ενεργοποίησης διαφόρων προγραμμάτων από απόσταση, κ.λ.π. Μια όμως από τις πιο σημαντικές και γνωστές υπηρεσίες του διαδικτύου, είναι ο Παγκόσμιος Ιστός (World Wide Web, WWW) [1].

Ο όγκος των πληροφοριών που είναι διαθέσιμος μέσω του WWW είναι τεράστιος και συνεχώς αυξανόμενος. Με αποτέλεσμα οι χρήστες να χάνονται και να μην μπορούν να εκμεταλλευτούν στο έπακρο, τον πλούτο των διαθέσιμων πληροφοριών. Τη λύση σε αυτό το πρόβλημα ήρθαν να δώσουν οι υπηρεσίες αναζήτησης. Εμφανίστηκαν αρχικά στα μέσα της δεκαετίας του 90 και σήμερα αποτελούν απαραίτητα και χρήσιμα εργαλεία για τους χρήστες του διαδικτύου. Στις μέρες μας, για τους περισσότερους χρήστες του διαδικτύου είναι πραγματικά δύσκολο να φανταστούν την περιήγηση τους στον κυβερνοχώρο χωρίς την βοήθεια των υπηρεσιών αναζήτησης. Παρόλα αυτά πρέπει να

λαμβάνουν υπόψη τους, ότι ενδέχεται να αναζητούν πληροφορίες που δεν είναι πάντα διαθέσιμες [3].

1.2 Στόχοι και γενική μεθοδολογία της εργασίας

Ο κύριος στόχος της εργασίας αυτής όπως έχουμε προαναφέρει, ήταν η δημιουργία ενός στοχευμένου λογισμικού αράχνης για πολιτιστικό περιεχόμενο. Η παρούσα εργασία, επιχειρεί να παρουσιάσει στους αναγνώστες όλες τις απαραίτητες πληροφορίες που χρειάζονται, για να προσδιοριστούν τα είδη μηχανών αναζήτησης, καθώς και για να γίνει αντιληπτός ο τρόπος λειτουργίας και κατασκευής, τόσο των στοχευμένων όσο και του γενικού σκοπού μηχανών. Προτού προχωρήσουμε στην κατασκευή της αράχνης, χρειάσθηκε να φτιάξουμε ένα πρόγραμμα για την αυτοματοποιημένη δημιουργία ιστοσελίδων με συγκεκριμένο πολιτιστικό περιεχόμενο και μέγεθος, ορισμένα πάντα, από τον χρήστη της εφαρμογής. Η διαμόρφωση του περιεχομένου γίνεται αντλώντας λέξεις, από δυο λεξικά. Το ένα εμπεριέχει λέξεις πολιτιστικού περιεχομένου και το άλλο κοινές / γενικές λέξεις. Στην συνέχεια με την κατασκευή των ιστοσελίδων προς αναζήτηση και το «ανέβασμα» τους στο διαδίκτυο, ακολούθησε ο προγραμματισμός της αράχνης. Είναι μια αρκετά πολύπλοκη διαδικασία, που χρειάζεται ιδιαίτερη προσοχή και αρκετούς ελέγχους, για την αποφυγή σημαντικών σφαλμάτων.

Το λογισμικό αράχνης δέχεται από τον χρήστη το αρχικό URL (Uniform Resource Locator) της σελίδας, από την οποία θα ξεκινήσει την αναζήτηση. Ακολουθώντας τα URL που συναντά στον HTML (HyperText Markup Language) κώδικα, πηγαίνει από την μια ιστοσελίδα στην άλλη, αναζητώντας και συγκρίνοντας όλες τις λέξεις ενδιαφέροντος, σχετικά με την θεματική που αναζητάμε. Στο τέλος της διαδικασίας, παρουσιάζονται τα ποσοστά και οι υπέρ-συνδέσεις των ποιον σχετικών ιστοσελίδων που έχει βρει, ιεραρχημένα σε φθίνουσα σειρά.

Τέλος πρέπει να αναφερθεί ότι όλοι οι στόχοι που είχαν τεθεί κατά την φάση της σχεδίασης και προγραμματισμού της πτυχιακής εργασίας, ολοκληρώθηκαν επιτυχώς. Βέβαια, σε περίπτωση που περισσότερος χρόνος ήταν διαθέσιμος, θα μπορούσε να είχε

επεκταθεί περαιτέρω, προσθέτοντας περισσότερες δυνατότητες και κάνοντας την ακόμη πιο λειτουργική.

1.3 Περίληψη των κεφαλαίων

Στο Κεφάλαιο 2 παραθέτουμε γενικές πληροφορίες σχετικές με το Διαδίκτυο, τον παγκόσμιο ιστό, την HTML (HyperText Markup Language) και τα CSS (Cascaded Style Sheets), τις διάφορες τεχνολογίες για ανάπτυξη ιστό-τόπων και γιατί επιλέχθηκε για την υλοποίηση της εφαρμογής μας, η αντικειμενοστραφής γλώσσα Java. Στο Κεφάλαιο 3, μιλάμε για τα διάφορα είδη μηχανών αναζήτησης που έχουμε, τα μέρη από τα οποία αποτελούνται, αλλά και πως λειτουργούν αυτά μεταξύ τους. Παραθέτοντας και κάποια στατιστικά στοιχεία των πιο μεγάλων μηχανών αναζήτησης. Στο Κεφάλαιο 4, γίνεται η παρουσίαση του ψευδοκώδικα, της λειτουργίας, αλλά και του τρόπου κατασκευής, της στοχευμένης αράχνης. Το Κεφάλαιο 5 αναφέρει γενικά συμπεράσματα για την εργασία, δίνοντας επιπλέον, κατευθύνσεις για μελλοντική έρευνα. Τέλος, στο παράρτημα της παρούσας εργασίας, παραθέτουμε ενδεικτικά ένα μέρος από τον κώδικα του λογισμικού αράχνης.

2. Κεφάλαιο : Γενικές πληροφορίες για το Διαδίκτυο, τον Παγκόσμιο Ιστό και τις τεχνολογίες του

Στο κεφάλαιο αυτό προσδιορίζουμε το διαδίκτυο και τον κυβερνοχώρο, παρουσιάζοντας επίσης τις διάφορες τεχνολογίες που χρησιμοποιούν. Τέλος αναφερόμαστε στην γλώσσα σήμανσης HTML, στα CSS, αλλά και στην αντικειμενοστραφής γλώσσα Java.

2.1 Το διαδίκτυο

Το δημόσιο Διαδίκτυο όπως έχουμε ήδη προαναφέρει, είναι ένα παγκόσμιο δίκτυο υπολογιστών, δηλαδή ένα δίκτυο που διασυνδέει εκατομμύρια υπολογιστικές συσκευές σε όλο τον κόσμο. Οι περισσότερες από αυτές, είναι παραδοσιακά επιτραπέζια PC, σταθμοί εργασίας UNIX και οι καλούμενοι διακομιστές (servers) , που αποθηκεύουν και μεταδίδουν πληροφορίες, όπως είναι οι ιστοσελίδες και τα μηνύματα e-mail. Όλο και περισσότερο, μη παραδοσιακά τερματικά συστήματα, όπως PDAs (προσωπικοί ψηφιακοί βοηθοί), τηλεοράσεις, αυτοκίνητα, ψυγεία κτλ. συνδέονται στο Διαδίκτυο. Στην ορολογία του Διαδικτύου, όλες αυτές οι συσκευές ονομάζονται υπολογιστές υπηρεσίας (hosts) ή τερματικά συστήματα (end systems). Τον Ιανουάριο του 2008 (σύμφωνα με στατιστικά του ISC), υπήρχαν κοντά στα 540 εκατομμύρια τερματικά συστήματα που χρησιμοποίησαν το Διαδίκτυο και αυτός ο αριθμός συνεχίζει να αυξάνεται εκθετικά [12].

Η λέξη διαδίκτυο, προέρχεται από την ένωση των λέξεων INTERconnection (διασύνδεση) και NETwork (δίκτυο). Έτσι λέμε Διαδίκτυο το διασυνδεδεμένο δίκτυο, το οποίο χρησιμοποιεί το πρωτόκολλο IP (Internet Protocol) και κατ' επέκταση το σύνολο όλων αυτών των δικτύων που συνεργάζονται, για να σχηματίσουν ένα μοναδικό εικονικό δίκτυο στους χρήστες του.

Το πρωτόκολλο IP σχεδιάστηκε για την υλοποίηση ενός δικτύου διασύνδεσης άλλων δικτύων (Διαδίκτυο). Οι υπολογιστές συνδέονται αρχικά μεταξύ τους, σε ένα τοπικό δίκτυο LAN (Local Area Network) και στην συνέχεια με την χρήση μιας συσκευής που ονομάζεται δρομολογητής (rooter), παρέχεται η σύνδεση μέσω του πρωτοκόλλου IP του τοπικού δικτύου με τον υπόλοιπο κόσμο του Διαδικτύου. Τα δεδομένα αποστέλλονται με την μορφή πακέτων, τα οποία μεταφέρονται από κόμβο σε κόμβο,

χωρίς να ακολουθείται προσχεδιασμένη διαδρομή. Ένα πακέτο αποτελείται από τρία βασικά μέρη, την διεύθυνση του παραλήπτη, την διεύθυνση του αποστολέα και τα δεδομένα που είναι να αποσταλούν. Επίσης διαχωρίζουμε τις IP διευθύνσεις σε στατικές και δυναμικές. Στατική IP οφείλουν να έχουν όσοι υπολογιστές είναι συνεχώς συνδεδεμένοι στο Διαδίκτυο (π.χ. web servers), ενώ δυναμικό IP παίρνουν οι υπολογιστές οι οποίοι συνδέονται ευκαιριακά και για σύντομο χρονικό διάστημα. Επειδή οι διευθύνσεις IP προκαλούν δυσκολίες στην χρήση τους και στην απομνημόνευση τους, χρησιμοποιούμε παράλληλα και το σύστημα DNS (Domain Name System) το οποίο δημιουργεί αντιστοιχίες μεταξύ των διευθύνσεων IP και των διαφόρων ονομάτων [14].

2.2 Παγκόσμιος Ιστός (World Wide Web)

Μέχρι και τις αρχές της δεκαετίας του 1990, στο Internet βρίσκονταν κυρίως ακαδημαϊκοί, κρατικοί οργανισμοί και βιομηχανικοί ερευνητές. Μια νέα εφαρμογή, ο Παγκόσμιος Ιστός ή WWW (World Wide Web) άλλαξε τα πάντα και έφερε εκατομμύρια νέους, μη ακαδημαϊκούς, χρήστες στο δίκτυο. Ο Παγκόσμιος Ιστός είναι μια αρχιτεκτονική, για την προσπέλαση διασυνδεδεμένων εγγράφων μέσω υπέρ-συνδέσμων (hyperlinks), τα οποία κατανέμομαι σε εκατομμύρια μηχανές σε ολόκληρο το Διαδίκτυο. Η τεράστια δημοτικότητα του, οφείλεται στο ότι έχει μια πολύχρωμη και κομψή διασύνδεση γραφικών που είναι εύκολη στη χρήση από αρχάριους, καθώς και στο ότι παρέχει ένα τεράστιο πλούτο πληροφοριών, για οποιαδήποτε σχεδόν θέμα μπορεί να αναζητά κανείς.

Ο Ιστός ξεκίνησε το 1989 στο CERN, το ευρωπαϊκό κέντρο πυρηνικής έρευνας, μέσα από την ανάγκη που είχαν μεγάλες ομάδες διάσπαρτων διεθνών ερευνητών για να συνεργάζονται, χρησιμοποιώντας μια συνεχώς μεταβαλλόμενη συλλογή αναφορών, ιδεών, φωτογραφιών και άλλων εγγράφων. Η αρχική πρόταση για έναν ιστό συνδεδεμένων εγγράφων, έγινε από το φυσικό του CERN, Tim Berners-Lee το Μάρτιο του 1989. Το πρώτο πρωτότυπο που βασιζόταν σε κείμενο, ξεκίνησε την λειτουργία του 18 μήνες αργότερα. Αν και υπήρχαν άφθονες και συναρπαστικές πληροφορίες στο Δίκτυο, ο τρόπος με τον οποίο μπορούσες να τις δεις, δεν ήταν ιδιαίτερα όμορφος. Για την πρόσβαση του χρήστη σε όλες αυτές τις πληροφορίες, ήταν απαραίτητο να κατέχει

