ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εξόρυξη Γνώσης από Πλοηγήσεις Χρηστών σε Πύλες Καταλόγων (Portal Catalogs)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΕΛΕΝΗΣ Γ. ΧΡΙΣΤΟΔΟΥΛΟΥ

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2005

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εξόρυξη Γνώσης από Πλοηγήσεις Χρηστών σε Πύλες Καταλόγων (Portal Catalogs)

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΕΛΕΝΗΣ Γ. ΧΡΙΣΤΟΔΟΥΛΟΥ

Επιβλέπων : Τιμολέων Σελλής
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή τη 19η Οκτωβρίου 2005

.....	.....	.....
Τιμολέων Σελλής	Ιωάννης Βασιλείου	Νικόλαος Μήτρου
Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.	Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2005

.....

ΕΛΕΝΗ Γ. ΧΡΙΣΤΟΔΟΥΛΟΥ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών Ε.Μ.Π.

Περίληψη

Στις μέρες μας, το διαδίκτυο είναι η μεγαλύτερη πηγή πληροφορίας και οι περισσότεροι από εμάς το χρησιμοποιούμε για την εύρεση στοιχείων που μας ενδιαφέρουν. Ιδιαίτερα οι πύλες καταλόγων λόγω της δομημένης οργάνωσής τους με βάση το σχήμα κατηγορία/ υποκατηγορία, προτιμούνται για την εξαγωγή πληροφοριών. Έτσι, η αναζήτηση γίνεται αποτελεσματικότερη. Στόχος της συγκεκριμένης διπλωματικής εργασίας είναι η υλοποίηση ενός συστήματος το οποίο εξετάζει τις πλοηγήσεις των χρηστών μιας πύλης καταλόγου και υποστηρίζει διαδικασίες εξόρυξης γνώσης. Πιο συγκεκριμένα, πραγματοποιεί αναζητήσεις πλοηγήσεων δοθείσης μιας αρχικής πλοήγησης-προτύπου, αναζητήσεις χρηστών με βάση διάφορα χαρακτηριστικά πλοήγησης, ομαδοποιήσεις χρηστών με βάση τον τρόπο πλοήγησής τους και ταυτοποίηση χρηστών. Αυτές οι διαδικασίες εξόρυξης γνώσης είναι ιδιαίτερα χρήσιμες για το διαχειριστή της πύλης και μπορούν να χρησιμοποιηθούν για την παρατήρηση συμπεριφορών χρηστών, για την εξαγωγή προσωπικών προτιμήσεων και για την αναδιοργάνωση της πύλης ώστε να εξυπηρετούνται καλύτερα οι χρήστες στην διάρκεια της αναζήτησης.

Λέξεις Κλειδιά

Πύλη καταλόγου, γράφος πύλης καταλόγου, ιεραρχία, πλοήγηση, εξόρυξη δεδομένων, συσταδοποίηση, δομική απόσταση

Abstract

Internet is nowadays a huge source of information and many people exploit it to search for data interested in. Portal Catalogs is a popular means of organizing information. Their structure is based on hierarchies in the form of categories/subcategories, which makes users' search more effective. This Diploma Thesis aims at the development of data mining techniques on user navigations in the hierarchies of portal catalogs. Specifically, the Thesis studies and implements navigation retrieval capabilities given various input patterns, and user clustering tasks based on the form of users' navigations. The above data mining tasks are quite useful for portal administrators, since they can be used to observe users' behaviour, extract personal preferences and re-organize the structure of the portal to satisfy better user needs and navigational habits.

Keywords

Portal catalog, portal catalog graph, hierarchy, navigation, data mining, clustering, structural distance.

Ευχαριστίες

Η διπλωματική εργασία εκπονήθηκε στο Εργαστήριο Συστημάτων Βάσεων Γνώσεων και Δεδομένων (ΕΣΒΓΔ) του Εθνικού Μετσοβίου Πολυτεχνείου. Αποτέλεσε μια πολύ καλή αφορμή για την ενασχόλησή μου με τον πολύ ενδιαφέροντα τομέα της εξόρυξης γνώσης από δεδομένα (data mining) καθώς και με το χώρο του προγραμματισμού. Στο σημείο αυτό θα ήθελα να ευχαριστήσω τον καθηγητή κ. Τιμολέοντα Σελλή για την ευκαιρία που μου προσέφερε να ασχοληθώ με ένα τέτοιο θέμα αλλά και για την πολύτιμη εποπτεία του κατά τη διάρκεια εκπόνησης της διπλωματικής. Επίσης, θα ήθελα να ευχαριστήσω θερμά το Δρ. Θοδωρή Δαλαμάγκα για την πολύ μεγάλη βοήθεια που μου παρείχε, για τη συνεχή καθοδήγησή του και τις χρήσιμες συμβουλές του, αλλά και για το χρόνο που μου αφιέρωσε όλο αυτό το διάστημα. Τέλος, ευχαριστώ ιδιαίτερα τους γονείς μου, την αδερφή μου και τους φίλους μου για την ψυχολογική κυρίως υποστήριξη που μου παρείχαν στο διάστημα αυτό.

Στους γονείς μου, Γεώργιο και Γεωργία

Πίνακας περιεχομένων

1

Εισαγωγή

Λόγω της ραγδαίας ανάπτυξης της τεχνολογίας των υπολογιστών και των δικτύων, το Διαδίκτυο (Internet) έχει καταστεί η κύρια πηγή πληροφόρησης. Το Διαδίκτυο προτιμάται από τους περισσότερους από εμάς, από τη στιγμή που μπορούμε εύκολα, με το απλό πάτημα ενός κουμπιού, να έχουμε άμεση πρόσβαση σε δεδομένα τα οποία σχετίζονται με την πληροφορία που αναζητούμε. Δοθέντος όμως του τεράστιου όγκου δεδομένων που βρίσκονται αποθηκευμένα στον Παγκόσμιο Ιστό, όλοι μας πολλές φορές νιώθουμε ανικανοποίητοι από τα αποτελέσματα των μηχανών αναζήτησης. Τα περισσότερα από αυτά απέχουν πολύ από αυτό που ψάχνουμε λόγω του τυχαίου τρόπου εμφάνισής τους, με βάση μόνο μια φράση που εμείς έχουμε τοποθετήσει στη μηχανή. Μία αρκετά καλή λύση στο πρόβλημα αυτό είναι η οργάνωση της πληροφορίας σε Πύλες Καταλόγων (Portal Catalogs). Οι Πύλες Καταλόγων είναι κόμβοι πληροφορίας που παρέχουν δυνατότητες πλοήγησης σε δεδομένα που είναι οργανωμένα σε ιεραρχίες με βάση τη δομή κατηγορία /υποκατηγορία [DMS03]. Αυτή η δομή καθιστά πιο εύκολη την ανάκτηση των ζητούμενων δεδομένων για ένα χρήστη ο οποίος ξέρει τι ψάχνει να βρει. Οι περισσότερες πύλες καταλόγων παρέχουν σαφή καθοδήγηση, με την τοποθέτηση των κατάλληλων συνδέσμων (links), στον Παγκόσμιο Ιστό. Υπάρχουν πύλες καταλόγων διαφόρων ειδών, ανάλογα με τις ολοένα τροποποιούμενες απαιτήσεις για αποτελεσματικότερη άντληση πληροφορίας. Υπάρχουν για παράδειγμα πολλές κάθετες (vertical) πύλες καταλόγων, δηλαδή κατάλογοι πάνω σε ένα συγκεκριμένο

θέμα ή τομέα (π.χ ειδικοί για ηλεκτρονικές αγορές για hardware, για φωτογραφικό εξοπλισμό, για πολιτιστικά θέματα και άλλα).

Σε πολλές περιπτώσεις, προκύπτει η ανάγκη για μελέτη των πλοηγήσεων που πραγματοποιούνται σε μια τέτοια ιεραρχική δομή καταλόγου. Τέτοιες περιπτώσεις είναι η ανάγκη για γνώση των συμπεριφορών των χρηστών της πύλης, αλλά και η ανάγκη αναδιοργάνωσης της πύλης με τρόπο που να εξυπηρετεί την αποτελεσματικότερη εύρεση της αναζητούμενης από τους χρήστες πληροφορίας. Από τη μελέτη αυτή μπορούν να εξαχθούν σημαντικά συμπεράσματα για τις προτιμήσεις των χρηστών, τα οποία με τη σειρά τους θα χρησιμοποιηθούν για τη βελτίωση των δυνατοτήτων της πύλης. Για παράδειγμα, μετά την εύρεση των δημοφιλέστερων πλοηγήσεων, μπορούν αυτές να δίνονται σαν σύνδεσμος στο πάνω μέρος της πύλης, έτσι ώστε να είναι άμεσα ορατές από όλους τους χρήστες. Ακόμα, με γνώση των όμοιων πλοηγήσεων μεταξύ χρηστών, μπορούν να ενταχθούν οι χρήστες με πανομοιότυπα ενδιαφέροντα σε κοινές ομάδες και να παρέχεται στις ομάδες αυτές πιο άμεση πρόσβαση στις κατηγορίες μεγαλύτερου ενδιαφέροντος.

1.1 Αντικείμενο της διπλωματικής

Στη διπλωματική αυτή εργασία, στόχος είναι η εξόρυξη δεδομένων από πλοηγήσεις χρηστών σε πύλες καταλόγων κατά τη διάρκεια μιας ολοκληρωμένης συνόδου (session) ενός χρήστη. Η πληροφορία αυτή μπορεί στη συνέχεια να αξιοποιηθεί από το διαχειριστή της πύλης, ο οποίος ενδιαφέρεται να παρακολουθήσει τις πορείες των χρηστών αυτής, με στόχο την αναδιοργάνωσή της ώστε να είναι πιο χρηστική. Για παράδειγμα, με τη γνώση των πιο δημοφιλών πλοηγήσεων, μπορεί η πύλη να αναδιαταχθεί έτσι ώστε να παρουσιάζονται πιο ψηλά οι πιο συχνά επισκεπτόμενες κατηγορίες. Ή, επίσης, με γνώση της χρονικής διάρκειας των πλοηγήσεων του κάθε χρήστη, μπορεί να φροντίσει ώστε η πύλη να παρέχει τις περισσότερες πληροφορίες σε τόσα επίπεδα, όσα μπορούν να προσπελαστούν από το χρήστη στο μέσο όρο των χρονικών διαρκειών.

Το σύστημα που αναπτύσσεται στη συγκεκριμένη διπλωματική εργασία μελετά τις πλοηγήσεις χρηστών στην πύλη dmoz, http://www.dmoz.org. Μπορεί όμως να προσαρμοστεί και να είναι εξίσου αποτελεσματικό και για οποιαδήποτε άλλη πύλη καταλόγου. Ονομάζεται “NaviMoz”, από σύντμηση των λέξεων “Navigation” και “Dmoz”. Το σύστημα αποθηκεύει τις πλοηγήσεις των χρηστών σε μια Βάση Δεδομένων και στη συνέχεια χειρίζεται τη Βάση αυτή μέσω διάφορων ερωτήσεων. Έτσι πραγματοποιείται η «εξόρυξη δεδομένων», αποτέλεσμα της οποίας είναι οι ομαδοποιήσεις των χρηστών με βάση τη συνολική πορεία των επισκέψεών τους. Στη διπλωματική εργασία πραγματοποιούνται οι ερωτήσεις και

λαμβάνονται τα αποτελέσματα μέσω μιας κατάλληλα διαμορφωμένης διασύνδεσης για το διαχειριστή της πύλης.

Ο πιο απλός τύπος ερωτήσεων είναι αυτός της ταυτοποίησης χρηστών. Για παράδειγμα, ο διαχειριστής μπορεί να βρει τις πλοηγήσεις συγκεκριμένων χρηστών σε διάφορες χρονικές περιόδους. Οι ερωτήσεις αυτές δεν προσφέρουν σημαντική υποστήριξη σε διαδικασίες εξόρυξης δεδομένων. Είναι όμως χρήσιμες για την μεμονωμένη παρατήρηση της συμπεριφοράς και των γενικότερων ενδιαφερόντων συγκεκριμένων χρηστών κατά την διάρκεια της αναζήτησης.

Μια πιο σύνθετη κατηγορία ερωτήσεων που υποστηρίζουν εξόρυξη δεδομένων είναι αυτές που λαμβάνουν μια πλοήγηση-πρότυπο και επιστρέφουν τις πλοηγήσεις που σχετίζονται κατά διάφορους τρόπους με αυτή. Ένα παράδειγμα είναι η εύρεση εκείνων των εργασιών που είναι κατά ένα ποσοστό όμοιες με τη δοσμένη. Ο διαχειριστής εισάγει μια πλοήγηση, έστω την /Arts/Music. Επίσης εισάγει ένα ποσοστό %, ενδεικτικό του βαθμού ομοιότητας μεταξύ των πλοηγήσεων. Το σύστημα επιστρέφει όλες εκείνες τις πλοηγήσεις που είναι κατά τουλάχιστον το συγκεκριμένο ποσοστό όμοιες με τη δοσμένη καθώς και τους χρήστες που τις έχουν πραγματοποιήσει. Η ομοιότητα υπολογίζεται με χρήση μετρικής απόστασης που συνδυάζει *δομική πληροφορία πλοήγησης* και *πληροφορία περιεχομένου κατηγοριών*. Έτσι, στο παραπάνω παράδειγμα, το σύστημα μπορεί να επιστρέψει την πλοήγηση Arts/Radio/Music ως παρόμοια με την /Arts/Music, αλλά να μην επιστρέψει την Arts/Radio.

Τέλος, υπάρχει και μια τρίτη κατηγορία ερωτήσεων που υποστηρίζουν ομαδοποιήσεις πλοηγήσεων και χρηστών. Αυτή είναι και η πιο ενδιαφέρουσα κατηγορία, και ταυτόχρονα η πιο δύσκολη στην υλοποίηση. Μια ερώτηση που υπάγεται στην κατηγορία αυτή είναι η *συσταδοποίηση (clustering)* των πλοηγήσεων και των χρηστών που τις πραγματοποίησαν, δηλαδή η δημιουργία ομάδων πλοηγήσεων χρηστών που μοιάζουν μεταξύ τους. Και εδώ η μετρική απόστασης που χρησιμοποιείται για την συσταδοποίηση συνδυάζει *δομική πληροφορία πλοήγησης* και *πληροφορία περιεχομένου κατηγοριών*. Οι τεχνικές συσταδοποίησης που έχουν υλοποιηθεί είναι οι: K-Μέσων (k-Means) και του Μονού Συνδέσμου (Single Link Hierarchical Clustering). Άλλο παράδειγμα ερώτησης είναι η εύρεση των πλοηγήσεων εκείνων που έχουν πραγματοποιηθεί τις περισσότερες φορές και άρα είναι και οι πιο δημοφιλείς. Τέλος, ένα άλλο παράδειγμα ερώτησης της κατηγορίας αυτής είναι η εύρεση των πιο αναποφάσιστων χρηστών του συστήματος. Για παράδειγμα, ο χρήστης που ακολούθησε /Arts/Music/Pop/Music/Pop/Concerts/Pop/Music/Rock/Concerts είναι πιο αναποφάσιστος από αυτόν που ακολούθησε /Arts/Music/Rock/Concerts.

Η κύρια διαφορά του συστήματος NaviMoz που υλοποιήθηκε στην διπλωματική αυτή σε σχέση με άλλα συστήματα καταγραφής συμπεριφοράς χρηστών στον Ιστό είναι ότι τα τελευταία εξετάζουν τα αρχεία καταγραφής του Web Server (Web logs). Η παρατήρηση της

συμπεριφοράς χρηστών στηρίζεται στις λέξεις των συνδέσμων των σελίδων που οι χρήστες διάβασαν, και είναι καταγραμμένες στα αρχεία αυτά, καθώς και σε λέξεις κλειδιά που χαρακτηρίζουν τις σελίδες των συνδέσμων. Στο σύστημα NaviMoz όμως, για να εξαχθούν συμπεράσματα σε σχέση με την συμπεριφορά των χρηστών, εξετάζονται *οι πλοηγήσεις των χρηστών σε κατηγορίες πριν φτάσουν στην τελική επιλογή συνδέσμων για τις ιστοσελίδες*. Στόχος του συστήματος NaviMoz είναι:

(α) η αναδιοργάνωση της πύλης ώστε να εξυπηρετούνται καλύτερα οι χρήστες στη διάρκεια της αναζήτησης. Για παράδειγμα, με τη γνώση των πιο δημοφιλών πλοηγήσεων, μπορεί η πύλη να αναδιαταχθεί έτσι ώστε να παρουσιάζονται πιο ψηλά οι πιο συχνά επισκεπτόμενες κατηγορίες. Επίσης, με τη γνώση των πιο αναποφάσιστων χρηστών, καθώς και των πλοηγήσεων που αυτοί ακολούθησαν, μπορεί να αναδιοργανωθεί το κομμάτι εκείνο της πύλης όπου παρατηρείται τέτοια συμπεριφορά.

(β) η παρατήρηση συμπεριφορών χρηστών. Για παράδειγμα, ερευνάται το χρονικό διάστημα που αφιερώνουν οι χρήστες στην ιεραρχία καταλόγου, συμπέρασμα που μπορεί να χρησιμοποιηθεί από το διαχειριστή για τη δημιουργία αναλόγου με το χρόνο μήκους μονοπατιών, προκειμένου να βρεθεί η πληροφορία.

(γ) η εξαγωγή προσωπικών προτιμήσεων. Για παράδειγμα, για κάθε χρήστη μπορούν να βρεθούν οι πλοηγήσεις που πραγματοποίησε. Ή για διάφορες ομάδες χρηστών μπορούν να προσδιοριστούν τα κοινά τους ενδιαφέροντα μέσω του προσδιορισμού των κοινών τους πλοηγήσεων. Αυτό το συμπέρασμα μπορεί να χρησιμοποιηθεί γενικότερα για την εύρεση των θεμάτων εκείνων που ενδιαφέρουν περισσότερο συγκεκριμένους χρήστες ή συγκεκριμένες ομάδες χρηστών πυλών καταλόγων.

1.2 Οργάνωση του τόμου

Η διπλωματική εργασία οργανώνεται στα παρακάτω κεφάλαια:

Στο Κεφάλαιο 2 γίνεται μια επιλεκτική παρουσίαση προηγούμενων ερευνητικών εργασιών που σχετίζονται με το αντικείμενο της διπλωματικής, εκ των οποίων δύο αναλύονται εκτενέστερα.

Στο Κεφάλαιο 3 παρουσιάζονται οι βασικοί ορισμοί και το θεωρητικό υπόβαθρο του θέματος. Επεξηγούνται οι διαδικασίες της εξόρυξης δεδομένων, της συσταδοποίησης και της εύρεσης της δομικής απόστασης δύο συμβολοακολουθιών..

Στο Κεφάλαιο 4 αναλύονται οι απαιτήσεις από το σύστημα και παρουσιάζεται η αρχιτεκτονική του. Παρουσιάζονται δηλαδή τα υποσυστήματα τα οποία το αποτελούν αλλά και οι εφαρμογές που εκτελεί κάθε υποσύστημα.

Στο κεφάλαιο 5 παρουσιάζεται η υλοποίηση του συστήματος μέσω της περιγραφής των κλάσεων που χρησιμοποιούνται. Επίσης αναφέρονται εν συντομία ο τρόπος εγκατάστασης του συστήματος καθώς και οι πλατφόρμες και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν.

Στο Κεφάλαιο 6 παρουσιάζεται ένα σενάριο χρήσης του συστήματος και, όπου είναι δυνατό, γίνεται σύγκριση των αποτελεσμάτων με τα αναμενόμενα.

Στο Κεφάλαιο 7 επιχειρείται μια σύνοψη της εργασίας και παρουσιάζονται τα συμπεράσματα αυτής. Επίσης γίνεται μια αναφορά σε πιθανές μελλοντικές επεκτάσεις του συστήματος που υλοποιήθηκε.

Τέλος, στο Κεφάλαιο 8 παρουσιάζεται η σχετική βιβλιογραφία.

2

Περιγραφή Θέματος

Σε αυτό το κεφάλαιο παρατίθεται μια σύντομη περιγραφή μερικών εργασιών, σχετικών με το αντικείμενο αυτής της διπλωματικής. Επίσης, σημειώνεται ο στόχος της παρούσας διπλωματικής.

2.1 Σχετικές εργασίες

Εδώ θα δοθεί η περιγραφή διαφόρων σχετικών εργασιών και του τρόπου με τον οποίο αντιμετωπίζουν παρόμοια ζητήματα.

2.1.1 Εξόρυξη δεδομένων από βιολογικά δεδομένα

Ένας από τους ταχύτατα αναπτυσσόμενους κλάδους στον οποίον έχουν πραγματοποιηθεί αρκετές σχετικές ερευνητικές εργασίες, είναι ο κλάδος της Βιοπληροφορικής. Πολλά από τα συστήματα που έχουν αναπτυχθεί πραγματοποιούν εξόρυξη πάνω σε πληροφορία αποθηκευμένη σε μια Βάση Δεδομένων. Τα βιολογικά δεδομένα έχουν σε αρκετές περιπτώσεις παρόμοια δομή με τις πλοηγήσεις τις οποίες επεξεργάζομαι στη διπλωματική εργασία. Για παράδειγμα, τα μόρια, οι πρωτεΐνες και άλλες βιολογικές οντότητες, ουσιαστικά αποτελούνται από ακολουθίες συστατικών κατά τον ίδιο τρόπο που οι πλοηγήσεις αποτελούνται από ακολουθίες ιστοσελίδων. Έτσι, η εξόρυξη γνώσης και οι όποιες ομαδοποιήσεις, πραγματοποιούνται με τρόπο παρεμφερή με αυτόν που χρησιμοποιώ στη

διπλωματική εργασία. Μερικά από τα συστήματα τα οποία έχουν αναπτυχθεί παρουσιάζονται στη συνέχεια.

Στην εργασία [KNO+03], εξετάζονται ακολουθίες βιολογικών αντιδράσεων με στόχο την ομαδοποίηση βιολογικών δεδομένων. Πιο συγκεκριμένα, το σύστημα αυτό έχει σαν στόχο να διατηρήσει, οπτικοποιήσει και πρωτίστως αναλύσει λειτουργίες των οργανισμών.

Καταρχήν, σε μοριακό επίπεδο, ορίζει ως «βιοχημικά μονοπάτια» (biochemical pathways) τα περίπλοκα δίκτυα που συνιστούν οι μοριακές αντιδράσεις που πραγματοποιούνται στον οργανισμό. Ο βασικός σκοπός του συστήματος αυτού είναι να επεξεργαστεί τις λειτουργίες των βιολογικών συστημάτων και τις συμπεριφορές των ζώντων οργανισμών μέσα από τη μελέτη αυτών των pathways. Μία σημαντική καινοτομία του συστήματος αυτού είναι ότι η λήψη της γονιδιακής πληροφορίας γίνεται, όχι απευθείας από το DNA αλλά από βιοχημικές οδούς στις οποίες συμμετέχουν τα αντίστοιχα γονίδια. Η αναγκαιότητα για κάτι τέτοιο γίνεται κατανοητή αν λάβουμε υπόψη μας ότι η ακολουθία με την οποία παρουσιάζονται τα γονίδια στο DNA δεν αντικατοπτρίζει το πλαίσιο μέσα στο οποίο δρουν τα γονίδια (δηλαδή, γονίδια των οποίων οι λειτουργίες σχετίζονται, συνήθως δεν είναι φυσικά ομαδοποιημένα στο DNA, σχετίζονται όμως μέσω των βιολογικών οδών). Επιπλέον, παρέχει γραφικά εργαλεία, τα οποία με κατάλληλο σχεδιασμό της Βάσης Δεδομένων, επιτρέπουν στο χρήστη να οπτικοποιήσει τα δεδομένα των βιοχημικών οδών σε διάφορα επίπεδα και να υποβάλλει προαποφασισμένα queries.

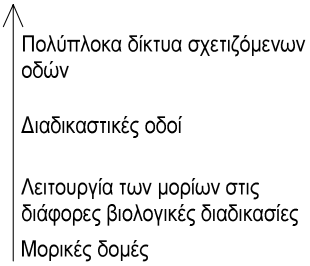
Η συγκεκριμένη εργασία ταξινομεί τις βιολογικές οδούς σε τρεις κλάσεις:

1) Βιοχημικές και μεταβολικές.

2) Αντιγραφής, διακανονισμού και πρωτεϊνοσύνθεσης.

3) Μετατροπής σήματος.

Για κάθε μια κλάση, οι πληροφορίες που ενδιαφέρουν είναι διαφορετικές. Στη Βάση Δεδομένων κρατούνται για παράδειγμα πληροφορίες που έχουν να κάνουν με την ταυτότητα των αντιδρώντων στοιχείων, των προϊόντων, των ενεργοποιητών, των συνενζύμων, πρότυπα RNA και πρωτεϊνικής έκφρασης και άλλες. Το παρακάτω σχήμα παρουσιάζει σχηματικά την ιεραρχία πάνω στην οποία είναι δομημένες οι πληροφορίες μας και με βάση την οποία υποβάλλονται τα queries του χρήστη.



Σχήμα 2.1: Ιεραρχία οργάνωσης των πληροφοριών

Όσον αφορά την αρχιτεκτονική του Pathways Database System, αυτό έχει 3 στρώματα:

Το πάνω-πάνω στρώμα είναι το GUI το οποίο αποτελεί τη διασύνδεση χρήστη. Εκεί ο χρήστης υποβάλλει τα ερωτήματά του. Το δεύτερο στρώμα είναι αποτελείται από τα λεγόμενα “Service Subsystems”. Το στρώμα αυτό δέχεται τα δεδομένα του χρήστη και αφού συνδεθεί με τη Βάση Δεδομένων, εξάγει τη ζητούμενη πληροφορία και την επιστρέφει σε μορφή XML. Το κατώτερο στρώμα ουσιαστικά είναι μια πλήρως λειτουργική Βάση Δεδομένων όπου αναζητείται η πληροφορία μετά από αίτημα του χρήστη. Η Βάση αυτή είναι οργανωμένη πάνω σε ένα μοντέλο Οντοτήτων-Συσχετίσεων. Οι βασικές οντότητες είναι i) Η μοριακή οντότητα, δηλαδή οποιοδήποτε μόριο, πρωτεΐνη, ένζυμο και άλλα, ii) Η διαδικασία που ουσιαστικά είναι μια βιοχημική αντίδραση και iii) Οι βιολογικές οδοί («μονοπάτια») δηλαδή οι συνδυασμοί αυτοί των αντιδράσεων έτσι ώστε ξεκινώντας από μια μοριακή οντότητα ή μια αντίδραση να φτάνουμε τελικά σε κάποιο επιθυμητό προϊόν. Αξίζει να σημειωθεί ότι τα βασικά δομικά στοιχεία είναι τα i και ii, ακριβώς επειδή το μοντέλο είναι δυναμικό(δηλαδή δεν είναι όλες οι βιολογικές οδοί προκαθορισμένες).

Στη Βάση Δεδομένων κάθε «μονοπάτι» αναπαρίσταται με γράφο όπου οι κόμβοι παριστάνουν μοριακές οντότητες, οι ακμές παριστάνουν βιοχημικές αντιδράσεις και η κατεύθυνση της ακμής υποδεικνύει ποια είναι τα αντιδρώντα στοιχεία και ποια τα προϊόντα. Η κατηγοριοποίηση των οντοτήτων δεν είναι σαφώς καθορισμένη, δηλαδή μια οντότητα μπορεί να ανήκει σε περισσότερες από μια κατηγορίες. Ένα παράδειγμα είναι το μόριο του νερού (H_2O), το οποίο μπορεί να κατηγοριοποιηθεί ως «βασικό μόριο» το οποίο είναι στοιχειώδες σε πολλές βιοχημικές ακολουθίες. Μπορεί όπως να καταταχθεί και στην κατηγορία «καταναλώσιμες οντότητες», υποδεικνύοντας ότι προέρχεται από εξωτερικές του οργανισμού πηγές.