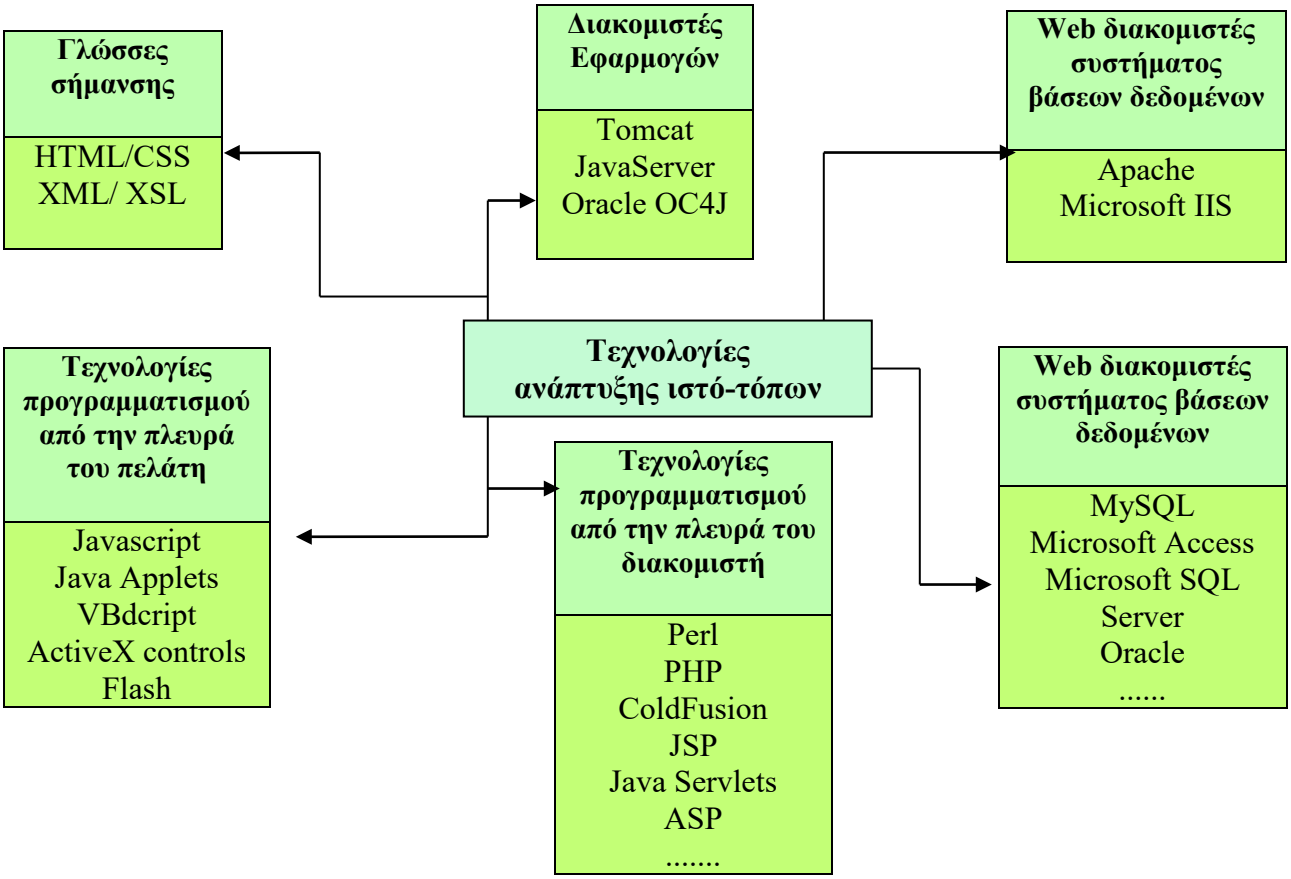
αρκετά εξειδικευμένες γνώσεις, με ικανότητα χειρισμού διαφόρων εντολών, παραμέτρων και πρωτόκολλων επικοινωνίας. Έτσι δημιουργήθηκε η ανάγκη για την δημιουργία ενός πιο εύχρηστου και προσιτού περιβάλλοντος, για να μπορεί να έχει πρόσβαση και ο μέσος χρήστης του διαδικτύου. Η επανάσταση έγινε με την εμφάνιση του Mosaic, της πρώτης εφαρμογής browser με γραφικό περιβάλλον, το Φεβρουάριο του 1993. Έτσι άνοιξε ο δρόμος για την εμφάνιση έγχρωμου κειμένου και γραφικών στις ιστοσελίδες. Αυτό είναι ένα από τα πιο ισχυρά σημεία του Web και αναμφίβολα, ο λόγος για τον οποίο έγινε τόσο δημοφιλές [13] [15].

Η πρόσβαση των χρηστών του web σε όλα τα ηλεκτρονικά έγγραφα που περικλείει, βασίζεται στο μοντέλο πελάτη/διακομιστή (Client/Server).Στο μοντέλο αυτό, ένας η περισσότεροι πελάτες, αποστέλλουν αιτήσεις ζητώντας την παροχή υπηρεσιών και πληροφοριών από ένα διακομιστή. Στο χώρο του διαδικτύου, πελάτες ονομάζουμε τα προγράμματα πλοήγησης (ή αλλιώς φυλλομετρητές), τα οποία επιτρέπουν την εμφάνιση του περιεχομένου των ιστοσελίδων. Ενώ διακομιστές, ονομάζουμε τους web διακομιστές. Τα προγράμματα πελάτες, επικοινωνούν με τους διακομιστές οι οποίοι φιλοξενούν τις ιστοσελίδες, με την βοήθεια των πρωτοκόλλων επικοινωνίας. Το κύριο πρωτόκολλο επικοινωνίας, είναι το HTTP (HyperText Transfer Protocol) το οποίο δημιουργήθηκε από τον Tim Berners-Lee, το 1990. Η κύρια διεργασία του HTTP είναι να ρυθμίζει τη μεταφορά αιτήσεων από τον περιηγητή στον διακομιστή και προς την αντίθετη κατεύθυνση. Ο μεγαλύτερος όγκος της πληροφορίας που διακινείται στο διαδίκτυο μέσω του HTTP, γίνεται χωρίς καμιά κωδικοποίηση περιεχομένου. Έτσι, τα δεδομένα μεταφέρονται από τον διακομιστή στον υπολογιστή του χρήστη, με την μορφή απλού κειμένου. Για την βελτίωση της ασφάλειας των δεδομένων, κατά την διάρκεια μεταφοράς τους στο διαδίκτυο, η εταιρεία Netscape πρότεινε και δημιούργησε ένα νέο πρωτόκολλο επικοινωνίας με ονομασία HTTPS (HyperText Transfer Protocol Secure). Το πρωτόκολλο αυτό προβλέπει επικοινωνία με αυθεντικοποίηση (authendication), καθώς και κρυπτογράφηση του περιεχομένου της ιστοσελίδας (encrypted) [14] [15].

2.3 Βασική ταξινόμηση των τεχνολογιών ανάπτυξης ιστοτόπων

Κατά την ανάπτυξη των διαδικτυακών τόπων, χρησιμοποιείται ένα ευρύ πεδίο τεχνολογιών που απεικονίζεται στην Εικόνα 1.

- γλώσσες σήμανσης όπως είναι η HTML και η XML καθώς και συνοδευτικές τεχνολογίες μορφοποίησης των περιεχομένων HTML και XML εγγράφων (CSS και XSL αντίστοιχα)
- γλώσσες για την ανάπτυξη προγραμμάτων που εκτελούνται στον περιηγητή (JavaScript, VBscript, Flash, Java Applets)
- γλώσσες για την ανάπτυξη προγραμμάτων που εκτελούνται από την πλευρά του διακομιστή (PHP, JSP, Java Servlets, ASP)
- συστήματα διαχείρισης βάσεων δεδομένων (MySQL, MS SQL Server, Oracle)
- web διακομιστές (Apache, Microsoft IIS)
- διακομιστές εφαρμογών (Tomcat, JavaServer)
- Web διακομιστές συστήματος βάσεων δεδομένων



Εικόνα 1: Ταξινόμηση των τεχνολογιών ανάπτυξης ιστοτόπων [15]

2.4 HTML (HyperText Markup Language) και CSS (Cascaded Style Sheets)

Το μεγάλο μέγεθος της πληροφορίας που είναι διαθέσιμη σήμερα στο διαδίκτυο, καθιστά απαραίτητη την ευκολία πρόσβασης και μεταφοράς των ηλεκτρονικών εγγράφων. Επίσης λόγω της πληθώρας διαφορετικών συστημάτων, τα έγραφα χρειάζεται να είναι ανεξάρτητα οποιοδήποτε συστήματος. Για την επίλυση αυτών των προβλημάτων, αναπτύχθηκαν οι γλώσσες σήμανσης.

Μια γλώσσα σήμανσης (όπως είναι η HTML), αποτελείται από ένα σύνολο εντολών, βοηθώντας μας στον προσδιορισμό της δομής και του τρόπου παρουσίασης, ενός εγγράφου. Επιτρέπει να δομήσουμε και να εμφανίσουμε διάφορα αντικείμενα στην οθόνη του υπολογιστή. Αυτά τα αντικείμενα μπορούν να είναι κείμενο, πίνακες, εικόνες και ήχος. Η εμφάνιση όλων αυτών των αντικείμενων στην οθόνη, λέγεται σελίδα ή ιστοσελίδα (web page). Ο κώδικας μιας ιστοσελίδας, αποτελείται από μια σειρά ειδικών κωδικών (tags) τοποθετημένων μέσα στο κείμενο και τα οποία επιτρέπουν να μορφοποιήσουμε το κείμενο και να προσθέσουμε συνδέσεις προς άλλες ιστοσελίδες. Οι κωδικοί (tags) λέγονται και σημειώσεις ή ετικέτες.

Οι υπάρχουσες γλώσσες σήμανσης, αποτελούν «απόγονους» της μετά-γλώσσας SGML (Structured Generalized Markup Language). Η SGML αποτελείται από ένα γενικευμένο σύνολο κανόνων που χρησιμοποιείται για την περιγραφή των γλωσσών σήμανσης. Τόσο η SGML όσο και οι άλλες μετά-γλώσσες και γλώσσες σήμανσης, έχουν προτυποποιηθεί από το W3C (World Wide Web Consortium), έναν ανεξάρτητο φορέα που έχει σαν στόχο τη καθιέρωση προτύπων και προδιαγραφών στο περιβάλλον του παγκόσμιου ιστού.

Η HTML(HyperText Markup Language, Γλώσσα Σήμανσης Υπερκειμένου) είναι η πιο βασική γλώσσα προγραμματισμού με την οποία μπορούμε να κατασκευάσουμε και να παρουσιάσουμε περιεχόμενο στον παγκόσμιο ιστό. Ουσιαστικά αποτελεί το πρότυπο που κυριαρχεί σήμερα στο διαδίκτυο. Οι ιστοσελίδες που εμφανίζονται στις οθόνες των περιηγητών, δομούνται σύμφωνα με τις προδιαγραφές που ορίζει η HTML. Η φιλοσοφία της, βασίζεται στην έννοια του υπερκείμενου. Δηλαδή στην ύπαρξη των λεγόμενων

υπέρ-συνδέσμων για την σύνδεση της μιας ιστοσελίδας με μια άλλη, δημιουργώντας ένα πλέγμα διασυνδεδεμένων εγγράφων.

Τα CSS (Cascading Style Sheets, επικαλυπτόμενα φύλλα στυλ) συνήθως αποτελούν ξεχωριστά αρχεία ή τμήματα ενός HTML εγγράφου, ορίζοντας το στυλ εμφάνισης διαφόρων στοιχείων, που εμπεριέχονται μέσα σε ένα HTML έγγραφο. Όταν ένα φύλλο στυλ εφαρμοστεί σε ένα στοιχείο, αυτόματα θα επηρεάσει και όλα τα αντίστοιχα στοιχεία του εγγράφου. Ο ρόλος των φύλων στυλ, είναι η απλοποίηση της δημιουργίας και συντήρησης των ιστοσελίδων, απαλείφοντας την ανάγκη επαναληπτικής προσθήκης ετικετών, με λεπτομερείς εντολές μορφοποίησης. Έτσι έχουμε την δυνατότητα, να τροποποιήσουμε εύκολα την εμφάνιση συγκεκριμένων στοιχείων του εγγράφου, κάνοντας απλά κάποιες αλλαγές στα αντίστοιχα φύλλα στυλ [15] [14].

2.5 Γλώσσα προγραμματισμού Java

Η Java είναι μια από τις πιο δημοφιλές γλώσσες αντικειμενοστραφούς προγραμματισμού. Δημιουργήθηκε από την Sun Microsystems, έχοντας ως βάση το μοντέλο της C++. Από το 1995 που η Sun Microsystems εξέδωσε την πρώτη έκδοση της γλώσσας προγραμματισμού Java, η τεχνολογία της έχει γίνει μια ιδιαίτερα δημοφιλής γλώσσα και έχει υιοθετηθεί από εκατομμύρια προγραμματιστές, για δημιουργία στιβαρών και ασφαλών εφαρμογών. Η Java σχεδιάστηκε με στόχο ο κώδικάς της να είναι απλός, μικρού μεγέθους και μεταφέρσιμος μεταξύ διαφορετικών υπολογιστικών αρχιτεκτονικών και λειτουργικών συστημάτων. Έτσι, χρησιμοποιώντας την γλώσσα Java για την δημιουργία διαφόρων εφαρμογών, έχουμε το μεγάλο πλεονέκτημα της ανεξαρτησίας της πλατφόρμας. Οι εφαρμογές που φτιάχνουμε, μπορούν να εκτελεστούν πρακτικά σε κάθε συσκευή, ανεξαρτήτως του λειτουργικού συστήματος που χρησιμοποιούν. Τα προγράμματα Java μεταγλωττίζονται σε μια μορφή, που καλείται bytecode και εκτελείται από κάθε λειτουργικό σύστημα, λογισμικό, ή συσκευή, σε διερμηνευτή Java. Το μόνο που είναι απαραίτητο για να εκτελεστεί ο Java κώδικας, είναι να υπάρχει εγκατεστημένη η εικονική μηχανή Java (Java Virtual Machine, JVM).