Ο χρήστης του συστήματος επιλέγει από μια δενδρική δομή, η οποία αναπαριστά την ιεραρχία και αποτελεί συστατικό μιας οθόνης χρήστη, κάποια βιοχημική οδό, κάποια αντίδραση ή κάποια μοριακή οντότητα. Μόλις κάνει την επιλογή του, αν αυτή είναι κάποιο

μονοπάτι ή αντίδραση, παρουσιάζεται στην οθόνη οπτικοποίησης η αντίστοιχη δομή σε μορφή γράφου. Έχοντας την κατάλληλη καθοδήγηση μέσω της διασύνδεσης χρήστη, μπορεί να κατασκευάσει μια ολοκληρωμένη ερώτηση, μέσα από κατάλληλη επιλογή οντοτήτων και παραμέτρων. Οι δυνατοί τύποι των ερωτήσεων είναι οι παρακάτω:

1)Ερωτήσεις πάνω σε ένα συγκεκριμένο βιοχημικό μονοπάτι ή βιοχημική αντίδραση ή μοριακή οντότητα. Αυτού του τύπου οι ερωτήσεις ζητούν μόνο μια παράμετρο, συνήθως το όνομα της οντότητας (του μονοπατιού, της αντίδρασης και της μοριακής οντότητας αντίστοιχα), και είναι συγκεκριμένα για κάθε οντότητα. Δοσμένης της οντότητας, βρίσκονται τα συστατικά που σχετίζονται με αυτή με τον τρόπο που όρισε ο χρήστης. Για παράδειγμα, μπορεί ο χρήστης να δώσει τα αντιδρώντα στοιχεία μιας αντίδρασης και να ζητήσει τα προϊόντα αυτής.

2) Ερωτήσεις οι οποίες ζητούν από το χρήστη να επιλέξει τιμές για κάποιες παραμέτρους ή να θέσει συγκεκριμένες συνθήκες. Ένα χαρακτηριστικό παράδειγμα αυτού του τύπου είναι τα λεγόμενα “neighborhood queries”(“ερωτήσεις γειτονίας”). Ο χρήστης μπορεί να επιλέξει μια οντότητα για την οποία υπάρχει μενού με προ-αποφασισμένες ερωτήσεις. Σε κάθε μια από τις ερωτήσεις αυτές, του ζητείται να συμπληρωθεί ένα πλήθος παραμέτρων σε διαδοχικές φόρμες. Έτσι λοιπόν, όταν για παράδειγμα ο χρήστης επιθυμεί να γνωρίζει ποιο συστατικό είναι «κοντά» σε μια δεδομένη αντίδραση, με την έννοια ότι μετά από κάποια βήματα παράγεται το συστατικό αυτό, του ζητείται μέσω μιας φόρμας να επιλέξει τον αριθμό των επιθυμητών αυτών βημάτων.

3)Τα λεγόμενα “path queries” τα οποία ζητούν από το χρήστη να δώσει δύο μοριακές οντότητες κι ένα μονοπάτι (μέσω κατάλληλων επιλογών από τη διαπροσωπεία). Επιστρέφουν τη διαδρομή που συνδέει τις οντότητες αυτές μέσα στο μονοπάτι.

Ένα παράδειγμα query είναι : *«Βρες μια λίστα από όλα τα αντιδρώντα, τα προϊόντα, τα συν-απαιτούμενα στοιχεία εισόδου και παράλληλα στοιχεία εξόδου τα οποία λαμβάνουν μέρος σε κάθε διαδικασία ενός δοσμένου μονοπατιού»*. Σκοπός όλων των queries είναι οι κατά διαφόρους τρόπους ομαδοποιήσεις των στοιχείων της Βάσης Δεδομένων.

- Το “Pathways Database System” που περιγράφηκε συγκρινόμενο με το σύστημα που αναπτύχθηκε στη διπλωματική μου εργασία , έχει τη βασική ομοιότητα ότι πραγματοποιεί ομαδοποιήσεις βιολογικών δεδομένων, δηλαδή βιοχημικών μονοπατιών, βιοχημικών αντιδράσεων ή μοριακών οντοτήτων, όπως και το σύστημα NaviMoz πραγματοποιεί ομαδοποιήσεις πλοηγήσεων χρηστών μιας πύλης καταλόγου. Πραγματοποιεί εργασίες αντίστοιχες με κάποιες του συστήματος μου, όπως για παράδειγμα, δοθείσης μιας μοριακής οντότητας ψάχνει να βρει τις αντιδράσεις στις οποίες συμμετέχει. Κάτι αντίστοιχο πραγματοποιεί και το σύστημα NaviMoz με την εργασία εύρεσης των πλοηγήσεων εκείνων που αποτελούν υπέρ-σύνολο μιας

δοσμένης. Επίσης, το “Pathways Database System”, έχει καταχωρημένες τις αντιδράσεις με βάση τα συστατικά που συμμετέχουν σε αυτές, και σε ανώτερο επίπεδο έχει ομαδοποιημένα τα βιοχημικά μονοπάτια με βάση τις αντιδράσεις που τα απαρτίζουν. Ουσιαστικά δηλαδή, έχει δημιουργήσει συστάδες αντιδράσεων και μοριακών οντοτήτων, έτσι ώστε όταν ο χρήστης ζητήσει τα συστατικά εκείνα που σχετίζονται με μια συγκεκριμένη οντότητα, το σύστημα να επεξεργάζεται τη συστάδα στην οποία ανήκει η οντότητα αυτή.

- Μια άλλη ομοιότητα των δύο συστημάτων είναι ότι για το “Pathways Database System”, η δομή που αναπαριστά τα βιολογικά δεδομένα είναι δομή γράφου. Έτσι, μία οντότητα (μοριακή, αντίδραση ή βιολογική οδός) μπορεί να συναντάται σε πολλαπλά σημεία του γράφου, όπως και στο σύστημα NaviMoz μια σελίδα βρίσκεται κάτω από πολλαπλές κατηγορίες.

Μεταξύ των δύο συστημάτων όμως υπάρχουν και βασικές διαφοροποιήσεις:

- Τα δεδομένα εδώ είναι βιολογικά ενώ στο σύστημά μου είναι Web δεδομένα. Αυτό σχετίζεται άμεσα με τις δυνατές εργασίες που μπορούν να πραγματοποιηθούν για την εξόρυξη δεδομένων από αυτά. Για παράδειγμα, μπορεί να ζητηθούν τα προϊόντα μιας αντίδρασης ή τα συστατικά που την καταλύουν. Αντίθετα, στο σύστημά μου δεν υπάρχουν τέτοιες δυνατότητες- δυνατότητες δηλαδή παράπλευρων δεδομένων μιας πλοήγησης. Επίσης, μπορεί να ζητηθούν πληροφορίες που σχετίζονται με ένα συστατικό όταν αυτό συμμετέχει ως αντιδρών είτε ως προϊόν μιας αντίδρασης.
- Το “Pathways Database System” δεν πραγματοποιεί συσταδοποίηση (clustering) με βάση κάποιο γνωστό αλγόριθμο , όπως πραγματοποιείται στο σύστημα NaviMoz με χρήση του Single-Link και του K-Means. Αντίθετα βρίσκει τις «γειτονιές» των οντοτήτων με τη δημιουργία ενός υπέρ-γράφου που αναπαριστά τη Βάση Δεδομένων, και την εφαρμογή των αλγορίθμων DFS και BFS.

Μια άλλη εργασία ή οποία σχετίζεται με εξόρυξη πληροφορίας από βιολογικά δεδομένα είναι η [KAO04].Η εργασία αυτή προτείνει μια μέθοδο για εξαγωγή προτάσεων, οι οποίες περιέχουν πληροφορία αλληλεπίδρασης μεταξύ πρωτεϊνικών οντοτήτων, με βάση τη δομή των πρωτεϊνών, η οποία βρίσκεται καταχωρημένη σε μια Βάση Δεδομένων. Είναι γνωστό ότι οι πρωτεΐνες αλληλεπιδρούν με άλλα χημικά συστατικά. Η συγκεκριμένη εργασία εξηγεί τη σημαντικότητα της συλλογής των πληροφοριών που σχετίζονται με τις αλληλεπιδράσεις αυτές στην ανάλυση του πλαισίου λειτουργίας των πρωτεϊνών. Στο μοντέλο το οποίο προτείνει η εργασία αυτή, οι αντιδράσεις μοντελοποιούνται με μια ακολουθία προτάσεων η οποία βασίζεται στην πληροφορία της αλληλεπίδρασης. Παρατηρείται ότι σε μια δομή πρωτεϊνικής σύνθεσης τα υπολείμματα που παρουσιάζονται σε προτάσεις που σχετίζονται με την αλληλεπίδραση είναι κοντά με τις οντότητες με τις οποίες αλληλεπιδρούν. Η φυσική

απόσταση μεταξύ τους μπορεί να υπολογιστεί με βάση τη δομή τους και την πληροφορία η οποία μπορεί να ληφθεί από τη δομή αυτή, και η οποία είναι αποθηκευμένη σε μια Βάση Δεδομένων. Η γειτνίαση των υπολειμμάτων αυτών αποτελεί μια ένδειξη ότι η πρόταση στην οποία περιέχονται παρέχει και την απαραίτητη πληροφορία αλληλεπίδρασης. Συνοπτικά, η μέθοδος που χρησιμοποιείται έχε ως εξής: Αρχικά, το υπόλοιπο της αντίδρασης και το συμμετέχον στην αλληλεπίδραση συστατικό επιλέγονται από μια πρόταση. Στη συνέχεια, υπολογίζεται η μεταξύ τους απόσταση και αν αυτή είναι μικρότερη από κάποιο κατώφλι, T_{Dc}, η πρόταση αυτή θεωρείται πια ότι σχετίζεται με την πληροφορία της αλληλεπίδρασης. Η εργασία αυτή υπολογίζει τους βαθμούς ομοιότητας στηριζόμενη σε δομική πληροφορία και σε απόσταση μεταξύ των δομών, όπως και το σύστημα NaviMoz, το οποίο υλοποιείται στη διπλωματική εργασία, υπολογίζει την ομοιότητα μεταξύ των πραγματοποιούμενων πλοηγήσεων. Η διαφορά τους έγκειται στο μέτρο της μεταξύ των εξεταζόμενων στοιχείων αποστάσεων. Το παρόν σύστημα εξετάζει μόνο τη δομική απόσταση των πρωτεϊνικών προτάσεων ενώ το σύστημα της διπλωματικής βασίζεται εκτός από τη δομική απόσταση και στο περιεχόμενο των πλοηγήσεων.

Η εργασία [PLB+04], ασχολείται με τον τομέα της ανάλυσης δεδομένων γονιδιακής έκφρασης. Στον τομέα αυτό, πολλοί ερευνητές έχουν επισημάνει πρόσφατα την πολλά υποσχόμενη εφαρμογή τεχνικών αναγνώρισης προτύπων, όπως η εξόρυξη γνώσης με βάση συσχετιστικούς κανόνες ή με βάση πίνακες boolean τιμών οι οποίοι κωδικοποιούν τις ιδιότητες των γονιδίων. Για να είναι αυτές οι προσεγγίσεις όσο το δυνατό περισσότερο αποδοτικές, ένα βήμα που πρέπει να ληφθεί, και του οποίου η επίδραση στην ποιότητα και τη σχετικότητα των εξαγόμενων προτύπων είναι κρίσιμη, είναι η κωδικοποίηση των ιδιοτήτων των γονιδίων, η οποία απαιτείται να διαφοροποιεί την πληροφορία έκφρασης μεταξύ των γονιδίων. Στην παρούσα εργασία μελετάται η επίδραση των παραγόντων διαφοροποίησης μέσω μιας σύγκρισης των δενδρογραμμάτων που προκύπτουν από την εφαρμογή ενός ιεραρχικού αλγορίθμου συσταδοποίησης, ο οποίος βασίζεται στη μη επεξεργασμένη πληροφορία έκφρασης των γονιδίων αλλά και στους πίνακες boolean τιμών που προαναφέρθηκαν. Εδώ παρατηρείται μια σημαντική διαφορά από το σύστημα NaviMoz, το οποίο εφαρμόζει τους αλγορίθμους Κ Μέσων και Μονού Συνδέσμου βασιζόμενο στις αποστάσεις μεταξύ των πλοηγήσεων. Η εν λόγω εργασία, βασιζόμενη στην εισαγωγή ενός νέου μέτρου ομοιότητας, διαφορετικού από τη δομική απόσταση σε συνδυασμό με την απόσταση περιεχομένου που χρησιμοποιήθηκε στη διπλωματική, και σε πρακτικές αξιολόγησης πολλών γονιδιακών εκφράσεων, προτείνει μια μέθοδο για την επιλογή της μιας έναντι της άλλης τεχνικής διαχωρισμού καθώς και για την επιλογή των παραμέτρων της, δοθέντος ενός συγκεκριμένου συνόλου δεδομένων.

2.1.2 Εξόρυξη δεδομένων με τη χρήση Web Logs

Μια άλλη κατηγορία εργασιών οι οποίες ασχολούνται με θέματα σχετικά με τη διπλωματική αυτή εργασία, είναι αυτές οι οποίες χρησιμοποιούν τα Web Logs για να εξάγουν χρήσιμα δεδομένα, κατά τρόπο αντίστοιχο που εγώ εξετάζω τις ακολουθίες των σελίδων που επισκέφτηκαν οι χρήστες του συστήματος κατά τη διάρκεια μιας ολοκληρωμένης πλοήγησης (session). Υπάρχει πληθώρα τέτοιων εργασιών και παρακάτω θα παρατεθούν μερικές μόνο από αυτές, αρκετές για να τοποθετήσουν τη διπλωματική εργασία εντός ενός ευρύτερου πλαισίου εργασιών.

Αρχικά, στην εργασία [PPP+04], παρουσιάζεται ένας αποδοτικός τρόπος για την ομαδοποίηση χρηστών του Web σε κοινότητες με βάση τις προτιμήσεις τους με στόχο την προσωποποίηση των υπηρεσιών του Web. Συγκεκριμένα γίνεται χρήση ενός συνδυασμού από μεθόδους clustering , μιας μεθόδου agglomerative clustering και μιας usage mining, οι οποίες είναι βασισμένες στο περιεχόμενο των Web σελίδων.

Είναι γεγονός ότι ο μεγάλος όγκος της πληροφορίας δυσχεραίνει την πλοήγηση του χρήστη σε περιοχές που τον ενδιαφέρουν. Η συγκεκριμένη εργασία παρουσιάζει έναν τρόπο διευκόλυνσης των πλοηγήσεων των χρηστών. Ουσιαστικά δημιουργεί κοινότητες χρηστών βασιζόμενη σε πληροφορίες οι οποίες προέρχονται από εξόρυξη δεδομένων από το Web και όχι από πληροφορίες που εισάγει ο χρήστης χειρωνακτικά σε κατάλληλες φόρμες, των οποίων η εγκυρότητα δεν μπορεί να πιστοποιηθεί, όπως συνέβαινε ως τώρα. Το αποτέλεσμα χρησιμοποιείται έτσι ώστε ο κάθε χρήστης να έχει τη δυνατότητα να ξεκινά την πλοήγησή του στο Web από τη σελίδα της ομάδας στην οποία ανήκει, εξασφαλίζοντας ότι θα φτάσει στην επιθυμητή πληροφορία γρηγορότερα.

Για τη δημιουργία των Web communities χρησιμοποιούνται δεδομένα τα οποία συλλέγονται από Proxy Servers μιας κεντρικής υπηρεσίας του Web. Η διαδικασία αυτή αποτελείται από τα εξής τρία βήματα:

- *Συλλογή δεδομένων και προεπεξεργασία*

  Στο στάδιο αυτό πραγματοποιείται συλλογή και καθαρισμός των δεδομένων, καθώς και χαρακτηρισμός τους με βάση το περιεχόμενο των Web σελίδων. Πιο αναλυτικά, τα δεδομένα λαμβάνονται από τα log files των proxy servers. Τα δεδομένα αυτά είναι πολλά σε όγκο αλλά και πολύ διαφορετικά μεταξύ τους, καθώς προέρχονται από πλοηγήσεις των χρηστών στο σύνολο του Web. Στο στάδιο αυτό εξετάζεται το log file και απορρίπτεται η λανθασμένη πληροφορία καθώς και η πληροφορία η σχετική με multimedia. Στη συνέχεια, με χρήση μιας μεθόδου hierarchical agglomerative clustering γίνεται η ομαδοποίηση των ιστοσελίδων και η δημιουργία μιας ιεραρχίας./* Οι σελίδες αντιμετωπίζονται σαν αρχεία (documents)*/ Αξίζει να σημειωθεί ότι ενώ μέχρι τώρα

χρησιμοποιούνταν μέθοδοι ταξινόμησης (classification) για τη δημιουργία της ιεραρχίας (όπως της ιεραρχίας του yahoo), δηλαδή ο αριθμός των ομάδων ήταν γνωστός από πριν, εδώ οι ομάδες δημιουργούνται δυναμικά. Κάθε ιστοσελίδα αντιστοιχεί σε ένα διάνυσμα χαρακτηριστικών με δυαδικές τιμές, όπου κάθε χαρακτηριστικό κωδικοποιεί την παρουσία ή μη ενός συγκεκριμένου όρου στη σελίδα. Στην τελική ιεραρχία, οι κόμβοι αντιστοιχούν στις κατηγορίες που ανακτήθηκαν. Κάθε κατηγορία εκπροσωπείται από μια ακολουθία αριθμών, ενδεικτικών του path που ακολουθήθηκε για να φτάσει ο χρήστης στην κατηγορία αυτή. Ένα παράδειγμα είναι η ακολουθία 1.2.8.19 όπου το 1 αντιστοιχεί στη ρίζα του δένδρου.

- *Εξαγωγή Web κοινοτήτων*

Στο στάδιο αυτό, με τη χρήση των διανυσμάτων που περιγράφηκαν προηγουμένως, δημιουργούνται μοντέλα κοινοτήτων χρηστών. Αυτό γίνεται με τη χρήση του αλγορίθμου “Community Directory Miner”(CDM), ο οποίος αποτελεί μια επεκταμένη έκδοση του cluster mining algorithm, με τρόπο που να εφαρμόζεται σε δεδομένα που λαμβάνονται από ιεραρχίες. Για να εφαρμόσει τον αλγόριθμο αυτό, χρησιμοποιεί ένα γράφο για κάθε χρήστη του οποίου οι κόμβοι αντιστοιχούν στις θεματικές κατηγορίες που εξήχθησαν στο προηγούμενο στάδιο. Ο αλγόριθμος βρίσκει τις συνεκτικές συνιστώσες του γράφου αυτού, οπότε ανακαλύπτει και πορείες κοινού ενδιαφέροντος του χρήστη. Ο γράφος αυτός είναι ένας γράφος με βάρη $G(A,E,W_A,W_E)$, όπου οι κόμβοι A αντιστοιχούν στις θεματικές κατηγορίες, οι ακμές E αντιστοιχούν στο συσχετισμό των κατηγοριών (δηλαδή μια ακμή μεταξύ δύο κατηγοριών σημαίνει ότι ο χρήστης πήγε από τη μια κατηγορία στην άλλη) Τα βάρη στους κόμβους W_A και στις ακμές W_E αντιστοιχούν στην ύπαρξη και συνύπαρξη αντίστοιχα των συστατικών.

Συνήθως, ο γράφος αυτός έχει μεγάλη συνεκτικότητα. Για το λόγο αυτό χρησιμοποιείται κάποιο κατώφλι (threshold), το οποίο αποτελεί έναν αριθμό βάρους ακμής και είναι ενδεικτικό του βαθμού συνεκτικότητας του γράφου. Οι ακμές που έχουν βάρος μικρότερο από το κατώφλι αυτό κλαδεύονται με αποτέλεσμα τη δημιουργία συνεκτικών συνιστωσών στο γράφο. Οι συνεκτικές αυτές συνιστώσες αντιστοιχούν στις κοινότητες χρηστών. Όσο αυξάνεται η τιμή κατωφλίου, τόσο περισσότερες είναι και οι κοινότητες. Αξίζει να σημειωθεί ότι ένας χρήστης μπορεί να ανήκει σε περισσότερες από μια κοινότητες. Για τον τρόπο συμπλήρωσης του γράφου η συχνότητα επίσκεψης κάθε κατηγορίας από το χρήστη υπολογίζεται ως το άθροισμα των συχνοτήτων επίσκεψης από το χρήστη της κατηγορίας αυτής και των συχνοτήτων επίσκεψης των κατηγοριών-παιδιών της κατηγορίας αυτής.

Ο αλγόριθμος CDM μπορεί να συνοψιστεί στα παρακάτω βήματα:

1) Υπολογισμός των συχνοτήτων επίσκεψης των κατηγοριών που αντιστοιχούν στα βάρη των κόμβων.

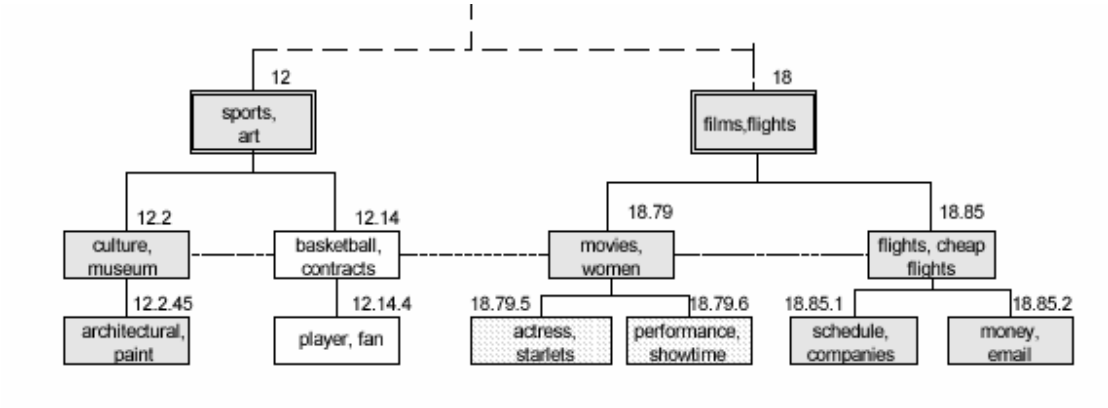
2) Υπολογισμός των συχνοτήτων σύμπραξης μεταξύ κατηγοριών που αντιστοιχούν στα βάρη των ακμών.

3) Ενημέρωση των βαρών των κόμβων (δηλαδή των κατηγοριών), προσθέτοντας τις συχνότητες των παιδιών τους. Η διαδικασία αυτή επαναλαμβάνεται αναδρομικά καθώς κατεβαίνουμε στην ιεραρχική δομή του καταλόγου. Κατά τον ίδιο τρόπο ενημερώνονται και τα βάρη των ακμών, όπως επίσης και τα βάρη όλων των προγόνων των σελίδων που παρουσιάζονται σε μία διαδρομή χρήστη (session).

4) Εύρεση όλων των μεγίστων κλικών στον αρχικό γράφο των κατηγοριών.

- *Μετά-επεξεργασία και αποτίμηση του μοντέλου*

Τα patterns που εξερευνήθηκαν μπορούν να αντιστοιχιστούν σε τοπικά δένδρα. Το παρακάτω αποτελεί ένα παράδειγμα μιας Web κοινότητας.



Σχήμα 2.2: Παράδειγμα μιας Web κοινότητας

Τα γκρι κουτιά είναι οι κατηγορίες που ανήκουν στη συγκεκριμένη κοινότητα και τα άσπρα όλες οι υπόλοιπες.

Οι μετρικές που χρησιμοποιούνται για την αποτίμηση του προτεινόμενου μοντέλου είναι οι εξής δύο:

1) Η *διαφορετικότητα* (distinctiveness). Τα μοντέλα κατηγοριών που είναι όσο το δυνατό πιο διαφορετικά μεταξύ τους είναι αυτά που παρουσιάζουν και το μεγαλύτερο ενδιαφέρον. Αν M είναι ένα σύνολο μοντέλων κοινοτήτων, J είναι το πλήθος των μοντέλων στο M, A_j το πλήθος των κατηγοριών που χρησιμοποιείται στο j-οστό μοντέλο και A' το πλήθος των διαφορετικών κατηγοριών που εμφανίζονται σε τουλάχιστον ένα μοντέλο, η διαφορετικότητα δίνεται από τον τύπο : Διαφορετικότητα(M)= $|A'|/ \sum_j |Aj|$.

2) Το πλήθος A' των διαφορετικών κατηγοριών, το οποίο είναι ενδεικτικό του βαθμού στον οποίο οι χρήστες επικεντρώνονται σε ένα υποσύνολο από κατηγορίες.

Το παρόν σύστημα έχει κάποιες διαφορές από αυτό που έχω υλοποιήσει στη διπλωματική εργασία, οι οποίες παρατίθενται παρακάτω:

- Καταρχήν χρησιμοποιεί διαδοχικά δύο μεθόδους clustering:αρχικά agglomerative clustering για να δημιουργήσει την ιεραρχία και μετά εφαρμόζει τον αλγόριθμο CDM. Η πραγματοποίηση διαδοχικών clustering διαφοροποιεί την εργασία αυτή από τις προηγούμενές της αλλά και από τη δική μου. Στην εργασία μου παίρνω έτοιμη την ιεραρχία και στη συνέχεια εφαρμόζω clustering τεχνικές στις πορείες που ακολουθούν οι χρήστες κατά τις πλοηγήσεις τους στην ιεραρχία αυτή.
- Ο αλγόριθμος που χρησιμοποιεί το paper αυτό για clustering είναι αρχικά ένας agglomerative clustering και μετά ο CDM. Και οι δύο είναι content-based (βασίζονται στο περιεχόμενο). Στη διπλωματική μου εργασία πραγματοποιώ clustering με τους αλγορίθμους Single-Link και K-Means, βασιζόμενη στο περιεχόμενο αλλά και στη δομή των μονοπατιών που ακολουθούν οι χρήστες.
- Στη συγκεκριμένη εργασία, οι πληροφορίες αντλούνται από τα log files των χρηστών. Οι ομαδοποιήσεις γίνονται με βάση τις διαφορετικές σελίδες, μία-μία που επισκέφτηκαν οι χρήστες. Αντιθέτως, εγώ, στη διπλωματική μου εργασία, δε χρησιμοποιώ log file αλλά όλη την πορεία του χρήστη όπως αυτή είναι αποθηκευμένη στη Βάση Δεδομένων του συστήματος, και ομαδοποιώ με βάση τη συνολική αυτή πορεία.
- Τέλος, η εργασία αυτή καταλήγει στη δημιουργία κοινοτήτων χρηστών και στην αξιολόγηση αυτών μέσω κάποιων μετρικών. Στη διπλωματική μου εργασία απλά παρατίθενται οι ομαδοποιήσεις των χρηστών. Οι ομαδοποιήσεις αυτές μπορούν να επεξεργαστούν περαιτέρω και να οδηγήσουν σε βελτιώσεις των προσφερόμενων υπηρεσιών Web.