Τα προγράμματα Java διακρίνονται σε δυο κύριες κατηγορίες:

- τις εφαρμογές (applications) που έχουν παρόμοια λογική και λειτουργία με εφαρμογές άλλων γλωσσών προγραμματισμού (δυνατότητες για διαχείριση αρχείων, εκτύπωσης, άνοιγμα και διαχείριση δικτυακών συνδέσεων, κ.λ.π.)

- τα applets τα οποία «κατεβαίνουν» από web διακομιστές σε web περιηγητές, όπου και εκτελούνται. Τα applets είναι ένας από τους κύριους λόγους που έγινε τόσο δημοφιλής η γλώσσα Java. Μπορούν να εκτελέσουν τις περισσότερες λειτουργίες των εφαρμογών Java, παρόλα αυτά για λόγους ασφάλειας, δεν έχουν πρόσβαση στο τοπικό σύστημα αρχείων [15].

3. Κεφάλαιο : Μηχανές αναζήτησης

Το κεφάλαιο αυτό, αναφέρεται κυρίως στα είδη και στις βασικές λειτουργίες τις οποίες επιτελούν οι μηχανές αναζήτησης. Παρουσιάζονται στατιστικά στοιχεία των μεγαλύτερων μηχανών αναζήτησης και τέλος δίνεται περισσότερη έμφαση στις στοχευμένες μηχανές αναζήτησης και στις κυριότερες διαφορές που εμφανίζουν, σε σχέση με εκείνες του γενικού σκοπού.

3.1 Εισαγωγή

Σε εκατοντάδες εκατομμύρια ανέρχεται σήμερα ο αριθμός των ιστοσελίδων στο διαδίκτυο, ενώ συνεχώς προστίθενται ολοένα και περισσότερες. Για έναν απλό χρήστη, που προσπαθεί μόνος του, χωρίς την χρήση κάποιου βοηθητικού λογισμικού αναζήτησης, να ανακτήσει από τον κυβερνοχώρο τις πληροφορίες που χρειάζεται σχετικά με ένα θέμα που τον ενδιαφέρει, είναι εξαιρετικά δύσκολο και χρονοβόρο. Ο χρήστης θα περιπλανιέται για αρκετές ώρες στο Ίντερνετ άσκοπα, ενώ πολλές από τις ιστοσελίδες “ενδιαφέροντος” μένουν ανεξερεύνητες. Την επίλυση του προβλήματος ήρθαν να φέρουν οι μηχανές αναζήτησης, που πραγματοποιούν για λογαριασμό του χρήστη την αναζήτηση σε όλο το διαδίκτυο.

Οι μηχανές αναζήτησης είναι προγράμματα που επιτρέπουν την αναζήτηση και ανάκτηση πληροφορίας με λέξεις κλειδιά, σε τεράστιες βάσεις δεδομένων αρχείων του διαδικτύου. Οι βάσεις αυτές περιλαμβάνουν ένα ευρετήριο, με το πλήρες κείμενο των ιστοσελίδων. Υπάρχουν ειδικά προγράμματα με διάφορες ονομασίες (spider, web crawler, robot κλπ.) όπου συλλέγουν και αποθηκεύουν με αυτοματοποιημένο τρόπο αντίγραφα εκατομμυρίων ιστοσελίδων του παγκόσμιου ιστού. Από τα αρχεία που συγκεντρώνονται και πιο συγκεκριμένα από τον τίτλο τους, το URL τους, το μέγεθος, το πλήρες τους κείμενο, κ.λ.π. δημιουργείται το ευρετήριο που προαναφέραμε. Όταν ένας χρήστης αναζητά πληροφορία σε μια μηχανή αναζήτησης, στην πραγματικότητα ερευνά την βάση δεδομένων των καταχωρημένων ιστοσελίδων που έχουν αποθηκευτεί νωρίτερα από την υπηρεσία αναζήτησης και όχι το ίδιο το WWW [3].

3.2 Τα είδη των μηχανών αναζήτησης

Υπάρχουν πολλοί τρόποι διαφοροποίησης των μηχανών αναζήτησης, ένας από αυτούς είναι με βάση τα αποτελέσματα που δίνουν στον χρήστη. Έτσι μπορούμε να τις διαχωρίσουμε σε μηχανές αναζήτησης γενικού ενδιαφέροντος και στις στοχευμένες. Η κύρια διαφορά μεταξύ τους, είναι ότι η στοχευμένες μηχανές αναζήτησης, επικεντρώνονται σε συγκεκριμένα θέματα. Προσπαθούν να βρουν και να καταγράψουν όσες περισσότερες ιστοσελίδες μπορούν για μια συγκεκριμένη θεματική, επισκεπτόμενες ένα περιορισμένο αριθμό δικτυακών τόπων που καλύπτουν το συγκεκριμένο θέμα. Σε αντίθεση με αυτές που είναι γενικού ενδιαφέροντος, οι οποίες προσπαθούν να καταγράψουν όσο το δυνατόν μεγαλύτερο τμήμα των ιστοσελίδων του διαδικτύου, ανεξαρτήτως θεματικής.

Επίσης παρά το γεγονός ότι οι μηχανές αναζήτησης δεν έχουν πολλά χρόνια λειτουργίας από την πρώτη τους εμφάνιση, λόγο της συνεχής ανάπτυξης της τεχνολογίας, αλλά και κάτω από την πίεση του ανταγωνισμού, έχουν διαφοροποιηθεί σε επιπλέον δυο κατηγορίες. Έτσι τις διακρίνουμε σε μηχανές αναζήτησης πρώτης και δεύτερης γενεάς. Πρώτης γενεάς, είναι αυτές που συσχετίζουν και παρουσιάζουν τα αποτελέσματα, με βάση το ποσοστό συνάφειας τους, ενώ δεύτερης γενεάς είναι οι μηχανές που μπορούν να παρουσιάσουν και να ιεραρχήσουν τα αποτελέσματα με ποικίλους τρόπους, όπως είναι η ιεράρχηση των αποτελεσμάτων σύμφωνα με την δημοτικότητα τους, σύμφωνα με το είδος ή τον τύπο των τεκμηρίων, η ακόμα και να δεχτούν ερωτήσεις σε φυσική γλώσσα και να δώσουν αποτελέσματα που έχουν καθοριστεί εκ των προτέρων [3].

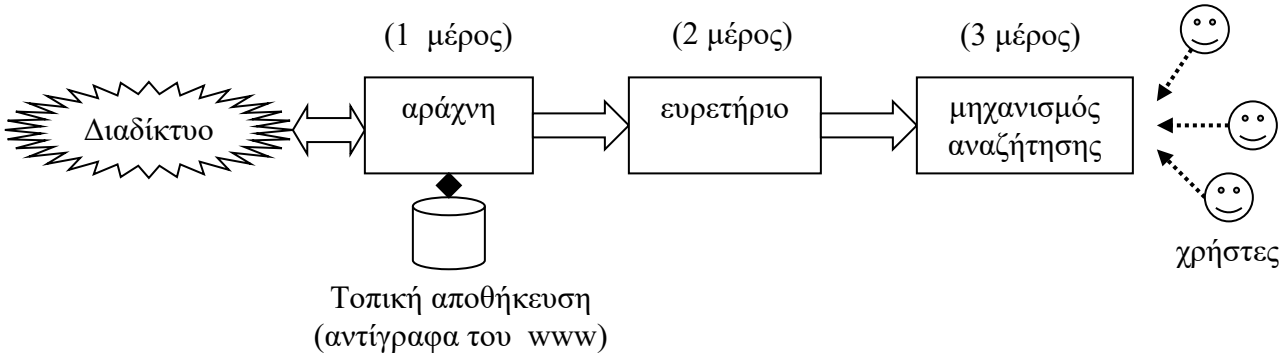
Ακόμη μια άλλη, ξεχωριστή κατηγορία στις μηχανές αναζήτησης, είναι οι Μετά-μηχανές. Σε αντίθεση με τις απλές, οι οποίες χρησιμοποιούν ένα λογισμικό αράχνης για την δημιουργία της βάσης δεδομένων τους, οι μετά-μηχανές δεν διαθέτουν δικό τους ευρετήριο, αλλά αντλούν τα αποτελέσματα τους από τα ευρετήρια άλλων μηχανών αναζήτησης. Έτσι σε κάθε αναζήτηση που γίνεται, στέλνουν τις λέξεις-κλειδιά ταυτοχρόνως σε μια σειρά προκαθορισμένων υπηρεσιών αναζήτησης. Ο μηχανισμός

αναζήτησης παραμένει λίγο χρόνο στο ευρετήριο κάθε βάσης και επιστρέφει ένα συγκεκριμένο ποσοστό των αποτελεσμάτων (συχνά μόνο το 10%) από κάθε βάση [3]. Η ιδέα της μετά-μηχανής μπορεί να είναι πάρα πολύ καλή, όμως η υλοποίηση της δεν φέρνει πάντοτε και τα καλύτερα αποτελέσματα. Αν και κερδίζουμε χρόνο από την χρήση τέτοιων υπηρεσιών, παρόλα αυτά έχει αποδειχθεί πολλές φορές ότι τα αποτελέσματα δεν είναι πάντοτε ικανοποιητικά. Αυτό οφείλεται όπως προαναφέραμε, στο γεγονός ότι επιστρέφουν ένα συγκεκριμένο ποσοστό αποτελεσμάτων από κάθε μηχανή αναζήτησης, με αποτέλεσμα να μην γίνεται πάντοτε με σωστό τρόπο, η επιλογή της πληροφορίας που χρειάζεται να αντληθεί.

3.3 Βασικές λειτουργίες των μηχανών αναζήτησης

Υπάρχουν σίγουρα διαφορές στο τρόπο λειτουργίας διαφόρων μηχανών αναζήτησης, ανάλογα με το είδος τους, όμως όλες τους επιτελούν τρεις βασικές λειτουργίες :

1. Αναζητούν και συλλέγουν συγκεκριμένες ιστοσελίδες του διαδικτύου, με βασικό κριτήριο σημαντικές σημασιολογικά λέξεις.
2. Δημιουργούν και διατηρούν ένα ευρετήριο με τις λέξεις και την τοποθεσία που τις βρίσκουν (URL).
3. Επιτρέπουν στους χρήστες να αναζητήσουν λέξεις, ή συνδυασμό λέξεων στο ευρετήριο της μηχανής αναζήτησης [4].



Εικόνα 2: Τα μέρη μιας μηχανής αναζήτησης [8]

3.4 Βασικά μέρη μηχανής αναζήτησης

Μια μηχανή αναζήτησης αποτελείται από τρία βασικά μέρη:

3.4.1 Η αράχνη (spider)

Τα προγράμματα αράχνες που ονομάζονται επίσης και web crawlers ή robots, είναι οι ανιχνευτές των μηχανών αναζήτησης. Ο βασικός τους στόχος είναι να εντοπίζουν και να ανακτούν ιστοσελίδες στο διαδίκτυο και εν συνεχεία, να τις μεταβιβάζουν για αποθήκευση στο ευρετήριο της υπηρεσίας αναζήτησης.

Οι αράχνες βρίσκουν ιστοσελίδες με δυο μεθόδους :

- Από φόρμες καταχώρησης διεύθυνσης (add URL) που διαθέτουν οι περισσότερες μηχανές αναζήτησης, επιτρέποντας σε κατασκευαστές και διαχειριστές ιστοσελίδων, να ενημερώνουν τις υπηρεσίες αναζήτησης, σχετικά με τις διευθύνσεις των ιστοσελίδων τους. Στα πρώτα χρόνια λειτουργίας των μηχανών αναζήτησης, αυτή η μέθοδος ήταν αποτελεσματική και έδινε αποτελέσματα, καθώς η αράχνες δέχονταν τις λίστες με τις διευθύνσεις και στην συνέχεια τις επισκεπτόντουσαν. Δυστυχώς στις μέρες μας χρησιμοποιείται από κακόβουλους χρήστες, οι οποίοι δημιουργούν λογισμικό που με αυτοματοποιημένο τρόπο, στέλνει εκατομμύρια αιτήσεις καταχώρησης ιστοσελίδων. Οι περισσότερες μηχανές αναζήτησης, απορρίπτουν κοντά στο 95% των καταχωρημένων νέων διευθύνσεων [3].
- Χρησιμοποιώντας την ύπαρξη υπέρ-συνδέσεων, στις ιστοσελίδες που επισκέπτεται και ερευνά. Όταν η αράχνη επισκεφθεί μια ιστοσελίδα, καταγράφει όλες τις διευθύνσεις των υπέρ-συνδέσεων και τις προσθέτει στην λίστα για μελλοντική προσπέλαση. Με αυτό τον τρόπο μειώνεται δραματικά ο αριθμός των χαμηλής ποιότητας ιστοσελίδων που θα επισκεφθεί η αράχνη, καθώς στην πλειοψηφία τους, οι σχεδιαστές ιστοσελίδων συνήθως εισάγουν

υπέρ-συνδέσεις μόνο σημαντικών ιστοσελίδων (σημασιολογικά αλλά και ποιοτικά).

Παρά το ότι η διαδικασία αναζήτησης πληροφορίας στο διαδίκτυο από τις αράχνες είναι σχετικά απλή, απαιτείται προσεκτικός προγραμματισμός του λογισμικού αυτού, προκειμένου να αντιμετωπιστούν σωστά όλα τα πιθανά ενδεχόμενα. Αρχικά η αράχνη, ακολουθεί μόνο τις υπέρ-συνδέσεις ιστοσελίδων που δεν έχει επισκεφτεί στο παρελθόν. Σε περίπτωση που έχει επισκεφτεί μια ιστοσελίδα ξανά, γίνεται έλεγχος για το αν έχει περάσει αρκετό χρονικό διάστημα από την τελευταία επίσκεψη της και αν κρίνεται αναγκαίος, ο επανέλεγχος του περιεχομένου της ιστοσελίδας [3]. Βέβαια ένα σημαντικό πρόβλημα που καλούνται να ξεπεράσουν οι αράχνες, είναι η δυσκολία που αντιμετωπίζουν, στον διαχωρισμό του κατά πόσο έχουν ξανά επισκεφθεί ένα δικτυακό τόπο η όχι, ειδικά όταν εμφανίζει μια υπέρ-σύνδεση με διαφορετική ονομασία, «δείχνοντας» στον εαυτό του. Έτσι, ένα πιθανό παράδειγμα θα ήταν ο δικτυακός τόπος http://www.a.com, να εμπεριέχει υπέρ-σύνδεσμο στην ιστοσελίδα http://www.a.com/home.htm. Οι περισσότεροι χρήστες του διαδικτύου, θα είχαν αντιληφθεί άμεσα ότι πρόκειται για την ίδια ιστοσελίδα, σε αντίθεση με τα λογισμικά αράχνης, όπου είναι εξαιρετικά δύσκολος ο εντοπισμός και η διόρθωση του φαινομένου, ειδικά στις πιο σύνθετες καταστάσεις του. Θεωρείται, ότι περίπου το 30% του σύνολου των δικτυακών τόπων, παρουσιάζονται διπλές φορές [5] [6].

Τέλος, πολλές μηχανές αναζήτησης για να μειώσουν το συνολικό κόστος, περιορίζουν τον μέγιστο αριθμό καταγραφής ιστοσελίδων, από κάθε δικτυακό τόπο. Έτσι είναι λανθασμένο να θεωρούμε, ότι αν μια μηχανή αναζήτησης έχει καταγράψει κάποιες από τις υπέρ-συνδέσεις ενός δικτυακού τόπου, έχει καταγράψει και το σύνολο των ιστοσελίδων της.

Οι μεγάλες μηχανές αναζήτησης, δεν αναζητούν πληροφορία στο διαδίκτυο με την χρήση μόνο μιας αράχνης, αλλά διανέμουν τις ιστοσελίδες που χρειάζεται να επισκεφθούν, σε πολλές αράχνες, για την όσο δυνατόν γρηγορότερη αναζήτηση πληροφορίας στο διαδίκτυο. Η κάθε αράχνη μπορεί να έχει ταυτόχρονα «ανοιχτές», εκατοντάδες συνδέσεις «τρέχοντας» από

έναν μόνο υπολογιστή. Είναι σημαντικό να διευκρινίσουμε ότι τα λογισμικά αράχνης, παρά το ότι το όνομα τους υπονοεί ότι ταξιδεύουν στον παγκόσμιο ιστό, δεν λειτουργούν όπως οι ιοί, που πηγαίνουν από υπολογιστή σε υπολογιστή. Η λειτουργία τους, είναι περίπου όμοια με ένα περιηγητή, που στέλνει αίτηση (HTTP request) για μια ιστοσελίδα σε έναν διακομιστή, κατεβάζει την ιστοσελίδα και την διαβιβάζει στον μηχανισμό του ευρετηρίου. Η μόνη διαφορά τους, είναι ότι στην αράχνη, η όλη διαδικασία γίνεται με αυτοματοποιημένο τρόπο [4].

3.4.2 Μηχανισμός ευρετηρίου (indexer)

Καθώς η αράχνη πηγαίνει από ιστοσελίδα σε ιστοσελίδα στον χώρο του διαδικτύου, αυτό που κάνει πρωταρχικά προτού εκτελέσει οποιαδήποτε άλλη διεργασία, είναι η παράδοση της ιστοσελίδας στον μηχανισμό ευρετηρίου, όπου αποθηκεύεται το πλήρες κείμενο της ιστοσελίδας στη βάση δεδομένων της μηχανής αναζήτησης, συνήθως με την μορφή ανεστραμμένου ευρετηρίου. Το ανεστραμμένο ευρετήριο είναι αλφαβητικά ταξινομημένο. Με την κάθε του καταχώρηση, να περιλαμβάνει μια λέξη, μια λίστα με ιστοσελίδες στις οποίες περιλαμβάνεται η λέξη και σε ορισμένες περιπτώσεις και την ακριβή θέση στην οποία βρίσκεται η λέξη, στο κάθε έγγραφο. Μια τέτοια δομή είναι ιδανική για έρευνες με λέξεις κλειδιά, παρέχοντας όσο το δυνατόν γρηγορότερη πρόσβαση στις ιστοσελίδες που περιέχουν τις λέξεις αυτές.

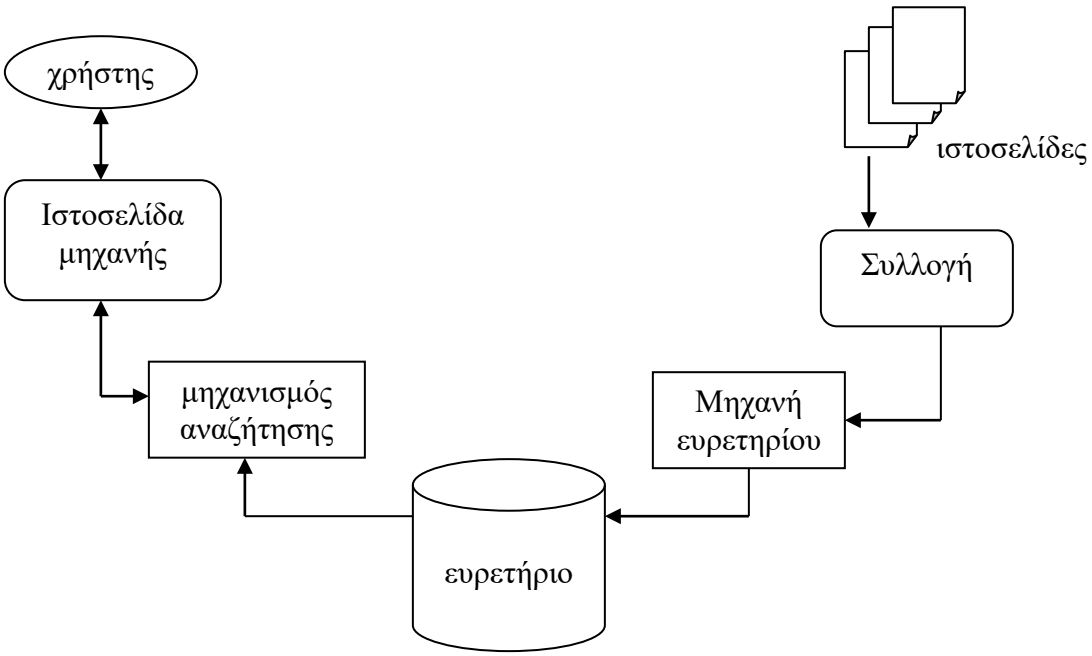
Με σκοπό την βέλτιστη και ταχύτερη αναζήτηση, ορισμένες μηχανές αναζήτησης εξαλείφουν από το ευρετήριο, συνηθισμένες λέξεις, σημεία στίξης, πολλαπλά διαστήματα και ορισμένες φορές μετατρέπουν όλα τα γράμματα σε πεζά. Βέβαια η καταχώρηση ολόκληρου του κειμένου έχει άλλα πλεονεκτήματα, όπως είναι η δυνατότητα χρήσης τελεστών εγγύτητας (NEAR) για τον περιορισμό του αριθμού αποτελεσμάτων των αναζητήσεων, καθώς επίσης και η δυνατότητα αναζήτησης φράσεων ή ακόμα και μεγαλυτέρων κομματιών κειμένου. Τέλος, αν η μηχανή αναζήτησης αποθηκεύσει πλήρως το κείμενο μαζί με τον κώδικα HTML, η αναζήτηση μπορεί να περιοριστεί και σε κάποια χαρακτηριστικά της ιστοσελίδας όπως είναι ο τίτλος, η διεύθυνση της (URL) κλπ.

Σε τακτά χρονικά διαστήματα (συνήθως σε μηνιαία βάση) που ορίζουν οι διαχειριστές των μηχανών αναζήτησης, οι αράχνες επιστρέφουν ξανά τις διάφορες ιστοσελίδες και σε περίπτωση που έχουν ανακαλύψει αλλαγές σε κάποιες από αυτές, τότε ενημερώνουν και τα αντίγραφα του ευρετήριου χρησιμοποιώντας μια από τις ενέργειες της προσθήκης, διαγραφής ή ανανέωσης της πληροφορίας. Οι περισσότερες αξιόλογες μηχανές αποθηκεύουν το πλήρες κείμενο των ιστοσελίδων, υπάρχουν όμως και μερικές που ευρετηριάζουν μόνο τον τίτλο και τις πρώτες γραμμές κειμένου μιας ιστοσελίδας [6] [3].

3.4.3 Μηχανισμός αναζήτησης (searcher)

Το πιο πολύπλοκο και δύσκολο τμήμα στην υλοποίηση μιας μηχανής αναζήτησης, είναι χωρίς αμφιβολία ο μηχανισμός αναζήτησης. Περιλαμβάνει πολλά τμήματα όπως είναι η διασύνδεση με τον χρήστη, μέσω της φόρμας αναζήτησης, τον μηχανισμό που αξιολογεί τα ερωτήματα και εντοπίζει τις πιο σχετικές ιστοσελίδες στην βάση δεδομένων και τέλος το μορφοποιητή των αποτελεσμάτων. Ο εντοπισμός και η ιεράρχηση των σημαντικών και των πιο σχετικών ιστοσελίδων σύμφωνα με το ερώτημα ενός χρήστη, γίνεται με ειδικούς αλγόριθμους που δεν στηρίζονται συνήθως απλά στο περιεχόμενο των ιστοσελίδων, αλλά και σε πολλά άλλα κριτήρια. Κριτήρια όπως είναι η δημοτικότητα μιας ιστοσελίδας, δηλαδή πόσες επισκέψεις δέχεται μια ιστοσελίδα για μια συγκεκριμένη αναζήτηση, ή βάση του ποσοστού που συγκεντρώνει από άλλες ιστοσελίδες που «δείχνουν» με τις υπέρ-συνδέσεις τους σε αυτήν, προσδιορίζοντας με αυτό τον τρόπο τη σημαντικότητα της. Σε γενικές όμως γραμμές, οι δυο κυριότεροι κανόνες (ιεράρχησης) που ακολουθούνται, αφορούν στην τοποθεσία (URL) και στην συχνότητα των λέξεων-κλειδιών μέσα σε μια ιστοσελίδα [7] [3].

Τέλος, τόσο η φόρμα αναζήτησης, όσο και η μορφοποίηση των αποτελεσμάτων, είναι όσον αφορά την δομή τους, ίδιες σε όλες τις μηχανές αναζήτησης. Όλες τους, προσφέρουν φόρμες απλής και προχωρημένης αναζήτησης και δίνουν στους χρήστες τη δυνατότητα να περιορίσουν τη αναζήτηση με διάφορες παραμέτρους.



Εικόνα 3: Λειτουργία μιας μηχανής αναζήτησης [3]

3.5 Γενικά για τις μηχανές αναζήτησης

Κάθε μηχανή αναζήτησης διαθέτει το δικό της δικτυακό τόπο στο διαδίκτυο. Τα έσοδα που προέρχονται από τις διαφημίσεις, δίνουν την δυνατότητα για δωρεάν παροχή υπηρεσιών αναζήτησης στους χρήστες.