Μία άλλη εργασία η οποία βασίζεται σε πληροφορίες που μπορούν να εξαχθούν από τη χρήση Web logs είναι η [FS05]. Σκοπός της εργασίας αυτής είναι ο έξυπνος σχεδιασμός ψηφιακών διοικητικών Web πυλών, έτσι ώστε να παρέχουν διαδικτυακές υπηρεσίες (e-services) αντίστοιχες με τις απαιτήσεις του κάθε χρήστη. Σύμφωνα με την εν λόγω μελέτη, για την ανάπτυξη της ηλεκτρονικής διοίκησης (“digital government” ή “e-government”), υψίστης σημασίας είναι το πώς θα δοθεί ένα αποτελεσματικό σύνολο επιλογών στους πολίτες και στις διάφορες κοινότητες των πολιτών μέσω των διαδιτυακών πυλών. Η αρχική σελίδα μιας πύλης ηλεκτρονικής διοίκησης είναι το σημείο έναρξης της αναζήτησης ηλεκτρονικών

υπηρεσιών, μέσω μερικών υπέρ-συνδέσμων (links). Προκειμένου όμως να οδηγηθούν σωστά οι χρήστες, χρειάζεται να εξεταστούν εξαντλητικά όλοι οι πιθανοί συνδυασμοί των παρεχόμενων ηλεκτρονικών υπηρεσιών, πράγμα πολύ ακριβό υπολογιστικά. Η συγκεκριμένη εργασία διαιρεί ένα Web log σε περιόδους (sessions), όπου κάθε περίοδος αντιστοιχεί σε μια ακολουθία web προσβάσεων που έχουν πραγματοποιηθεί από τον ίδιο χρήστη. Με βάση αυτό, αναπτύσσει μια ευριστική μέθοδο, εν ονόματι “Service Finder” για την αποτελεσματική επιλογή των υπερ-συνδέσμων που θα είναι διαθέσιμοι μέσω μιας πύλης ηλεκτρονικής διοίκησης. Η μέθοδος “Service Finder” βασίζεται σε συνδυασμό προτύπων τα οποία εξάγονται από τη δομή της πύλης ηλεκτρονικής διοίκησης και από εξόρυξη την Web logs τα οποία καταγράφουν τις συμπεριφορές των χρηστών. Εξερευνά δηλαδή τις σχέσεις ανάμεσα στη δομή της πύλης και στις προσβάσεις στους διάφορους ιστοτόπους της και στη συνέχεια υπολογίζει πόσο προτιμούνται οι υπερσύνδεσμοι, αρχικά ο καθένας ατομικά και στη συνέχεια σαν ομάδες. Αντίθετα, στη διπλωματική εργασία δεν χρησιμοποιείται καθόλου το Web log, αλλά μόνο τις πλοηγήσεις των χρηστών τις οποίες και επεξεργάστηκα ως σύνολο. Για την αξιολόγηση της μεθόδου χρησιμοποιούνται τρεις μετρικές, η αποτελεσματικότητα, η επάρκεια και η χρηστικότητα.

Μία άλλη επίσης σχετική εργασία είναι αυτή των F. Toolan και N. Kusmerick [TF02], οι οποίοι βασίζονται σε τεχνικές εξόρυξης γνώσης από τη χρήση του διαδικτύου (Web usage mining) με στόχο τη δημιουργία προσωποποιημένων χαρτών ιστοσελίδων (Site maps), εξειδικευμένων στα ενδιαφέροντα κάθε χρήστη. Η βασική τους πρόκληση είναι η εύρεση της χρυσής τομής μεταξύ απλότητας (όπου σε κάθε υπερσύνδεσμο θα υπάρχει απλά το σχετικό περιεχόμενο) και κατανόησης (όπου σε κάθε υπερσύνδεσμο υπάρχει επαρκής πληροφορία, τόση ώστε οι χρήστες να μπορούν να κατανοήσουν ποια η σχέση της παρεχόμενης πληροφορίας με την ολική δομή του site). Στη συγκεκριμένη εργασία αναπτύσσονται δύο αλγόριθμοι, ένας που απεικονίζει απλά τα πιο μικρά μονοπάτια κι ένας που λαμβάνει μετά από εξαντλητική εξέταση του log του εξυπηρετητή (server) τα πιο δημοφιλή μονοπάτια, οι οποίοι συγκρίνονται με μια νέα προσέγγιση. Η προσέγγιση αυτή πραγματοποιεί εξόρυξη δεδομένων από το log του εξυπηρετητή για την εύρεση δημοφιλών τμημάτων μονοπατιών, τα οποία στη συνέχεια μπορούν να συναρμολογηθούν δυναμικά και να δημιουργήσουν μεγαλύτερα δημοφιλή μονοπάτια. Τα πειράματα που πραγματοποίησαν σε δύο μεγάλους ιστοτόπους επιβεβαίωσαν ότι η τεχνική αυτή είναι αποτελεσματική, με την έννοια του ότι επιτρέπει τη δυναμική ανάπτυξη προσωποποιημένων χαρτών ιστοσελίδων, οι οποίες περιέχουν διάσημα, σε αντίθεση με απλά μικρά μονοπάτια-πλοηγήσεις. Η διαφορά από τη διπλωματική μου εργασία έγκειται στο ότι ο στόχος της [TF02] είναι η δημιουργία προσωποποιημένων χαρτών ιστοσελίδων ενώ στη διπλωματική απλά παρατίθενται οι ομαδοποιήσεις των χρηστών οι οποίες μπορούν στη συνέχεια να χρησιμοποιηθούν για

ποικίλους σκοπούς, ένας από τους οποίους είναι και η προσωποποίηση της πύλης. Επίσης, άλλη μια διαφορά είναι ότι η παρούσα εργασία, ασχολείται μόνο με ένα από τα προβλήματα με τα οποία ασχολούμαι στη διπλωματική εργασία, με την εύρεση δημοφιλών μονοπατιών με τη διαδοχική εφαρμογή των αλγορίθμων που αναφέρθηκαν. Αντίθετα, στη διπλωματική δε χρησιμοποιείται αυτός ο συνδυασμός αλγορίθμων.

Επίσης, διαφόρων άλλων ειδών τεχνικές συσταδοποίησης και ακολουθιακής εξερεύνησης πλοηγήσεων έχουν χρησιμοποιηθεί για τη δημιουργία μοντέλων χρηστών, κατ' αντιστοιχία με τις μεθόδους που έχω χρησιμοποιήσει στην παρούσα διπλωματική εργασία για την ομαδοποίηση των χρηστών. Στην πρώτη περίπτωση παράδειγμα αποτελεί η εργασία [KJ00], όπου χρησιμοποιούνται robust ασαφείς αλγόριθμοι συσταδοποίησης (robust fuzzy clustering methods) και συσταδοποίηση με χρήση κανόνων συσχέτισης, σε αντίθεση με τη διπλωματική εργασία όπου χρησιμοποιούνται οι αλγόριθμοι των K Μέσων και του Μονού Συνδέσμου. Στη δεύτερη περίπτωση παράδειγμα αποτελεί η εργασία [SF98], όπου οι προσβάσεις που έχουν καταγραφεί στο log file αποθηκεύονται σε μια αποθήκη δεδομένων και στη συνέχεια εφαρμόζεται πάνω στα δεδομένα αυτά η γλώσσα ερωτήσεων MINT για εξόρυξη γνώσης από αυτά, και πιο συγκεκριμένα για την εύρεση των στατιστικών και δομικών ιδιοτήτων των προσβάσεων πιθανού ενδιαφέροντος. Στη διπλωματική εργασία δε χρησιμοποιείται κάποια εξειδικευμένη γλώσσα ερωτήσεων πέρα από την SQL. Τα μοντέλα αυτά χρησιμοποιούνται στη συνέχεια για την προσωποποίηση της ιστοσελίδας και για την παροχή προτεινόμενων υπέρ-συνδέσμων (links). Επίσης δεδομένα χρήσης έχουν επίσης συνδυαστεί με το περιεχόμενο ιστοσελίδων στην αναφορά [MDL+00]. Στην προσέγγιση αυτή, διάφορα προφίλ περιεχομένου έχουν δημιουργηθεί με τη χρήση τεχνικών συσταδοποίησης. Τα προφίλ περιεχομένου αντιπροσωπεύουν το ενδιαφέρον των χρηστών για πρόσβαση σε σελίδες με παρόμοιο περιεχόμενο και συνδυάζονται με αντίστοιχα προφίλ χρησιμοποίησης για να υποστηρίξουν τη διαδικασία της πρότασης συγκεκριμένων ιστοσελίδων. Η διαφορά εδώ από τη διπλωματική έγκειται στο ότι εξετάζει το περιεχόμενο των σελίδων που επισκέπτονται οι χρήστες, ενώ στη διπλωματική εξετάζονται οι πλοηγήσεις. Μια άλλη, πολύ ενδιαφέρουσα μέθοδος συσταδοποίησης χρησιμοποιείται στην εργασία[AR03]. Εδώ τα δεδομένα που συλλέγονται από τα log files, αφού καθαριστούν, χρησιμοποιούνται για την ταυτοποίηση των μοντέλων χρήσης, με βάση μοντέλα της συμπεριφοράς αποικιών μυρμηγκιών (ant-colonies), των οποίων η ικανότητα για αυτό-οργάνωση είναι πολύ μεγάλη. Η συγκεκριμένη εργασία χρησιμοποιεί έναν ant-clustering αλγόριθμο για την εύρεση των Web μοντέλων χρήσης (data clusters) και έναν γενετικό αλγόριθμο για την ανάλυση των τάσεων των επισκεπτών. Αντίθετα, στη διπλωματική εργασία χρησιμοποιούνται οι αλγόριθμοι των K Μέσων και του Μονού Συνδέσμου. Τέλος, μία προσέγγιση της προσωποποίησης μιας ιστοθέσης κατά αυτόματο τρόπο, με τη βοήθεια μεθόδων εξόρυξης χρησιμοποίησης (usage mining), είναι το

σύστημα Montage [AH02]. Το σύστημα αυτό χρησιμοποιείται για να δημιουργήσει προσωποποιημένες πύλες (portals), τα οποία θα αποτελούνται πρωτίστως από υπέρ-συνδέσμους στις ιστοσελίδες που ένας συγκεκριμένος χρήστης έχει επισκεφθεί, οργανωμένες σε θεματικές κατηγορίες σύμφωνα με κάποιον κατάλογο ανοιχτού συστήματος (ODP-Open Directory Project), όπως είναι και ο κατάλογος του dmoz ο οποίος χρησιμοποιείται στη διπλωματική εργασία. Για τη δημιουργία του μοντέλου του χρήστη χρησιμοποιείται ένα πλήθος ευριστικών μετρικών, όπως είναι το ενδιαφέρον για μια ιστοσελίδα ή ένα θέμα, η πιθανότητα επανεπίσκεψης της ιστοσελίδας και άλλες. Αντίθετα, στη διπλωματική εργασία χρησιμοποιούνται μέθοδοι συσταδοποίησης, οι δομικές αποστάσεις μεταξύ των πλοηγήσεων, νέοι αλγόριθμοι (όπως ο αλγόριθμος εύρεσης των πιο αναποφάσιστων χρηστών του συστήματος) σε συνδυασμό με κατάλληλες ερωτήσεις (queries) στη Βάση Δεδομένων, για την πραγματοποίηση της εξόρυξης γνώσης η οποία μπορεί στη συνέχεια να χρησιμοποιηθεί για τη δημιουργία μοντέλων χρηστών.

2.2 Στόχος

Σκοπός της διπλωματικής εργασίας είναι η εξόρυξη δεδομένων από τις πλοηγήσεις που έχουν πραγματοποιηθεί στις θεματικές κατηγορίες μιας πρότυπης πύλης καταλόγου ,κατά τη διάρκεια μιας ολοκληρωμένης συνόδου (session) ενός χρήστη. Οι πορείες των χρηστών εξετάζονται ως σύνολο και όχι μόνο ως τελικός στόχος. Ένα τέτοιο σύστημα θα μπορεί να είναι ιδιαίτερα χρήσιμο για το διαχειριστή της πύλης, ο οποίος θα έχει τη δυνατότητα να μελετά τις πλοηγήσεις των εγγεγραμμένων στην πύλη χρηστών και, με βάση τις πληροφορίες που θα λαμβάνει, να παίρνει κατάλληλες αποφάσεις για τη βελτίωση των δυνατοτήτων της πύλης. Οι εργασίες τις οποίες μπορεί να ζητήσει ο διαχειριστής να υλοποιηθούν μέσω του συστήματος είναι οι παρακάτω:

1) **Ερωτήσεις ταυτοποίησης χρηστών**

   Εμφάνιση των πλοηγήσεων που έχει πραγματοποιήσει κάποιος χρήστης του συστήματος σε συγκεκριμένο χρονικό διάστημα, καθώς και των χρονικών στιγμών έναρξης και λήξης των πλοηγήσεων αυτών.

2) **Ερωτήσεις εξόρυξης δεδομένων**

   Εύρεση των πλοηγήσεων εκείνων που αποτελούν υπέρ-σύνολο μιας δοσμένης, που είναι ίδιες με μια δοσμένη και που είναι κατά ένα ποσοστό, το οποίο αποφασίζει ο διαχειριστής όμοιες με μια δοσμένη.

3) **Ερωτήσεις ομαδοποίησης πλοηγήσεων και χρηστών**

   Εμφάνιση των δημοφιλέστερων πλοηγήσεων, των περισσότερο αναποφάσιστων χρηστών και συσταδοποίηση των πλοηγήσεων που έχουν πραγματοποιηθεί σε

συγκεκριμένο χρονικό διάστημα, καθώς και των χρηστών που τις έχουν πραγματοποιήσει. Με τον όρο «συσταδοποίηση» εννοούμε την ομαδοποίηση των πλοηγήσεων των χρηστών ανάλογα με το πόσο μοιάζουν μεταξύ τους.

3

Θεωρητική Μελέτη

Στο κεφάλαιο αυτό παρατίθενται οι βασικοί ορισμοί και τα απαραίτητα θεωρητικά εργαλεία στα οποία στηρίχτηκα για την εκπόνηση της διπλωματικής εργασίας. Πιο συγκεκριμένα , αρχικά ορίζονται οι γράφοι πυλών καταλόγων, γιατί με έναν τέτοιο ασχολείται και η εργασία αυτή, το γράφο του dmoz. Στη συνέχεια δίνονται κάποιες γενικές πληροφορίες πάνω στη διαδικασία εξόρυξης δεδομένων, η οποία χρησιμοποιείται στην παρούσα διπλωματική. Ακολουθεί ορισμός και επεξήγηση της δομικής απόστασης μεταξύ δύο συμβολοακολουθιών, ένα μέγεθος υψίστης σημασίας από τη στιγμή που με βάση αυτό υπολογίζονται τα ποσοστά ομοιότητας των πλοηγήσεων των χρηστών. Τέλος γίνεται γενική αναφορά στις μεθόδους συσταδοποίησης (Clustering Techniques) και στη συνέχεια παρουσιάζονται οι αλγόριθμοι των K-Μέσων, του Μονού Συνδέσμου και του C-Index, οι οποίοι χρησιμοποιούνται από το σύστημα που αναπτύχθηκε για τη συσταδοποίηση των πλοηγήσεων και των χρηστών.