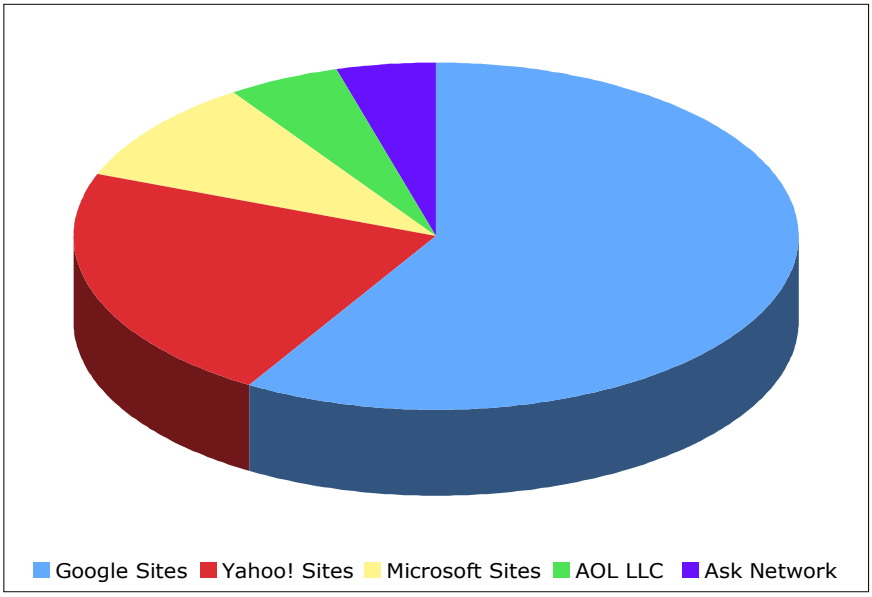
Η αναζήτηση της πληροφορίας από την μεριά του χρήστη, γίνεται πληκτρολογώντας τις λέξεις-κλειδιά για το θέμα που αναζητά, σε ειδικά πεδία της φόρμας (search forms). Η μηχανή αναζήτησης, αφού έχει ακολουθήσει προηγουμένως την διαδικασία που προαναφέραμε, επιστρέφει τα αποτελέσματα που συνήθως περιλαμβάνουν, τίτλους ιστοσελίδων, ένα μικρό απόσπασμα του κειμένου της κάθε ιστοσελίδας ή μια σύντομη περιγραφή, καθώς και μία υπέρ-σύνδεση που οδηγεί σε αυτήν. Οι περισσότερες υπηρεσίες αναζήτησης, προσφέρουν και καταλόγους ιστοσελίδων οργανωμένους κατά θεματική, δίνοντας την δυνατότητα στον χρήστη να πλοηγηθεί αναζητώντας κάτι που τον ενδιαφέρει. Για την προσέλκυση όσο το δυνατόν περισσότερων επισκεπτών, προσφέρουν και μια σειρά από άλλες υπηρεσίες που δεν

έχουν άμεση σχέση με την λειτουργία τους, όπως δωρεάν ηλεκτρονικό ταχυδρομείο, chat, ειδήσεις, χρηματιστήριο, κλπ [3].

3.6 Ποσοστά των μηχανών αναζήτησης

Τα ποσοστά της αγοράς που κατείχαν οι πιο δημοφιλές μηχανές αναζήτησης σύμφωνα με την comscore.com τον Ιανουάριο του 2008 φαίνονται παρακάτω:

Μηχανές αναζήτησης	Ιανουάριος-2008
Google	58.8%
Yahoo	22.2%
Microsoft	9.8%
AOL LLC	4.9%
Ask Network	4.5%
Συνολικό ποσοστό	100%



Εικόνα 4: Ποσοστά στην αγορά των μηχανών αναζήτησης [16]

Το Google κατέχει το μεγαλύτερο μερίδιο επισκεψιμότητας και θεωρείται η μεγαλύτερη μηχανή αναζήτησης στις μέρες μας. Σε καθημερινή βάση δέχεται περισσότερες από 180 εκατομμύρια αιτήσεις, ενώ στην βάση της έχει καταχωρημένα πάνω από 3 δισεκατομμύρια έγγραφα όλων των τύπων. Η μεγάλη επιτυχία του Google, σε σχέση με τις άλλες μηχανές αναζήτησης, οφείλεται στην εμφάνιση υψηλών ποσοστών συνάφειας των αποτελεσμάτων της, σε σχέση με τους όρους της αναζήτησης. Αυτό επιτυγχάνεται χρησιμοποιώντας σε μεγάλο βαθμό για την ιεράρχηση των αποτελεσμάτων, το κριτήριο της ανάλυσης υπέρ-συνδέσεων (PageRank technology). Επίσης διαθέτει ένα περιβάλλον για σύνθετες αναζητήσεις, δίνοντας την δυνατότητα στον χρήστη πέρα από την εξειδικευμένη αναζήτηση συγκεκριμένων τύπων εγγράφων (format) και την δυνατότητα αναζήτησης σε μια συγκεκριμένη διαδικτυακή περιοχή (domain) ή σε μια μόνο γλώσσα.

3.7 Στοχευμένες Μηχανές αναζήτησης

Ο σκοπός των στοχευμένων μηχανών αναζήτησης, είναι η επιλεκτική αναζήτηση πληροφορίας σε συγκεκριμένες ιστοσελίδες του διαδικτύου, που είναι σχετικές με μια ορισμένη θεματική. Αντί να συλλέγουν και να ευρετηριάζουν όλα τα προσβάσιμα έγραφα του διαδικτύου, προκειμένου να μπορούν να ανταποκριθούν σωστά σε όλα τα πιθανά ερωτήματα αναζήτησης, οι στοχευμένες μηχανές αναζήτησης αναλύουν και ορίζουν συνεχώς τα όρια της αναζήτησης τους, έτσι ώστε να ανακαλύψουν τις υπέρ-συνδέσεις εκείνες, που είναι πιο σχετικές με την θεματική που τους ενδιαφέρει. Αποφεύγοντας με αυτό τον τρόπο, περιοχές του διαδικτύου που δεν έχουν να προσφέρουν κάποια πληροφορία σχετική με το θέμα [9].

3.8 Πλεονεκτήματα των στοχευόμενων μηχανών αναζήτησης

Οι στοχευμένες μηχανές αναζήτησης καλύπτουν ένα πολύ μικρό τμήμα του διαδικτύου, με αποτέλεσμα τόσο το κόστος δημιουργίας μιας τέτοιας υπηρεσίας όσο και το κόστος συντήρησης τους να είναι αρκετά πιο χαμηλό, σε σχέση με τις μηχανές

γενικού σκοπού. Ο καθορισμός και ο έλεγχος των δικτυακών τόπων που επισκέπτονται, γίνεται από ειδικό προσωπικό, περιορίζοντας τον αριθμό των ακατάλληλων αποτελεσμάτων που λαμβάνουν οι χρήστες κατά την εκτέλεση μιας αναζήτησης. Έτσι τα αποτελέσματα της αναζήτησης τους, έχουν μεγαλύτερη αξιοπιστία και ενημερώνονται πολύ πιο γρήγορα σε σχέση με τις άλλες, καθώς περιδιαβαίνουν τους δικτυακούς τόπους «ενδιαφέροντος», πολύ πιο συχνά και πιο σχολαστικά από τις συνηθισμένες μηχανές [9] [3].

4. Παρουσίαση στοχευμένου web crawler για αναζήτηση πολιτιστικού περιεχομένου

Στο κεφάλαιο αυτό, περιγράφεται η γενική μεθοδολογία που ακολουθήθηκε για την δημιουργία του λογισμικού αράχνης καθώς και του προγράμματος αυτοματοποιημένης παραγωγής ιστοσελίδων. Επίσης παρουσιάζεται η δομή του γραφικού τους περιβάλλοντος, ο τρόπος λειτουργίας τους, καθώς και ο ψευδοκώδικας στον οποίο βασιστήκαμε για το προγραμματισμό του στοχευμένου λογισμικού αράχνης.

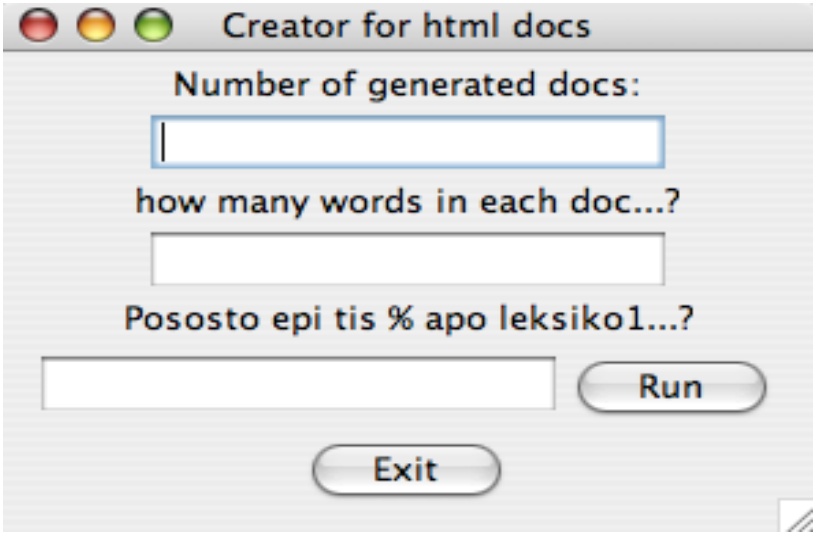
4.1 Μεθοδολογία

Αρχικά χρειάστηκε να προγραμματιστεί το λογισμικό που ήταν απαραίτητο, για την δημιουργία με έναν αυτοματοποιημένο τρόπο, όλων των ιστοσελίδων στις οποίες θα αναζητούσε πληροφορίες, το λογισμικό της αράχνης. Πριν το προγραμματισμό του λογισμικού (Creator.java), δημιουργήθηκαν δυο λεξικά (δυο txt αρχεία), όπου το πρώτο εμπεριέχει οχτώ ενδεικτικές λέξεις πολιτιστικού περιεχομένου και το δεύτερο περιέχει κοινότυπες λέξεις (π.χ. «και», «ίσως», «άμα», «όχι»), λέξεις που όσον αφορά το περιεχόμενο τους, μας είναι αδιάφορες. Το λογισμικό αυτό, προγραμματίστηκε με τέτοιο τρόπο, ώστε ο χρήστης να μπορεί να επιλέξει τον αριθμό των ιστοσελίδων που επιθυμεί να παράγει το πρόγραμμα, το σύνολο των λέξεων που θα περιλαμβάνει το κάθε HTML έγγραφο και τέλος το ποσοστό (επί τις εκατό) των πολιτιστικών λέξεων που θα εμπεριέχουν στο σύνολο τους, οι ιστοσελίδες από το πρώτο λεξικό. Έτσι για τις «ανάγκες» της αναζήτησης, δημιουργήθηκαν αυτόματα εκατό HTML έγραφα, των δέκα λέξεων. Επί του συνόλου των λέξεων που εμπεριέχονται στις ιστοσελίδες, με καθαρά τυχαία (random) τοποθέτηση των λέξεων στο κάθε έγγραφο, το 30% είναι πολιτιστικού περιεχομένου
(από λεξικό 1) και το υπόλοιπο 70%, έχει περιεχόμενο που δεν έχει σχέση με την θεματική που αναζητάει η μηχανή αράχνης (λεξικό 2).

Μετά την δημιουργία του δικτυακού «χώρου» στο οποίο θα αναζητήσει πληροφορία η αράχνη, ακολούθησε η κατασκευή της. Η κύρια λειτουργία της, είναι ο υπολογισμός του ποσοστού εμφάνισης των πολιτιστικών ορών (των λέξεων ενδιαφέροντος) στην κάθε ιστοσελίδα που επισκέπτεται, εντοπίζοντας πόσο σχετική είναι με το θέμα μας. Ο προσδιορισμός των λέξεων ενδιαφέροντος γίνεται από ένα txt αρχείο (SpiderDict.txt) που διαβάζει η μηχανή αναζήτησης και το οποίο περιέχει όλες τις λέξεις που μας ενδιαφέρει να βρούμε στις ιστοσελίδες που ευρετηριάζει. Μετά την διεκπεραίωση της διαδικασίας αυτής και αφού έχει επισκεφθεί ένα συγκεκριμένο αριθμό ιστοσελίδων ακολουθώντας τις υπέρ-συνδέσεις τους, υπολογίζει και παρουσιάζει με φθίνουσα ταξινόμηση, τις ιστοσελίδες με τα μεγαλύτερα ποσοστά εμφάνισης των λέξεων, βάση της θεματικής που μας ενδιαφέρει.

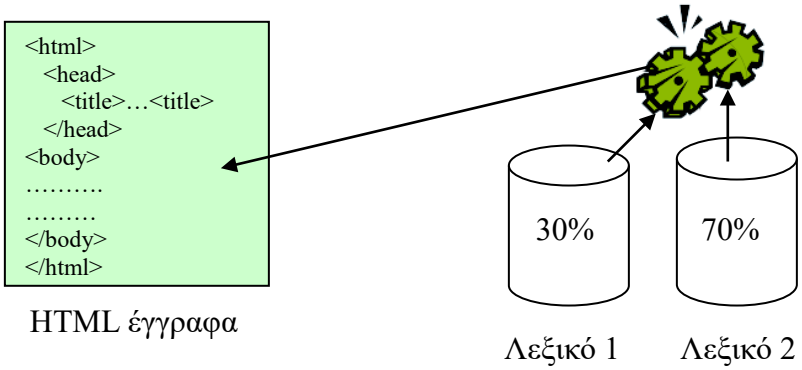
4.2 Πρόγραμμα αυτοματοποιημένης παραγωγής ιστοσελίδων

Για να μπορέσουμε να ελέγξουμε την αποτελεσματικότητα της αράχνης σε ένα συγκεκριμένο «χώρο» ιστοσελίδων, όπου θα μας ήταν εκ των προτέρων, γνωστό το ποσοστό τους σε πολιτιστικό περιεχόμενο, δημιουργήσαμε τον **Creator**, ένα πρόγραμμα αυτόματης παράγωγης ιστοσελίδων. Το περιβάλλον της εφαρμογής, είναι όπως φαίνεται στην εικόνα.



Εικόνα 5: Το γραφικό περιβάλλον της εφαρμογής Creator

Όπως ήδη προαναφέραμε, μέσω του Creator φτιάχτηκαν 100 ιστοσελίδες των 10 λέξεων, με 30% ποσοστό χρήσης από το πρώτο λεξικό (πολιτιστικού περιεχομένου) και 70% από το δεύτερο. Η λειτουργία του Creator μπορεί να φανεί και στο παρακάτω διάγραμμα.



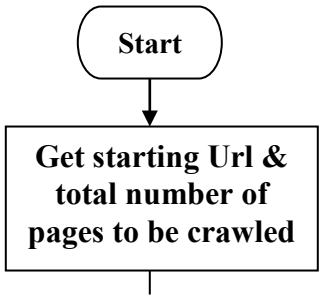
Εικόνα 6: Τρόπος λειτουργίας του λογισμικού Creator

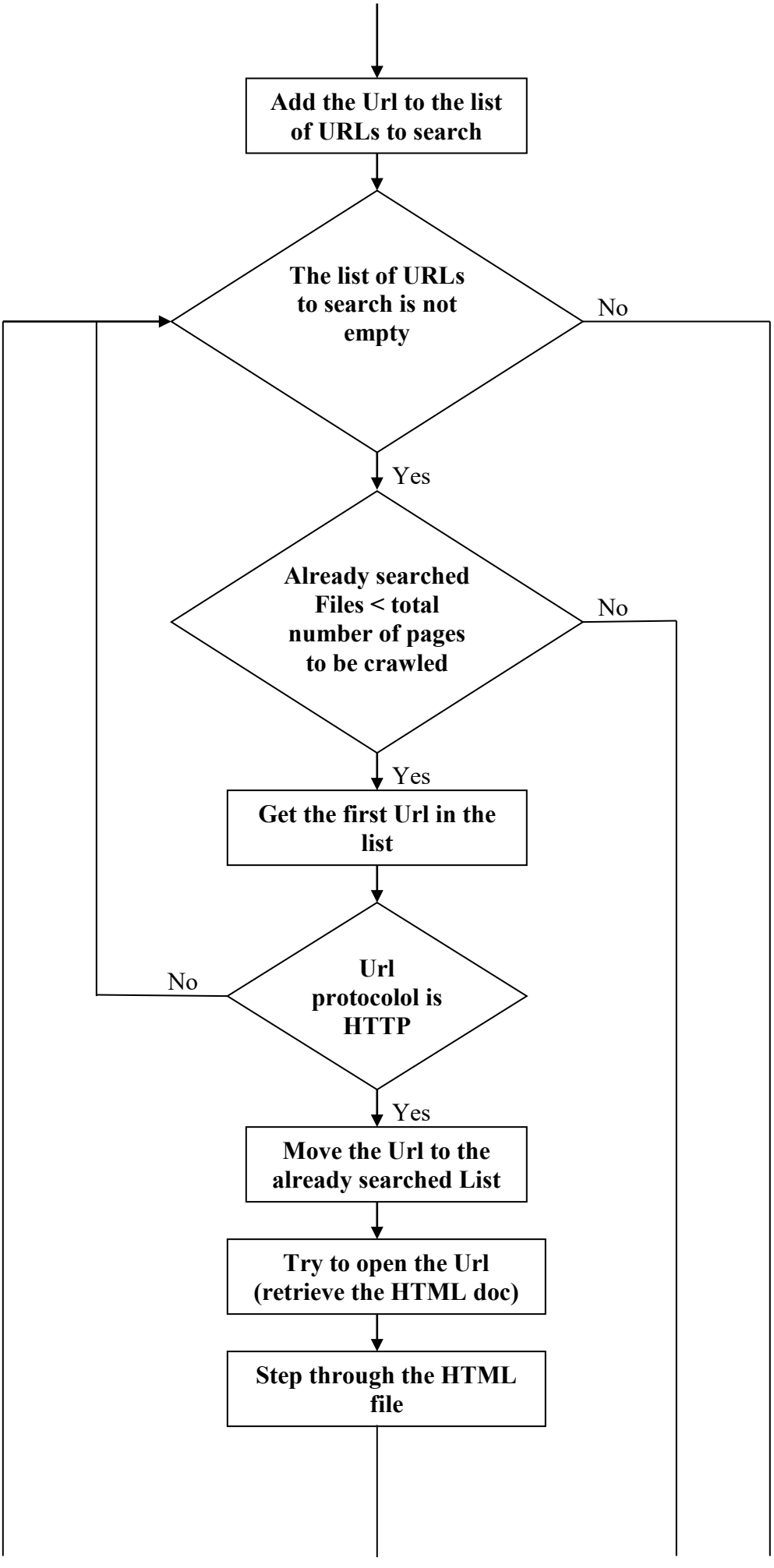
Αρχικά ο Creator διαβάζει τα δυο λεξικά και δημιουργεί 2 vectors που περιέχουν τις λέξεις των λεξικών. Στην συνέχεια, για κάθε έγγραφο HTML που κατασκευάζει, χρησιμοποιεί την συνάρτηση Random() για την τυχαία επιλογή και τοποθέτηση των λέξεων στα έγγραφα από τα δυο vectors, πάντα βέβαια, βάση του ποσοστού που έχει ορίσει ο χρήστης. Όλες οι ιστοσελίδες περιέχουν υπέρ-συνδέσεις με την επόμενη τους. Επίσης ανά δυο ιστοσελίδες (ζυγού αριθμού), προστίθενται με τυχαίο τρόπο επιπλέον υπέρ-συνδέσμοι. Τέλος, έχει προγραμματισθεί έτσι ώστε ανά πέντε ιστοσελίδες, να υπάρχει 50% πιθανότητα για να προστεθεί ακόμη ένας επιπλέον, εντελώς τυχαία ορισμένος, υπέρ-σύνδεσμος.

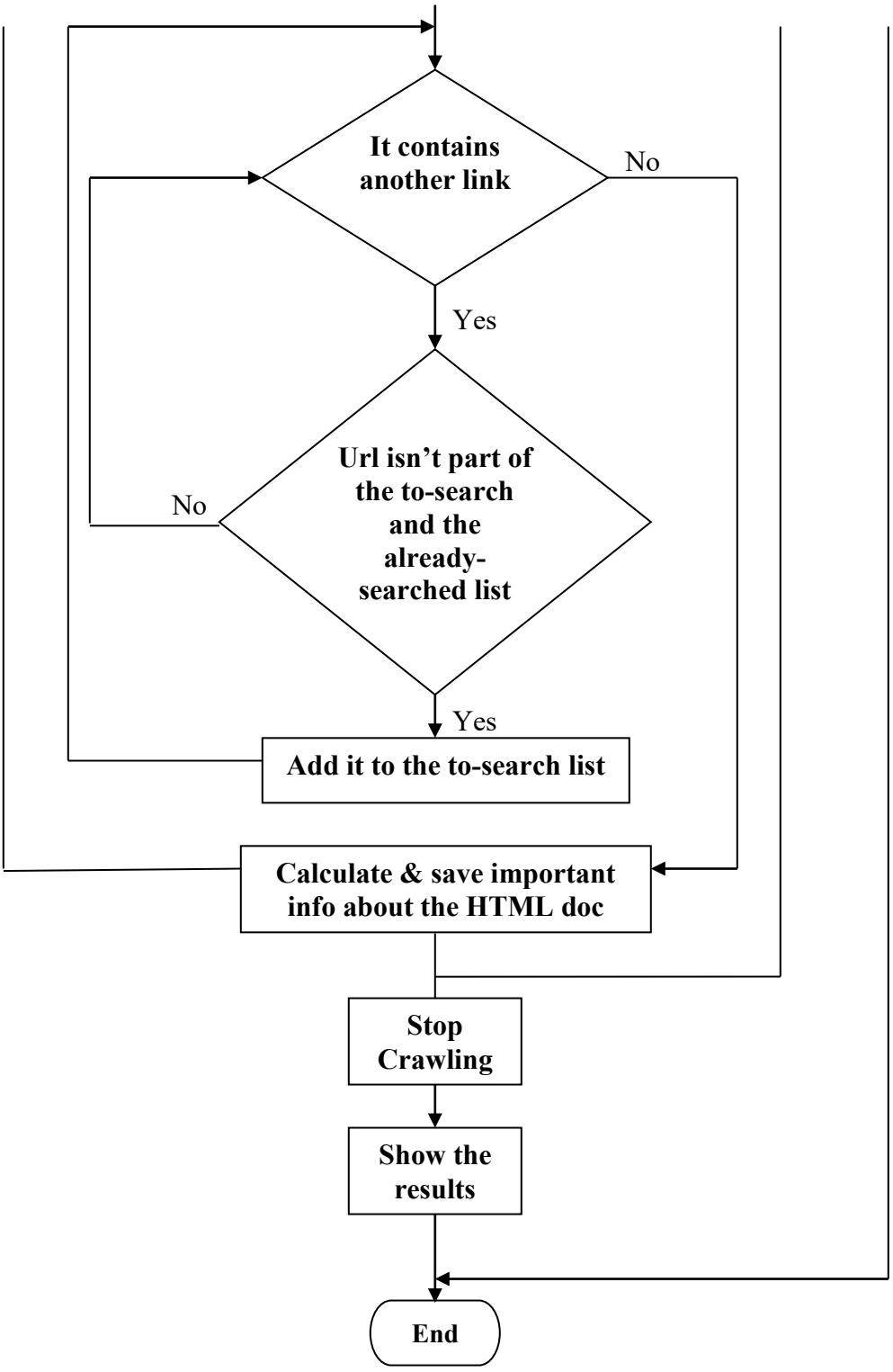
4.3 Ψευδοκώδικας της Μηχανής αράχνης (Spider.java)

```
1.Get the user's input: the starting URL
and the number of desired HTML files to be crawled.
2.Add the URL to the currently empty list of URLs
to search.
3.While the list of URLs to search is not empty
{
4.If the total number of the already searched HTML
files, is smaller than the number of pages that the
user wants to crawl,
{
5.Get the first URL in the list.
6.Check the URL to make sure its protocol is HTTP
(if not, break out of the loop)
7.Move the URL to the list of URLs already
searched.
8.Try to "open" the URL
(that is, retrieve that document From the Web).
9.Step through the HTML file. While the HTML text
contains another link,{
10.If the URL isn't present in either the to-search
list or the already-searched list,
11.Add it to the to-search list.
}
12.Calculate and save all the important information
about the HTML document in an array
}
13.Else
14.Stop the crawling procedure
15.Show the results of the searching
}
```

Εικόνα 7: Ψευδοκώδικας της αράχνης




Add the Url to the list of URLs to search
The list of URLs to search is not empty
No
Yes
Already searched Files < total number of pages to be crawled
No
Yes
Get the first Url in the list
Url protocolol is HTTP
No
Yes
Move the Url to the already searched List
Try to open the Url (retrieve the HTML doc)
Step through the HTML file



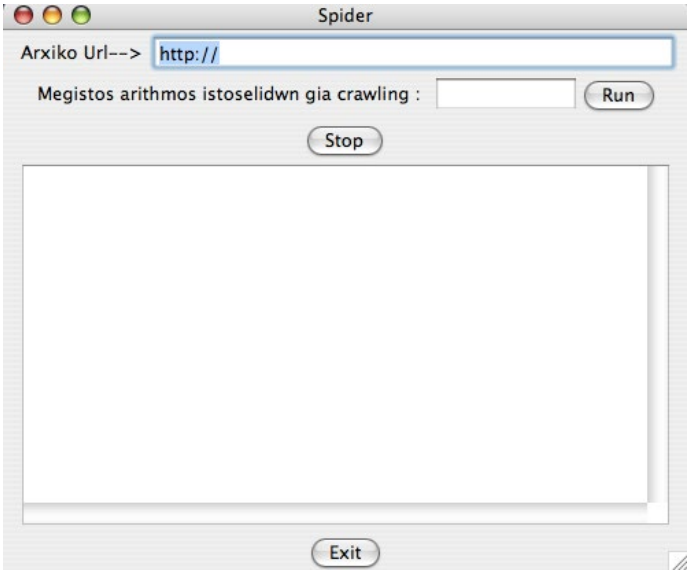
Εικόνα 8: Διάγραμμα ροής της αράχνης

Η αράχνη δουλεύει ως εξής, αρχικά παίρνει από την φόρμα της εφαρμογής το URL που τοποθετεί ο χρήστης και το μέγιστο συνολικό αριθμό ιστοσελίδων που είναι προς αναζήτηση. Στην συνέχεια, προσθέτει το URL στην λίστα των ιστοσελίδων προς

μελλοντική αναζήτηση. Όσο αυτή η λίστα δεν είναι κενή και το άθροισμα των ψαγμένων ιστοσελίδων είναι μικρότερο, από τον μέγιστο αριθμό ιστοσελίδων που μπορούν να αναζητηθούν, παίρνει από την λίστα (προς αναζήτηση), το πρώτο URL. Ελέγχει αν το πρωτόκολλο του είναι HTTP και σε περίπτωση που δεν είναι, βγαίνει έξω από τον βρόγχο. Στην περίπτωση που έχει το σωστό πρωτόκολλο, μεταφέρεται στην λίστα με τις ιστοσελίδες που έχουν ήδη αναζητηθεί. Στην συνέχεια, ανοίγει μια σύνδεση με το συγκεκριμένο URL, προσπαθώντας να κατεβάσει το έγγραφο από το διαδίκτυο. Αφού το κατεβάσει, ελέγχει τον κώδικα του εγγράφου για το αν περιέχει άλλες υπέρ-συνδέσεις. Σε περίπτωση που εμπεριέχει υπέρ-συνδέσμους και το URL που έχει βρει, δεν υπάρχει στην λίστα των ιστοσελίδων προς αναζήτηση, αλλά ούτε και στην λίστα των ήδη ψαγμένων, προστίθεται στην λίστα για μελλοντική αναζήτηση.