3.1 Γράφοι Πυλών Καταλόγων

Πύλες Καταλόγων ονομάζονται εκείνα τα πληροφοριακά συστήματα, όπως οι εταιρικές μνήμες, οι κατακόρυφες αθροίσεις και άλλα, τα οποία επιτρέπουν σε χρήστες διαφορετικής εμβέλειας όσον αφορά τις προτιμήσεις τους, να επιλέγουν, να ταξινομούν και να έχουν πρόσβαση, με τρόπο αποδοτικό, σε πολλαπλές πηγές πληροφόρησης (για παράδειγμα sites, αρχεία, δεδομένα) [ACK+01]. Αυτά τα συστήματα στηρίζουν την επιτυχία τους σε μετά-πληροφορίες που σχετίζονται με τα στοιχεία του καταλόγου. Με στόχο λοιπόν οι διαφορετικού τύπου χρήστες να έχουν αποδοτική πρόσβαση στη γνώση που τους ενδιαφέρει, οι Πύλες Καταλόγων εξομοιώνουν και οργανώνουν την πληροφορία με πολλούς τρόπους, οι οποίοι είναι πιο ευέλικτοι και αποδοτικοί από τους αντικειμενοστρεφείς, σχεσιακούς και βασισμένους σε δομή XML τρόπους. Θεωρούμε ότι οι πύλες καταλόγων οργανώνουν τα δεδομένα τους σε μορφή γράφου. **Γράφος G**, ονομάζεται ένα διατεταγμένο ζεύγος συνόλων (V, E) όπου V είναι ένα μη κενό σύνολο στοιχείων και E ένα σύνολο μη διατεταγμένων ζευγών του V. Πιο απλά, V είναι το σύνολο των κόμβων του γράφου και E το σύνολο των ακμών του. Η πύλη καταλόγου του Dmoz η οποία χρησιμοποιείται στη συγκεκριμένη

διπλωματική εργασία, αλλά και άλλες, έχουν τη δομή γράφου, δηλαδή ο χρήστης μπορεί να φτάσει σε μια κατηγορία ακολουθώντας διαφορετικά μονοπάτια. Αυτό σημαίνει ότι ο κάθε κόμβος μπορεί να έχει περισσότερες από μια προσπίπτουσες σε αυτόν ακμές κι έτσι μπορεί να φτάνει κανείς σε μια κατηγορία έχοντας ακολουθήσει διαφορετική διαδρομή από κάποιον άλλον. Για παράδειγμα για να φτάσει στη σελίδα “learning Chinese” μπορεί να έχει ακολουθήσει την πορεία Science/ Social science/ Linguistic/ Language learning ή την πορεία Computers / Software/ Educational/ Language.

3.2 Εξόρυξη δεδομένων

Στην εποχή μας η ψηφιακή πληροφορία έχει πλέον μπει στη ζωή μας, και έχουν αρχίσει ήδη να παρουσιάζονται προβλήματα χειρισμού της λόγω του μεγάλου όγκου της. Για το λόγο αυτό έχουν αρχίσει να αναπτύσσονται με ταχύτατους ρυθμούς οι τομείς της Ανακάλυψης δεδομένων σε Βάσεις Δεδομένων (Knowledge Discovery in Databases-KDD), και της εξόρυξη δεδομένων (Data Mining) [FPS96]. Οι τομείς αυτοί εισάγουν μια νέα γενιά υπολογιστικών τεχνικών και εργαλείων τα οποία υποστηρίζουν την εξαγωγή χρήσιμων δεδομένων από το συνεχώς επεκτάσιμο σύνολο των δεδομένων.

Είναι γεγονός ότι οι Βάσεις Δεδομένων, ή και οι ακόμα μεγαλύτερες Αποθήκες Δεδομένων είναι πανταχού παρόν στη ζωή μας. Πέρα από την απλή πρόσβαση στις πληροφορίες που περιέχουν, αξία έχει και η γνώση που μπορεί να συναχθεί από τα δεδομένα και να τεθεί σε εφαρμογή. Για παράδειγμα η Βάση Δεδομένων μιας καταναλωτικής εταιρείας μπορεί να υπαινίσσεται συσχετίσεις μεταξύ των πωλήσεων συγκεκριμένων προϊόντων και αντίστοιχων δημογραφικών ομάδων, κάτι το οποίο με τη σειρά του μπορεί να χρησιμοποιηθεί από τη διαφημιστική καμπάνια της εταιρείας, με στόχο την αύξηση του κέρδους της.

Το πρόβλημα με την εξαγωγή δεδομένων από μεγάλες Βάσεις Δεδομένων εμπεριέχει πολλά βήματα, τα οποία ποικίλλουν από την ανάκτηση και χειρισμό των δεδομένων μέχρι και τα στοιχειώδη, μαθηματικά και στατιστικά συμπεράσματα. Για το λόγο αυτό κρίνεται απαραίτητος ο αυτοματισμός στα όσα μπορούν να εξαχθούν από μια Βάση Δεδομένων.

Η εύρεση χρήσιμων προτύπων πληροφοριών είναι γνωστή με πολλά ονόματα, όπως «εξόρυξη δεδομένων». Με τον όρο KDD αναφερόμαστε στη συνολική διαδικασία του να ανακαλύπτουμε χρήσιμα δεδομένα από την πληροφορία, και η εξόρυξη δεδομένων αποτελεί απλώς ένα βήμα στη διαδικασία αυτή. Η διαδικασία KDD έχει εφαρμοστεί σε πολλούς τομείς, όπως μηχανική μάθηση, αναγνώριση προτύπων, τεχνητή νοημοσύνη και άλλα. Αυτό που πρέπει να τονιστεί εδώ είναι ότι η εξόρυξη δεδομένων πρέπει να έχει έναν απώτερο, καθορισμένο στόχο, να μην είναι δηλαδή τυφλή. Ένας γενικά αποδεκτός ορισμός για την KDD διαδικασία είναι ο παρακάτω:

«Ανακάλυψη δεδομένων σε Βάσεις Δεδομένων καλείται η ουσιώδης διαδικασία ταυτοποίησης έγκυρων, πρωτότυπων, πιθανώς χρήσιμων και πρωτίστως κατανοητών προτύπων στα δεδομένα».

Παρακάτω επεξηγούνται περαιτέρω οι όροι που χρησιμοποιούνται στον παραπάνω ορισμό.

Πρότυπο: Συγκεκριμένο μοντέλο ή δομή των δεδομένων. Στον ορισμό αυτό, τα δεδομένα συνίστανται σε ένα σύνολο γεγονότων και το πρότυπο είναι μια έκφραση σε κάποια γλώσσα που περιγράφει ένα υποσύνολο των δεδομένων (ή ένα μοντέλο που εφαρμόζεται σε αυτό το υποσύνολο). Για παράδειγμα, στην περίπτωση που αναφερόμαστε σε μια Ιατρική Βάση Δεδομένων, τα δεδομένα μπορεί να συνίστανται σε ασθένειες και στους ασθενείς που πάσχουν από αυτές και το πρότυπο να είναι μία συγκεκριμένη ασθένεια για την οποία ψάχνουμε να εξάγουμε περισσότερες πληροφορίες (όπως στην περίπτωση που ψάχνουμε ποιοι από τους ασθενείς πάσχουν από διαβήτη)

Διαδικασία: Ο όρος αυτός υπονοεί ότι υπάρχουν αρκετά βήματα τα οποία επαναλαμβάνονται σε διαδοχικές επαναλήψεις προκειμένου να φτάσουμε στην πληροφορία που αναζητούμε, συμπεριλαμβανομένων των: προετοιμασία των δεδομένων, αναζήτηση για πρότυπα, αποτίμηση των δεδομένων που εξάγουμε και τέλος καθαρισμός. Παραμένοντας στο προηγούμενο παράδειγμα της Ιατρικής Βάσης Δεδομένων, μπορεί να χρειαστεί ένας ασθενής να υποστεί πολλές εξετάσεις προκειμένου να βρεθούν οι ασθένειες από τις οποίες πάσχει καθώς και οι μεταξύ τους συσχετίσεις. Επιπλέον, πραγματοποιούνται πολλές ενημερώσεις σε περίπτωση που κάποιος ασθενής ξεπεράσει μια ασθένεια ή προσβληθεί από κάποια άλλη. Η αναζήτηση για πρότυπα μπορεί επίσης να επαναληφθεί για διάφορες ιδιότητες αυτών. Για παράδειγμα, μπορεί στη μία περίπτωση να αναζητούνται οι ασθενείς οι οποίοι πάσχουν από διαβήτη τύπου 1 και στην άλλη αυτοί που πάσχουν από διαβήτη τύπου 2.

Ουσιώδης: Η διαδικασία υποτίθεται ότι είναι ουσιώδης στο ότι υπολογίζει κλειστού-τύπου ποσότητες, που σημαίνει ότι πρέπει να περιλαμβάνει αναζήτηση στη δομή, στα μοντέλα, στα πρότυπα η σε παραμέτρους.

Τα πρότυπα που ανακαλύπτονται πρέπει να είναι έγκυρα για νέα δεδομένα με κάποιο βαθμό βεβαιότητας. Επίσης θέλουμε τα πρότυπα να είναι πρωτότυπα (τουλάχιστον για το σύστημα και κατά προτίμηση και για το χρήστη) και πιθανώς χρήσιμα για το χρήστη ή την εργασία που εκτελείται. Τέλος, πρέπει να είναι κατανοητά – αν όχι απευθείας, τότε σίγουρα μετά από κάποια προεπεξεργασία.

Ο παραπάνω ορισμός υπονοεί ότι μπορούμε να καθορίσουμε ποσοτικά μέτρα για να αξιολογήσουμε τα εξαγόμενα πρότυπα. Ένα πολύ σημαντικό κριτήριο, το πόσο ενδιαφέρον κρίνεται το πρότυπο (“interestingness”), είναι αυτό που συνήθως λαμβάνεται ως ένα συνολικό μέτρο της αξίας του, συνδυάζοντας εγκυρότητα, καινοτομία, χρηστικότητα και απλότητα. Η διαδικασία KDD αλληλεπιδρά με το χρήστη και αποτελείται από μια σειρά

βημάτων, μεταξύ των οποίων υπάρχουν πολλές αλληλεξαρτήσεις. Η βασική σειρά των βημάτων αυτών, η οποία δεν ακολουθείται πάντα ως έχει, είναι η παρακάτω:

1) Εκμάθηση του τομέα της εκάστοτε εφαρμογής.
2) Επιλογή του συνόλου-στόχου μεταβλητών ή δειγμάτων δεδομένων, το οποίο θα υποστεί επεξεργασία.
3) Καθαρισμός και προεπεξεργασία των δεδομένων.
4) Ελάττωση των διαστάσεων των δεδομένων και προβολή τους. Προσπάθεια ώστε τα δεδομένα να αναπαρασταθούν με λιγότερες μεταβλητές.
5) Επιλογή της λειτουργικότητας της διαδικασίας εξόρυξης δεδομένων: Περιλαμβάνει απόφαση πάνω στο στόχο του μοντέλου που προκύπτει από τον αλγόριθμο εξόρυξης δεδομένων (για παράδειγμα περίληψη, ταξινόμηση και συσταδοποίηση–clustering).
6) Επιλογή του κατάλληλου αλγορίθμου εξόρυξης δεδομένων ανάλογα με το μοντέλο και την πληροφορία που θέλουμε να εξάγουμε.
7) Εξόρυξη δεδομένων. Περιλαμβάνει αναζήτηση στα patterns ενδιαφέροντος για κάποια συγκεκριμένη αναπαράσταση ή για ένα σύνολο από τέτοιες αναπαραστάσεις. Περιλαμβάνει κανόνες ταξινόμησης στα δένδρα, παλινδρόμηση, συσταδοποίηση (clustering), μοντελοποίηση ακολουθιών και άλλα.
8) Μετάφραση των αποτελεσμάτων σε μορφή κατανοητή από τους χρήστες.
9) Χρησιμοποίηση της πληροφορίας που ανακαλύφθηκε.

Η μεγαλύτερη επικέντρωση γίνεται στο στάδιο εξόρυξης δεδομένων. Το στάδιο αυτό περιλαμβάνει ταίριασμα μοντέλων σε παρατηρούμενα δεδομένα ή εξαγωγή προτύπων από παρατηρούμενα δεδομένα. Τα αποτελέσματά του παίζουν το ρόλο της υπαινισσόμενης δεδομένων. Φυσικά, η διαδικασία της εξόρυξης δεδομένων υπόκειται σε περιορισμούς όσον αφορά το χώρο υπολογισμού, από τη στιγμή που τα πρότυπα που μπορούν να καταμετρηθούν σε ένα πεπερασμένο σύνολο δεδομένων είναι πιθανώς άπειρα και που η καταμέτρηση των προτύπων περιλαμβάνει κάποιου είδους αναζήτηση σε ένα μεγάλο χώρο. Η απόφαση του εάν τα μοντέλα αντανακλούν ή όχι χρήσιμα δεδομένα είναι μέρος της συνολικής KDD διαδικασίας, στην οποία απαιτείται και η ανθρώπινη κρίση. Οι αλγόριθμοι που χρησιμοποιούνται για εξόρυξη δεδομένων αποτελούνται κυρίως από τρία βασικά συστατικά:

- **Το μοντέλο.** Με τον όρο αυτό εννοούμε τη *λειτουργία* του μοντέλου, για παράδειγμα ταξινόμηση ή συσταδοποίηση και τον *τρόπο αναπαράστασης* του μοντέλου. Ένα μοντέλο περιέχει παραμέτρους που αποφασίζονται από τα δεδομένα.

Οι βασικότερες *λειτουργίες* του μοντέλου περιλαμβάνουν:

-Ταξινόμηση(classification): αντιστοίχιση ενός αντικειμένου σε μια κατηγορία από ένα σύνολο προ-αποφασισμένων κατηγοριών). Δηλαδή έστω ότι υπάρχουν ήδη οι κατηγορίες (α, ε, ο) και (κ, λ, μ, ν) οι οποίες έχουν διαχωρίσει τα παραπάνω γράμματα της αλφαβήτου σε φωνήεντα και σύμφωνα. Αν τώρα εισαχθεί το πρότυπο «τ» στο σύστημα, αυτό θα ταξινομηθεί στην ήδη υπάρχουσα κατηγορία των συμφώνων (κ, λ, μ, ν).