Τέλος, για κάθε HTML έγγραφο που επισκέπτεται, μαζεύει και αποθηκεύει σε ένα πίνακα όλες τις απαραίτητες πληροφορίες σχετικά με το περιεχόμενο τους, όπως είναι το πλήθος λέξεων στο έγγραφο, το ποσοστό σχετικότητας με την θεματική μας και ο εντοπισμός των κοινών πολιτιστικών όρων με το λεξικό SpiderDict.txt. Το τέλος της αναζήτησης γίνεται, όταν ο συνολικός αριθμός των ψαγμένων ιστοσελίδων ξεπεράσει, τον μέγιστο αριθμό ιστοσελίδων προς αναζήτηση, εμφανίζοντας τα τελικά αποτελέσματα.

4.4 Γενικά για την εφαρμογή και την λειτουργιά της αράχνης.



Εικόνα 9: Γραφικό περιβάλλον του λογισμικού αράχνης

Ο χρήστης βάζει στο πρώτο πεδίο, την αρχική σελίδα από την οποία θα ξεκινήσει η αράχνη να αναζήτα πληροφορίες (seed), για την θεματική που μας ενδιαφέρει. Είναι καλό το αρχικό URL που θα βάλει, να «δείχνει» σε έναν δικτυακό τόπο, που παρουσιάζει αξιοπιστία και περιεκτικότητα, σε πληροφορία πολιτιστικού περιεχομένου. Καθώς τα αποτελέσματα που μας δίνει είναι τελείως διαφορετικά, ανάλογα με το σημείο εκκίνησης του λογισμικού αράχνης. Στην συνέχεια το μόνο που μένει, είναι να ορίσει, το μέγιστο συνολικό αριθμό ιστοσελίδων που επιθυμεί να ευρετηριάσει η αράχνη και να πατήσει το κουμπί Run.

Η αράχνη αφού κάνει τους κατάλληλους ελέγχους (όπως π.χ. το πρωτόκολλο να είναι HTTP), ανοίγει μια σύνδεση με το συγκεκριμένο URL και κατεβάζει τον κώδικα της ιστοσελίδας, αποθηκεύοντας το σε μια μεταβλητή String. Στην συνέχεια, ξεκινάει την αναζήτηση υπέρ-συνδέσεων και άλλων πληροφοριών που επιθυμεί να συλλέξει από τον κώδικα της, με την χρήση της μεθόδου indexOf(). Κάθε URL που βρίσκει, το προσθέτει στην λίστα για μελλοντική προσπέλαση, αφού ελέγξει πρώτα ότι δεν είναι ήδη και ότι δεν το έχει ξανά επισκεφθεί, στο παρελθόν.

Αφού τελειώσει με την καταγραφή όλων το ιστοσελίδων που δείχνει ο δικτυακός τόπος, συνεχίζει με την καταγραφή συγκεκριμένων δεδομένων, που βρίσκονται στο

«body» του HTML εγγράφου. Δεδομένα, που αφορούν την συχνότητα εμφάνισης των λέξεων του (SpiderDict.txt) λεξικού που έχουμε φτιάξει, το άθροισμα των λέξεων του κάθε δικτυακού τόπου, ή το ποσοστό που δείχνει πόσο σχετική είναι η κάθε ιστοσελίδα, με το περιεχόμενο που αναζητούμε. Αυτή η διαδικασία, ακολουθείται με τον ίδιο ακριβώς τρόπο, σε κάθε ιστοσελίδα που το λογισμικό αράχνης αναζητά. Η διαδικασία της αναζήτησης, θα τερματίσει σε περίπτωση που ο χρήστης πατήσει το κουμπί Stop, είτε σε περίπτωση που συμπληρωθεί ο μέγιστος αριθμός ιστοσελίδων προς αναζήτηση, ή όταν τελειώσει η λίστα με τις υπέρ-συνδέσεις προς «επίσκεψη».

Καθώς αναζητά πληροφορίες από τον έναν δικτυακό τόπο στον άλλον, διαμορφώνεται σταδιακά ο πίνακας Spider, που έχει την δομή που φαίνεται παρακάτω.

Πίνακας Spider

λέξεις από το SpiderDict.txt, που εμφανίζονται στις ιστοσελίδες

Λέξη 1	0	0	0	....	....
Λέξη 2	1	0	1	....	....
Λέξη 3	0	0	1	....	....
Λέξη 4	0	0	0	....	....
Λέξη 5	0	1	1	....	....
Λέξη 6	0	0	1	....	....
Λέξη 7	1	0	1	....	....
Λέξη 8	0	0	0	....	....
Σύνολο λέξεων ενδιαφέροντος	2	1	5	....	....
Συνολικός αριθμός λέξεων SpiderDict.txt	8	8	8	....	....
Ποσοστό (%) σχετικότητας με την θεματική μας	25	12.5	62.5	....	....
#HTML doc	1	2	3	....	....

Εικόνα 10: Πίνακας καταγραφής γενικών πληροφοριών(Spider)

Όσες από τις λέξεις του λεξικού μας εμφανίζουν μονάδα, σημαίνει ότι έχουν εντοπιστεί από την μηχανή αναζήτησης, στο κείμενο του HTML εγγράφου.

Η αράχνη, για κάθε ιστοσελίδα που επισκέπτεται, αθροίζει τις λέξεις ενδιαφέροντος που έχει βρει και τις καταχωρεί στην ένατη γραμμή του πίνακα. Τέλος, υπολογίζει το ποσοστό σχετικότητας της κάθε ιστοσελίδας, σε σχέση με την θεματική μας, πολλαπλασιάζοντας τις λέξεις ενδιαφέροντος που έχει βρει στο έγγραφο, με το 100 και διαιρώντας το στην συνέχεια, με το συνολικό αριθμό των λέξεων από το SpiderDict.txt.

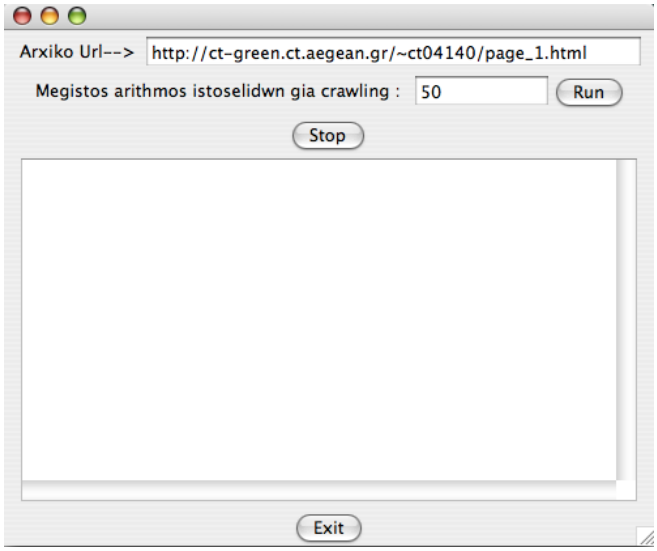
Με το τέλος της αναζήτησης, για την καλύτερη εμφάνιση των αποτελεσμάτων, δημιουργείται ένας πίνακας που ονομάζεται apotelesmata (string) και περιέχει στην κάθε του γραμμή, ένα vector με δυο στοιχεία. Το πρώτο στοιχείο αναφέρεται στο ποσοστό σχετικότητας που εμφανίζει μια ιστοσελίδα με το θέμα μας και το δεύτερο, στο URL της. Αφού κατασκευάσει τον πίνακα, τον ταξινομεί (sorting), με φθίνουσα σειρά από το μεγαλύτερο στο μικρότερο ποσοστό και τον εκτυπώνει.

[80.0, http://ct-green.ct.aegean.gr/~ct04140/page11.html]
[55.0, http://ct-green.ct.aegean.gr/~ct04140/page45.html]
.....
.....
[12.5, http://ct-green.ct.aegean.gr/~ct04140/page21.html]
[0.0, http://ct-green.ct.aegean.gr/~ct04140/page76.html]

Εικόνα 11: Πίνακας-αποτελεσμάτων της αράχνης

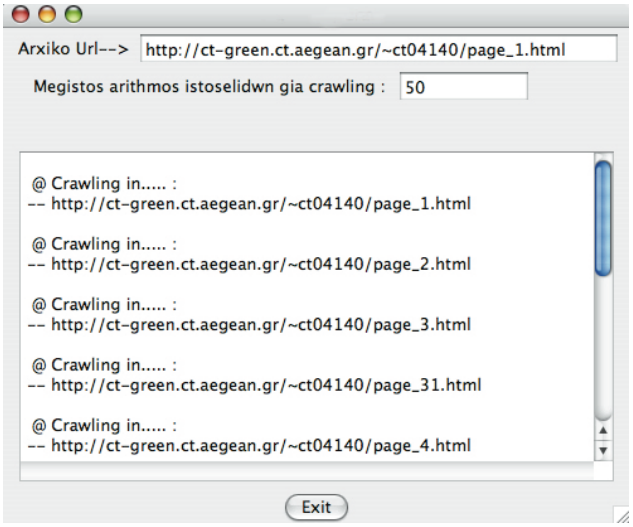
4.5 Αναζήτηση πολιτιστικού περιεχομένου με την χρήση της αράχνης

Όπως προαναφέραμε, έχουμε φτιάξει προηγουμένως με έναν αυτοματοποιημένο τρόπο και έχουμε ανεβάσει στον Server (ct-green.ct.aegean.gr), τις ιστοσελίδες εκείνες στις οποίες θα αναζητήσει πολιτιστική πληροφορία, η αράχνη.



Εικόνα 12: Τοποθέτηση αρχικού URL και μέγιστου αριθμού ιστοσελίδων προς αναζήτηση

Επιλέγουμε ως αρχικό URL το http://ct-green.ct.aegean.gr/~ct04140/page_1.html και βάζουμε ένα όριο 50 ιστοσελίδων, τις οποίες θα ευρετηριάσει η μηχανή αναζήτησης (βλέπε παραπάνω εικόνα). Με το που πατήσουμε το Run, ξεκινάει να ψάχνει στις διάφορες ιστοσελίδες, ακολουθώντας τις υπέρ-συνδέσεις τους, ενημερώνοντας μας ταυτόχρονα σε ποια ιστοσελίδα, αναζητά πολιτιστική πληροφορία, εκείνη την στιγμή (εμφάνιση του URL).

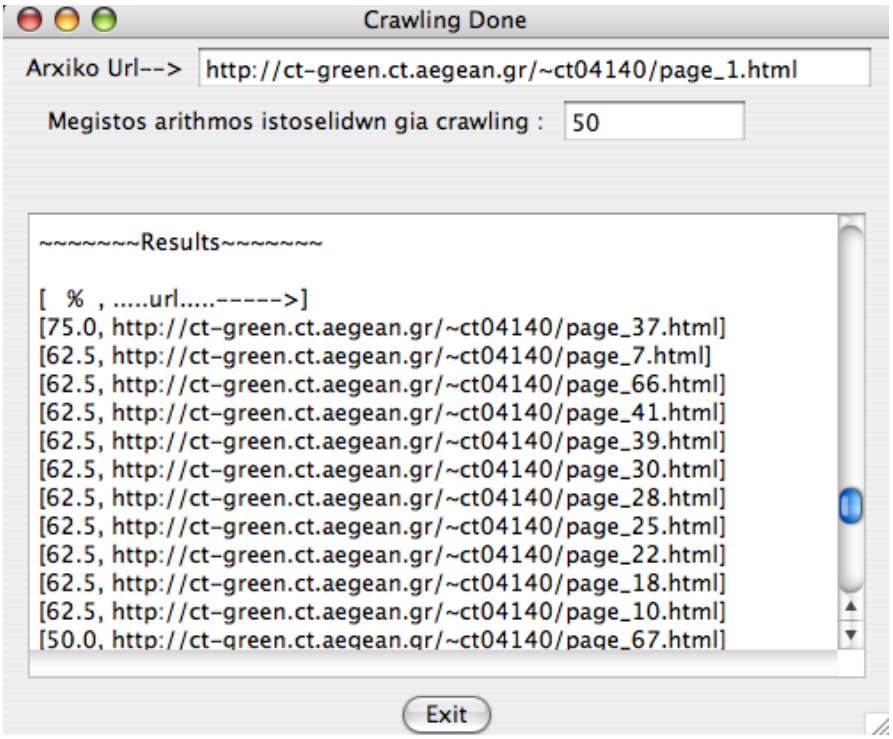


Εικόνα 13: Παρουσίαση όλων των ιστοσελίδων που αναζητούνται

Καθώς τρέχει η εφαρμογή, εμφανίζονται ταυτόχρονα στο Terminal διάφορες πληροφορίες που αφορούν την αναζήτηση. Όπως το URL της ιστοσελίδας στην οποία έχει ανοίξει σύνδεση, το ποσοστό σχετικότητας της ιστοσελίδας με την θεματική μας, αλλά και την στήλη του πίνακα Spider, που αφορά την συγκεκριμένη ιστοσελίδα.