-Συσταδοποίηση (clustering): αντιστοίχιση ενός αντικειμένου σε μια κατηγορία-cluster από ένα σύνολο κατηγοριών το οποίο καθορίζεται δυναμικά από τα δεδομένα). Στο προηγούμενο παράδειγμα με τα φωνήεντα και τα σύμφωνα, εισάγονται όλα μαζί τα πρότυπα α, ε, ο, τ, κ, λ, μ, ν στο σύστημα. Οι κατηγορίες δεν προϋπάρχουν αλλά σχηματίζονται δυναμικά από το σύστημα. Τελικά καταχωρούνται τα πρότυπα στις σωστές ομάδες (α, ε, ο) και (κ, λ, μ, ν, τ).

-Παλινδρόμηση: αντιστοίχιση ενός αντικειμένου σε μια πραγματικής-τιμής μεταβλητή πρόβλεψης.

-Περίληψη: παρέχει μια συνεπτυγμένη περιγραφή για ένα υποσύνολο δεδομένων. Ένα παράδειγμα θα ήταν η μέση και η σταθερή παρέκκλιση όλων των πεδίων των δεδομένων.

-Μοντελοποίηση εξαρτήσεων:περιγραφή αξιοσημείωτων εξαρτήσεων, δομικών ή ποσοτικών, μεταξύ των μεταβλητών.

-Ανάλυση συνδέσμων: Καθορισμός σχέσεων μεταξύ διαφόρων πεδίων στη Βάση Δεδομένων. Ένα παράδειγμα εδώ είναι οι συσχετιστικοί κανόνες “association rules” οι οποίοι εφαρμόζονται κυρίως σε αγορές προϊόντων και περιγράφουν ποια αντικείμενα αγοράζονται συνήθως μαζί με άλλα αντικείμενα. Για παράδειγμα, όταν μια γυναίκα-πελάτης ενός καταστήματος αγοράζει μια τσάντα, είναι πιθανό να αγοράσει και παπούτσια.

-Ακολουθιακή ανάλυση: Μοντελοποίηση των καταστάσεων της διαδικασίας η οποία παράγει την ακολουθία ή εξαγωγή και δήλωση παρεκκλίσεων ανά πάσα στιγμή. Για παράδειγμα, μια ακολουθία μπορεί να είναι ότι μετά την επίσκεψη μιας συγκεκριμένης σελίδας Α, οι χρήστες του διαδικτύου επισκέπτονται πάντα τη σελίδα Β. Η πορεία της επίσκεψης είναι πάντα σε αυτή τη σειρά.

Αναπαράσταση του μοντέλου:

Πολλά μοντέλα αναπαράστασης περιλαμβάνουν δένδρα απόφασης και κανόνες, γραμμικά μοντέλα, μη γραμμικά μοντέλα, μεθόδους βασισμένες σε παραδείγματα και άλλα. Η αναπαράσταση του μοντέλου πρέπει να λαμβάνει υπόψη της την ευελιξία του μοντέλου στην αναπαράσταση των δεδομένων καθώς και την ικανότητα μετάφρασής του σε όρους κατανοητούς από τον άνθρωπο.

Επίσης υπάρχουν κάποια κριτήρια προτιμήσεων του μοντέλου τα οποία καθορίζουν το πόσο καλά ένα μοντέλο και οι παράμετροί του συναντούν τα κριτήρια της KDD διαδικασίας. Οι αλγόριθμοι αναζήτησης είναι δύο τύπων: παραμετρική αναζήτηση δοθέντος του μοντέλου και

αναζήτηση με βάση το μοντέλο στο χώρο των μοντέλων. Οι κανόνες βελτιστοποίησης των αλγορίθμων εξόρυξης δεδομένων βασίζονται σε σχετικά απλές τεχνικές βελτιστοποίησης (όπως για παράδειγμα η συνάρτηση κλίσης), αν και στην πραγματικότητα μπορούν να χρησιμοποιούνται και πιο επιτηδευμένες μέθοδοι. Το σημαντικό με τις διάφορες τεχνικές εξόρυξης δεδομένων που έχουν αναπτυχθεί και που συνεχώς αναπτύσσονται, είναι ότι μια τεχνική που είναι καλή για κάποιο πρόβλημα μπορεί για κάποιο άλλο να μη δίνει τα επιθυμητά αποτελέσματα. Άρα, η τεχνική που κάθε φορά χρησιμοποιείται εξαρτάται πρωτίστως από τον τύπο του προβλήματος.

Οι απώτεροι στόχοι της διαδικασίας εξόρυξης δεδομένων είναι η δημιουργία μοντέλων πρόβλεψης, περιγραφής (δηλαδή με ικανότητα ικανοποιητικής αντανάκλασης της πραγματικότητας) ή συνδυασμός των δύο. Το τελευταίο είναι και αυτό που προτιμάται πιο συχνά.

- **Το κριτήριο προτίμησης κάποιου μοντέλου**

  Το μοντέλο που προτιμάται εξαρτάται από τα δεδομένα. Το κριτήριο είναι συνήθως το πόσο καλά ταιριάζει η λειτουργία του μοντέλου με τα δεδομένα, περιέχοντας ίσως έναν παράγοντα μετρίασης για αποφυγή της περίπτωσης υπερεκπαίδευσης.

- **Ο αλγόριθμος αναζήτησης**

  Ο προσδιορισμός ενός αλγορίθμου για εύρεση συγκεκριμένων δομών και παραμέτρων, δοθέντων των δεδομένων, ενός μοντέλου (ή μιας οικογένειας μοντέλων) και ενός κριτηρίου προτίμησης, όπως αυτό ορίστηκε παραπάνω.

3.3 Δομική Απόσταση

Σύμφωνα με την αναφορά [WF74], η **δομική απόσταση (structural distance)** μεταξύ δύο συμβολοακολουθιών είναι ουσιαστικά ένας αριθμός μεταξύ 0 και 1, ο οποίος είναι μια ένδειξη του πόσο όμοιες είναι αυτές. Για παράδειγμα οι ακολουθίες "Arts/ Music/ Artists/ Young" και "/Arts/ Photography/ Artists/ Young" είναι κατά 0.286 όμοιες, αριθμός που δηλώνει μεγάλη ομοιότητα. Αντίθετα, οι συμβολοακολουθίες “Sports/ Multi-Sports/ Multi-Events /Multi-Sports/ Duathlon/ Events/ Duathlon/ Multi-Sports/ Triathlon /Cycling/ BMX/ Racing” και “Arts/ Photography/ Artists” είναι κατά 0.8 όμοιες, αριθμός που δηλώνει μεγάλη ανομοιότητα. Πιο συγκεκριμένα, δομική απόσταση, είναι η απόσταση μεταξύ δύο Strings όπως αυτή υπολογίζεται από την ακολουθία «συντακτικών λειτουργιών» (“edit operations”) με το μικρότερο κόστος, οι οποίες απαιτούνται για να μετατραπεί το ένα String στο άλλο. Η απόσταση αυτή υπολογίζεται σε χρόνο ανάλογο με το γινόμενο των μηκών των δύο Strings. Οι πράξεις–λειτουργίες που θεωρεί η συγκεκριμένη πηγή είναι οι:

i. Μετατροπή κάποιου χαρακτήρα του ενός String σ' έναν άλλον του άλλου String. Για παράδειγμα, για να μετατραπεί το String "xyxzw" στο "xsxzw" πρέπει ο χαρακτήρας y του πρώτου String να μετατραπεί στο χαρακτήρα s του δεύτερου.

ii. Διαγραφή ενός χαρακτήρα από κάποιο String. Για παράδειγμα, για να μετατραπεί το String "xzwwst" στο "xzwst" πρέπει από το πρώτο να διαγραφεί ο ένας χαρακτήρας w.

iii. Εισαγωγή ενός χαρακτήρα σε κάποια θέση μέσα στο String.Για παράδειγμα, για να μετατραπεί το String "yxzw" στο "ysxzw" πρέπει στο πρώτο να εισαχθεί ένας χαρακτήρας s μεταξύ του y και του x.

Δομική απόσταση

Πριν δοθεί ο ορισμός της δομικής απόστασης, θα δοθούν κάποιοι βοηθητικοί ορισμοί. Έστω λοιπόν:

A:Μια ακολουθία χαρακτήρων ή συμβόλων

A<i>:Ο i-οστός χαρακτήρας του A.

A<i:j>: Οι ενδιάμεσοι χαρακτήρες των A<i>,A<j>, συμπεριλαμβανομένων των άκρων.

A<i:j>=Λ όταν η ακολουθία A είναι κενή.

|A|: Το πλήθος των χαρακτήρων (το μήκος) του A.

Ορισμός 3.3.1:*Μια «συντακτική λειτουργία» είναι ένα ζευγάρι (a, b) από ακολουθίες διάφορων της κενής με μήκος μικρότερο ή ίσο του 1 και συνήθως συμβολίζεται ως* $a \rightarrow b$*. Η ακολουθία B προκύπτει μετά από εφαρμογή της πράξης* $a \rightarrow b$ *στην ακολουθία A, και γράφουμε* $A \Rightarrow B$ *μέσω της* $a \rightarrow b$*. Καλούμε την* $a \rightarrow b$ *μία πράξη μετατροπής όταν ισχύει* $a \neq \Lambda$, $b \neq \Lambda$*, μία πράξη διαγραφής όταν* $b=\Lambda$ *και μια πράξη εισαγωγής όταν* $a=\Lambda$*.*

Ορισμός 3.3.2:*Έστω S μια ακολουθία* $s_1,s_2,\ldots,s_m$ *από συντακτικές λειτουργίες. Μία S-παραγωγή ("S-derivation") από το A στο B είναι μια ακολουθία από strings* $A_0,A_1,\ldots,A_m$ *τέτοια ώστε* $A=A_0$ *και* $B=A_m$ *και* $A_{i-1} \Rightarrow A_i$ *μέσω της* s_i *για* $1<=i<=m$*. Λέμε ότι η S μετατρέπει το A στο B αν υπάρχει μια S-παραγωγή από το A στο B.*

Έστω τώρα ότι η γ είναι μια αυθαίρετη συνάρτηση κόστους, η οποία αντιστοιχίζει σε κάθε συντακτική λειτουργία a→b έναν θετικό αριθμό γ(a→b). Αν επεκτείνουμε το γ σε μία ακολουθία από συντακτικές λειτουργίες $S=s_1,s_2,\ldots, s_m$, θα έχουμε $\gamma(S)=\sum_{i=1}^{m}\gamma(si)$ (για m=0 ορίζουμε γ(S)=0). Έχοντας αυτά ως βάση μπορούμε να προχωρήσουμε στον παρακάτω ορισμό:

Ορισμός 3.3.3: *Ως «συντακτική απόσταση» δ(A, B) της ακολουθίας A από την ακολουθία B ορίζουμε το μικρότερο από όλα τα κόστη των ακολουθιών από συντακτικές λειτουργίες οι*

οποίες μετατρέπουν το A στο B. Επισήμως, δ(A, B)=min{γ(S)|S είναι μια συντακτική ακολουθία που μετατρέπει το A στο B}.

Ίχνη (Traces)

Για να διευκολυνθούμε στο πρόβλημα εύρεσης της συντακτικής απόστασης μεταξύ δύο Strings, A και B, ορίζουμε μια συνάρτηση κόστους πάνω σε κάποιες βοηθητικές δομές, οι οποίες καλούνται ίχνη. Ο επίσημος ορισμός του ίχνους δίνεται αμέσως παρακάτω:

Ορισμός 3.3.4: *Ένα ίχνος από το A στο B είναι μια τριπλέτα (T, A, B), όπου το T είναι ένα οποιοδήποτε σύνολο από διατεταγμένα ζεύγη ακεραίων (i, j) που ικανοποιούν τα παρακάτω κριτήρια:*

1)$1 \le i \le |A|$ *και* $1 \le j \le |B|$

2)Για δύο οποιαδήποτε διαφορετικά ζεύγη (i_1,j_1) *και* (i_2,j_2) *στο T:* *(a)*$i_1 \neq i_2$ *και* $j_1 \neq j_2$

(b)$i_1 < i_2$ *αννν* $j_1 < j_2$

Αποδεικνύεται ότι το δ(A, B) είναι ίσο με το ελαχίστου κόστους ίχνος από το A στο B, οπότε η προσοχή μας βρίσκεται πλέον στο να βρούμε το ίχνος αυτό. Διαισθητικά, ένα ίχνος είναι μια περιγραφή του πώς μια συντακτική ακολουθία S μετατρέπει το A στο B αγνοώντας τη σειρά με την οποία συμβαίνουν οι πράξεις καθώς και οποιονδήποτε πλεονασμό στην S.

<u>Παράδειγμα</u>

Στη συνέχεια εξηγείται η εύρεση του κόστους ενός ίχνους με τη βοήθεια κι ενός παραδείγματος. Έστω τα δύο παρακάτω Strings και ότι θέλουμε να μετατρέψουμε το A στο B.

String A: x y z w t w x z x

String B: y w x z x y x w

Όταν μια γραμμή ενώνει ένα στοιχείο του A με ένα του B σημαίνει ότι το στοιχείο του A μετατρέπεται, άμεσα ή μετά από την εφαρμογή κατάλληλων τελεστών, στο αντίστοιχο στοιχείο του B. Τα στοιχεία του A που δε σχετίζονται με τις γραμμές αντιστοιχούν σε διαγραφές από το A, ενώ τα στοιχεία του B που δε σχετίζονται με τις γραμμές αντιστοιχούν σε εισαγωγές στο A μέσω της S.