Εικόνα 14: Αποτελέσματα τρέχουσας αναζήτησης στο Terminal

Στο τέλος, με την ολοκλήρωση της αναζήτησης πολιτιστικού περιεχομένου από την αράχνη, εμφανίζονται τα τελικά αποτελέσματα. Οι ιστοσελίδες που παρουσιάζονται είναι ιεραρχημένες (φθίνουσα σειρά), με βάση το ποσοστό σχετικότητας που παρουσιάζουν με τους πολιτιστικούς όρους από το αρχείο κειμένου (SpiderDict.txt).



Εικόνα 15: Παρουσίαση των τελικών αποτελεσμάτων

5. Συμπεράσματα & Κατευθύνσεις για μελλοντική έρευνα

Στόχος της εργασίας ήταν η δημιουργία ενός στοχευμένου λογισμικού αράχνης, για αναζήτηση πολιτιστικού περιεχομένου. Ο προγραμματισμός έγινε στη γλώσσα Java. Όσο περισσότερο προχωρούσε στην ανάπτυξη της η εφαρμογή, τόσο περισσότερο συνειδητοποιούσες το πλήθος των δυνατοτήτων που έχει η γλώσσα Java και πόσο τεράστια είναι η βάση των βιβλιοθηκών της, για διάφορες διεργασίες. Η τεράστια αυτή βάση «γλιτώνει» τους προγραμματιστές από στην συγγραφή μεγάλου αριθμού γραμμών κώδικα (εντοπίσαμε ένα μεγάλο αριθμό έτοιμων βιβλιοθηκών για parsing των λέξεων, από τα HTML έγραφα).

Στις μέρες μας με τον τεράστιο όγκο πληροφορίας που υπάρχει στο διαδίκτυο, η αναζήτηση και η ανάκτηση πληροφορίας αποτελούν την μεγαλύτερη πρόκληση για τους χρήστες του Διαδικτύου και ειδικά για αυτούς που δεν γνωρίζουν συγκεκριμένες διευθύνσεις των ιστοσελίδων στις οποίες χρειάζεται να απευθυνθούν.

Η σημερινή μορφή του Διαδικτύου και ο ρυθμός με τον οποίο αναπτύσσεται, δεν επιτρέπει την απλή και εύκολη πλοήγηση όπως αυτή γινόταν όταν ήταν περιορισμένος ο αριθμός ιστοσελίδων. Αυτό μπορεί να το συνειδητοποιήσει κανείς και από τα αποτελέσματα που παρουσιάζει η εφαρμογή μας. Σε ένα σχετικά πάρα πολύ μικρό πλήθος ιστοσελίδων (100 ιστοσελίδες), ακόμη και αν γνώριζες τα URL τους, η αναζήτηση και ο εντοπισμός της πιο σχετικής σε πολιτιστικό περιεχόμενο ιστοσελίδας, είναι πολύ χρονοβόρα και σχετικά επίπονη διαδικασία. Καθώς θα έπρεπε ο χρήστης να ακολουθήσει την ροή των υπέρ-συνδέσμων και πηγαίνοντας από ιστοσελίδα σε ιστοσελίδα, να καταγράφει και να υπολογίζει, το ποσοστό σχετικότητας του περιεχομένου τους, με την θεματική που τον ενδιαφέρει. Σε αντίθεση με τον άνθρωπο που θα χρειαζόταν ώρες για να το πραγματοποιήσει, η διεργασία αυτή πραγματοποιείται από το λογισμικό αράχνης (Spider) σε λίγα μόνο δευτερόλεπτα.

Μέσα από την ολοκλήρωση της εργασίας μαθαίνεις περισσότερα και κατανοείς ακόμη καλύτερα, τον τρόπο λειτουργίας των στοχευμένων μηχανών αναζήτησης. Εμβαθύνεις περισσότερο στον προγραμματισμό και αντιλαμβάνεσαι πόσοι πολλοί παράγοντες πρέπει να λαμβάνονται υπόψη για την σωστή εκτέλεση χωρίς σφάλματα, ενός τέτοιου λογισμικού. Σίγουρα η εμπειρία που αποχτάς μέσα από την δημιουργία

τέτοιων λογισμικών, σε βοηθάει ώστε να μπορείς να εξελίσσεις συνεχώς τον τρόπο λειτουργία τους, με σκοπό να γίνονται ολοένα και πιο λειτουργικά όσο και πρακτικά, ανταποκρινόμενα στις ανάγκες των χρηστών.

Θα μπορούσαμε σίγουρα αν διαθέταμε περισσότερο χρόνο, να βελτιώσουμε ακόμη περισσότερο τον τρόπο λειτουργίας και να προσθέσουμε ακόμη μεγαλύτερες δυνατότητες στην μηχανή αράχνης. Μια από αυτές, είναι η δυνατότητα επιλογής της στρατηγικής που ακολουθείται από το web crawler, για την αναζήτηση της πληροφορίας. Έτσι θα ήταν καλό να μπορεί ο χρήστης να επιλέγει, μεταξύ αναζήτησης κατά βάθος (depth-first search) και κατά πλάτος (breadth-first search). Στην αναζήτηση κατά βάθος, επεκτείνει πάντα στο βαθύτερο κόμβο του τρέχοντος συνόρου, του δέντρου αναζήτησης. Ενώ κατά πλάτος, επεκτείνεται πρώτα ο κόμβος ρίζα, μετά επεκτείνονται όλοι οι διάδοχοι του κόμβου ρίζας, μετά οι δικοί τους διάδοχοι κ.ο.κ [17]. Επίσης κάτι ακόμα που θα μπορούσαμε να συμπεριλάβουμε, είναι η παρουσίαση σε μορφή διαγράμματος όλων των εγγράφων που έχει επισκεφθεί η αράχνη, καθώς και με ποιο τρόπο συνδέονται αυτά μεταξύ τους.

Η αράχνη με το τερματισμό της αναζήτησης, αποθηκεύει σε ένα πίνακα τα ποσοστά και τα URL των ιστοσελίδων που έχει ψάξει. Θα μπορούσαμε επιπλέον, να αποθηκεύουμε και τον συνολικό HTML κώδικα τους, έτσι ώστε να δίνονται επιπλέον δυνατότητες αναζήτησης στον χρήστη. Όπως έχουμε ξανά αναφέρει, η μηχανή αναζήτησης αποτελείται από τρία μέρη, την αράχνη που αναζητά περιεχόμενο, το ευρετήριο που αποθηκεύονται οι ιστοσελίδες με τα δεδομένα που καταγράφει και το μηχανισμό αναζήτησης. Έτσι θα μπορούσαμε εκτός από την δημιουργία ενός πληρέστερου ευρετηρίου, να φτιάξουμε και μια φόρμα με την οποία θα μπορούσε ο χρήστης να αναζητήσει περιεχόμενο με συγκεκριμένες λέξεις κλειδιά. Βέβαια αυτό προϋποθέτει και την αντίστοιχη διαδικασία καταγραφής δεδομένων στο ευρετήριο, δηλαδή την καταγραφή και αποθήκευση στην βάση δεδομένων, της θέσης και του περιεχομένου των πιο συχνά εμφανιζόμενων λέξεων ενδιαφέροντος, καθώς και το URL της σελίδας στην οποία εντοπίστηκαν.

6. Βιβλιογραφία

[1] Virtual School, The sciences of Education Online, http://web.auth.gr/virtualschool/1.2/praxis/TheInternet/1.html

[2] Aron O'Cass, Tino Fenech "Web retailing adoption: exploring the nature of internet users Web retailing behaviour", 2002

[3] Ανδρέας Βέγλης, Ανδρέας Πομπόρτσης, Ευαγγελία Αβραάμ, "Έρευνα & συλλογή πληροφοριών στο διαδίκτυο"

[4] Monica Peshave, Kamyar Dezhgosha: "How search engines work and a web crawler application", University of Illinois

[5] Mike Thelwall: "Methodologies for Crawler Based Web Surveys", School of Computing University of Wolverhampton, 2002

[6] Lan Huang: "A Survey On Web Information Retrieval Technologies"

[7] Arvind Arasu, Junghoo Cho, Hector Garcia-Molina, Andreas Paepcke and Spiram Raghavan: "Searching the Web"

[8] Knut Magne Risvik, Rolf Michelsen: "Search engines and Web dynamics", 2002

[9] Soumen Chakrabarti, Martin van den Berg, Byron Dom: "Focused crawling: a new approach to topic-specific Web resource discovery", 1999

[10] Rogers Cadenhead, Laura Lemay, "Πλήρες Εγχειρίδιο Java 2"

[11] Laura Lemay, "Πλήρες Εγχειρίδιο HTML 4"

[12] James F. Kurose, Keith W. Ross, "Δικτύωση Υπολογιστών"

[13] Andrew S. Tanenbaum, "Δίκτυα Υπολογιστών"

[14] Αλέξανδρος Καράκος "Διαδίκτυο Παγκόσμιος Ιστός & Τεχνικές προγραμματισμού" κεφάλαιο 1, «Εισαγωγή στο Διαδίκτυο»

[15] Δ. Γαβαλάς, "Τεχνολογίες Ηλεκτρονικού Εμπορίου", κεφάλαιο στο «Κυβερνοχώρος και η Βιομηχανία της Ψηφιακής Επικοινωνίας: Ηλεκτρονικό Εμπόριο και Νέα Μέσα στην Κοινωνία των Πληροφοριών και της Γνώσης», Γ. Γκαντζιάς (επιμέλεια), υπό έκδοση.

[16] Comscore, http://www.comscore.com/

[17] Stuart Russell, Peter Norvig, "Τεχνητή Νοημοσύνη Μια σύγχρονη προσέγγιση"

Παράρτημα: Πηγαίος Κώδικας (source code)

Παραθέτουμε παρακάτω, ένα ενδεικτικό κώδικα από την εφαρμογή του λογισμικού αράχνης. Η αράχνη δουλεύει ως έξης, αφού πάρει το αρχικό URL που δίνει ο χρήστης και ανοίξει μια σύνδεση μαζί του, αναζητάει και ψάχνει στον κώδικα του εγγράφου και για άλλους υπέρ-συνδέσμους, ώστε να τους καταχωρήσει στην λίστα προς μελλοντική αναζήτηση. Η διαδικασία αυτή επαναλαμβάνεται πηγαίνοντας από ιστοσελίδα σε ιστοσελίδα και τερματίζει μόνο όταν είναι κενή η λίστα των ιστοσελίδων προς αναζήτηση, η όταν ξεπεραστεί ο μέγιστος αριθμός ιστοσελίδων που είναι για ψάξιμο.

```
   public void run() {

            setTitle("running..");
            GiaPsaksimo.removeAllElements();
          Psaxthike.removeAllElements();
GiaPsaksimo.addElement(StartUrl);
            Thread thisTread=Thread.currentThread();
         while (runner==thisTread && GiaPsaksimo.size()>0){

         if (metritis_htmls<max_htmls_Xristi){
                String strurl=" ";
                URL  urlnow;
               String line="";
               String html="";
                 StringBuffer buf= new StringBuffer();
                 strurl =(String) GiaPsaksimo.elementAt(0);
                 metritis_htmls++;
                 try {
                           urlnow = new URL(strurl);
                           if (urlnow.getProtocol().compareTo("http") != 0)
                          break;
                 } catch (MalformedURLException e) {
```

```
                setTitle("ERROR: invalid URL " +strurl);
                break;
            }
               GiaPsaksimo.removeElementAt(0);
                Psaxthike.addElement(strurl);
                Url_links.addElement(strurl);

                try {
                        URLConnection conn =urlnow.openConnection();
                        conn.connect();
                text.append("\n"+" @ Crawling in..... : ");
                        text.append("\n"+"-- "+strurl);
                        text.append("\n");
                        System.out.println("Connection opened...  "+strurl );
                        InputStreamReader in=new
InputStreamReader(conn.getInputStream());
                        BufferedReader data=new BufferedReader(in);
                        while ((line=data.readLine())!=null)
                                buf.append(line+"\n");
                                 html=buf.toString();
                }catch (IOException e){
                        System.out.println("IO Error:"+e.getMessage());
                        continue;
                }


                String lowerHtml=html.toLowerCase();

int index=0;
while ((lowerHtml.indexOf("<a",index)!=-1))
{
        if ((lowerHtml.indexOf("href",(lowerHtml.indexOf("<a",index))))==-1)
        break;
```

```
        else
        if
((lowerHtml.indexOf("=",(lowerHtml.indexOf("href",(lowerHtml.indexOf("<a",index))))))==-
1)
        break;
      else
          {
        index=lowerHtml.indexOf("=",(lowerHtml.indexOf("href",(lowerHtml.indexOf("<a",i
ndex)))));

index=(index+2);
int start=index;
int end=0;
if  ((lowerHtml.indexOf("\"",index))==-1)
break;

else {
        end=lowerHtml.indexOf("\"",index);
        String remainlink=lowerHtml.substring(start,end);
        String strLink="";

        try{

        URL urlLink=new URL(urlnow,remainlink);
         if (urlLink.getProtocol().compareTo("http") != 0)
                        break;
       strLink = urlLink.toString();
        }
        catch (MalformedURLException e) {
                setTitle("ERROR: invalid URL " +strurl);
                continue;
           }
```

```
                if ((!GiaPsaksimo.contains(strLink)) && (!Psaxthike.contains(strLink))) {
                        GiaPsaksimo.addElement(strLink);
                }

}
}
}
```