Έστω τώρα το ίχνος T από το A στο B. Έστω επίσης I και J τα σύνολα των θέσεων στα Strings A και B αντίστοιχα τα οποία δε σχετίζονται με τις γραμμές στο T. Ορίζουμε το κόστος στο T ως εξής:

$$\text{cost (T)}= \sum_{(i,j)\in T} \gamma(A < i > \rightarrow B < j >) + \sum_{i \in I} \gamma(A < i > \rightarrow \Lambda) + \sum_{j \in J} \gamma(\Lambda \rightarrow B < j >)$$

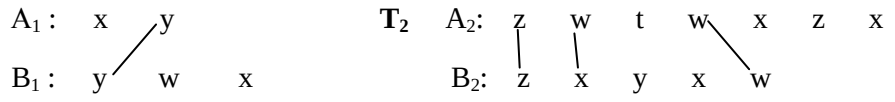
Επεξηγώντας την παραπάνω ισότητα, λέμε ότι το συνολικό κόστος είναι το άθροισμα του κόστους των πράξεων μετατροπής, διαγραφής και εισαγωγής που πραγματοποιούνται κατά τη μετατροπή του Α στο Β. Για τις πράξεις που έχουν σημειωθεί στο παραπάνω παράδειγμα το κόστος είναι 10.

Υπολογισμός της συντακτικής απόστασης

Για να προχωρήσουμε στον υπολογισμό της τελικής συντακτικής απόστασης πρέπει να λάβουμε υπόψη μας το γεγονός ότι τα ίχνη μπορούν να συντεθούν. Αν δηλαδή T_1 είναι ένα ίχνος από το Α στο Β και T_2 είναι ένα ίχνος από το Β στο Γ, τότε το $T=T_1 \circ T_2$ είναι ένα ίχνος από το Α στο Γ. Με βάση αυτό, αλλά και το παρακάτω θεώρημα, υπολογίζουμε τη συντακτική απόσταση.

Θεώρημα 1: $\delta(A, B)=\min\{cost(T)|T$ είναι ένα ίχνος από το Α στο Β$\}$

Έστω λοιπόν ότι θέλουμε να μετατρέψουμε μια ακολουθία Α σε μια άλλη Β. Αν $A=A_1A_2$, $B=B_1B_2$ και θεωρώντας ότι καμία γραμμή του Τ δεν ενώνει έναν χαρακτήρα του A_i με έναν άλλον του B_j ,για $i\neq j$, $i,j \in \{0,1\}$, ένα ίχνος (T, A, B) μπορεί να σπάσει σε δύο άλλα (T_1, A_1, B_1) και (T_2,A_2,B_2), όπως φαίνεται και στο παρακάτω σχήμα:



Επιπλέον ισχύει για το κόστος: $cost(T)=cost(T_1) + cost(T_2)$. Έτσι, αν Τ είναι το ελάχιστο ίχνος από το α στο Β, T_i θα είναι το ελάχιστο ίχνος από το A_i στο B_i . Επίσης, κάθε ίχνος από το Α στο Β μπορεί να σπάσει σε άλλα δύο,T_1 και T_2 με μήκος το πολύ 1 αλλά όχι και τα δύο 0. Έστω λοιπόν Α και Β δύο strings. A(i)=A<1:i>, B(j)=B<1:j> και D(i, j)=δ(A(i),B(j)), $0\leq i\leq|A|$, $0\leq j\leq|B|$. Από το θεώρημα 1, D(i, j) είναι επίσης το κόστος του ίχνους ελαχίστου κόστους από το A(i) στο B(j).

Θεώρημα 2: Ισχύει D(i, j)=min{D(i-1, j-1) + γ(A<i>→B<j>),

D(i-1, j)+ γ(A<i>→Λ),

D(i, j-1)+ γ(Λ→ B<j>)}

για όλα τα i, j τέτοια ώστε $1\leq i\leq|A|$, $1\leq j\leq|B|$.

Θεώρημα 3: $D(0,0)=0$, $D(i, 0)= \sum_{r=1}^{i} \gamma(A < r > \rightarrow \Lambda)$ και $D(0, j)= \sum_{r=1}^{j} \gamma(\Lambda \rightarrow B < r >)$,

με $1\leq i\leq|A|$, $1\leq j\leq|B|$.

Τα θεωρήματα και 2 δικαιολογούν γιατί ο παρακάτω αλγόριθμος υπολογίζει σωστά το D(i, j) για $0\leq i\leq|A|$, $0\leq j\leq|B|$.

Αλγόριθμος εύρεσης ελαχίστου κόστους μετατροπής του String A στο String B:

Ο <u>πίνακας D</u> είναι ένας δισδιάστατος πίνακας του οποίου κάθε στοιχείο D [i ,j] περιέχει το ελάχιστο κόστος για να πάμε από το A(i) στο B(j).

Θέσε D[0,0]=0.

Για i=1μέχρι |A| {/*Η πρώτη στήλη του D αντιστοιχεί σε διαγραφές από το A */

Θέσε D[i, 0]=D[i-1, 0]+γ(A<i>→Λ)

}

Για j=1 μέχρι |B| {/*Η πρώτη γραμμή του D αντιστοιχεί σε εισαγωγές στο B*/

Θέσε D[0,j]=D[0,j-1]+γ(Λ→B<j>)

}

Για i=1μέχρι |A|{

Για j=1 μέχρι |B|{

m1=D[i-1, j-1]+ + γ(A<i>→B<j>)

m2= D[i-1, j]+ γ(A<i>→Λ)

m3= D[i, j-1]+ γ(Λ→ B<j>)

D[i, j]=min(m1, m2, m3)

}

Το D[|A|, |B|] είναι το ελάχιστο κόστος που επιστρέφει ο αλγόριθμος. Αξίζει να σημειωθεί ότι δεν ενδιαφέρουν οι πράξεις με τις οποίες πάμε από το A στο B, αλλά το πλήθος αυτών.

<u>Παράδειγμα</u>

Έστω ότι θέλουμε να υπολογίσουμε τη δομική απόσταση μεταξύ των συμβολοακολουθιών A="xyzwtwxzx" και B="ywxzxyxw", οι οποίες εξετάστηκαν και στο προηγούμενο παράδειγμα. Ο αλγόριθμος εξετάζει πρώτα το στοιχείο x της πρώτης συμβολοακολουθίας με το στοιχείο y της δεύτερης και δίνει m1=1, m2=2, m3=2. Έτσι το στοιχείο D[1][1] του πίνακα παίρνει την τιμή 1. Στη συνέχεια, ο αλγόριθμος εξετάζει το στοιχείο x της πρώτης συμβολοακολουθίας με το στοιχείο w της δεύτερης. Τώρα οι τιμές των m1, m2 και m3 είναι αντίστοιχα m1=2, m2=3, m3=2. Έτσι το στοιχείο D[1][1] του πίνακα παίρνει την τιμή 2. Με διαδοχική εκτέλεση όλων των βημάτων κατά τον τρόπο αυτό, προκύπτει ότι το D[9][8], το οποίο αντιστοιχεί στο ελάχιστο κόστος και το οποίο επιστρέφει ο αλγόριθμος, είναι 6. Πράγματι, ο αριθμός αυτός είναι το ελάχιστο κόστος μετατροπής της πρώτης ακολουθίας στη δεύτερη. Οι πράξεις που πραγματοποιούνται και που οδηγούν στο ελάχιστο αυτό κόστος είναι οι εξής:

- Διαγραφή του x από την ακολουθία A.
- Το y παραμένει ως έχει.

- Διαγραφή του z από την ακολουθία A.
- Το w παραμένει ως έχει.
- Το t της A μετατρέπεται στο x της B.
- Το w της A μετατρέπεται στο z της B.
- Το x της A παραμένει ως έχει.
- Το z της A μετατρέπεται στο y της B.
- Το x της A παραμένει ως έχει.
- Εισαγωγή του w στη B.

Οι πράξεις εισαγωγής, διαγραφής και μετατροπής είναι 6.

Παρατηρούμε, τέλος, ότι η πολυπλοκότητα του αλγορίθμου αυτού είναι O(|A| * |B|).

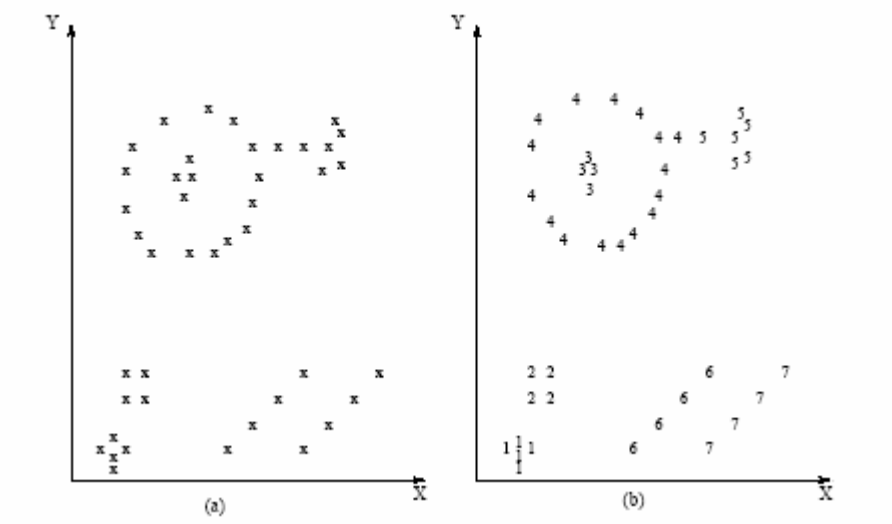
3.4 Συσταδοποίηση (Clustering)

3.4.1 Γενικά

Με τον όρο «συσταδοποίηση» (“Clustering”) εννοούμε τη χωρίς επίβλεψη ταξινόμηση των προτύπων (παρατηρήσεων, δεδομένων ή διανυσμάτων χαρακτηριστικών) σε ομάδες (συστάδες-clusters). Τονίζεται ιδιαίτερα ότι οι ομάδες αυτές δεν προϋπάρχουν αλλά αποφασίζονται από τον αλγόριθμο κατά δυναμικό τρόπο. Δεν πρόκειται δηλαδή για ομάδες με καθορισμένα συστατικά οι οποίες εξετάζονται για να διαπιστωθεί σε ποια από αυτές ανήκει ένα νέο δεδομένο που εισέρχεται στο σύστημα. Η χρηστικότητα της συσταδοποίησης, ως ένα από τα βασικά βήματα της διερευνητικής ανάλυσης δεδομένων, είναι πολύ μεγάλη. Οι τεχνικές ανάλυσης δεδομένων μπορούν να διαχωριστούν σε αναγνωριστικές και σε επικύρωσης, ανάλογα με τα μοντέλα που είναι διαθέσιμα για το συγκεκριμένο σύνολο δεδομένων. Το κλειδί και στις δύο περιπτώσεις είναι η ομαδοποίηση των μετρήσεων που έχουν ληφθεί. Η ανάλυση των συστάδων (cluster analysis) είναι η οργάνωση ενός συνόλου από πρότυπα, τα οποία συνήθως αντιπροσωπεύονται ως ένα διάνυσμα μετρήσεων ή ένα σημείο σε έναν πολυδιάστατο χώρο, σε ομάδες, η δημιουργία των οποίων γίνεται με βάση την ομοιότητα μεταξύ των προτύπων. Διαισθητικά τα πρότυπα ενός έγκυρου cluster είναι περισσότερο όμοια μεταξύ τους από ότι σε σύγκριση με ένα πρότυπο που ανήκει σε κάποιο άλλο cluster. Ένας αλγόριθμος συσταδοποίησης πρέπει να τοποθετεί τα όμοια πρότυπα στην ίδια συστάδα και τα διαφορετικά σε διαφορετικές συστάδες.

Παράδειγμα

Ένα παράδειγμα συσταδοποίησης δίνεται στην παρακάτω εικόνα, όπου αριστερά παρουσιάζεται το αρχικό σύνολο των στοιχείων πριν τη συσταδοποίηση και δεξιά η καταχώρηση των στοιχείων σε συστάδες.



Σχήμα 3.1: Συσταδοποίηση δεδομένων

Η διαφορά των τεχνικών συσταδοποίησης (clustering) από τις τεχνικές ταξινόμησης (classification) είναι ότι οι δεύτερες ανήκουν στην ευρύτερη κατηγορία της μάθησης υπό επίβλεψη και τα πρότυπα είναι κατηγοριοποιημένα εκ των προτέρων. Το πρόβλημα έγκειται στο ότι για ένα νέο πρότυπο που εισάγεται στο σύστημα πρέπει να βρεθεί η κατηγορία στην οποία ανήκει. Αντίθετα, στη συσταδοποίηση, το πρόβλημα είναι η ομαδοποίηση ενός δοσμένου συνόλου προτύπων σε συστάδες. Υπό μια έννοια, πάλι έχουμε κατηγορίες προτύπων, οι κατηγορίες αυτές όμως προκύπτουν δυναμικά με βάση τα δεδομένα (είναι δηλαδή *data driven*), και δεν είναι καθορισμένες από πριν.

Οι τεχνικές συσταδοποίησης εφαρμόζονται συχνά σε προβλήματα ανάλυσης προτύπων, ομαδοποίησης, λήψης αποφάσεων, μηχανικής μάθησης , εξόρυξης δεδομένων (data mining) και άλλα. Με τον όρο «πρότυπο» εννοούμε ουσιαστικά ένα διάνυσμα **x** που αποτελείται από *d* μετρήσεις: $\mathbf{x}=(x_1,x_2,\ldots,x_d)$. Τα συστατικά x_i του διανύσματος αυτού καλούνται «χαρακτηριστικά» ή «ιδιότητες» του προτύπου. Με *d* συμβολίζουμε τη διάσταση του προτύπου ή του χώρου των προτύπων.

Μια τυπική διαδικασία συσταδοποίησης αποτελείται από τα παρακάτω βήματα:

1) Αναπαράσταση των προτύπων(επιλεκτικά μπορεί να περιέχει εξαγωγή χαρακτηριστικών, και /ή επιλογή).
2) Καθορισμός μιας μετρικής, ενδεικτικής της γειτνίασης των προτύπων, ανάλογα με τον τύπο των δεδομένων.

3) Συσταδοποίηση ή ομαδοποίηση.
4) Συνοπτική περιγραφή των δεδομένων (αν χρειαστεί).
5) Εγκυρότητα των συστάδων (αν χρειαστεί).

Είναι χαρακτηριστικό ότι η συσταδοποίηση είναι μία διαδικασία με επανατροφοδότηση: το αποτέλεσμα της διαδικασίας επανατροφοδοτείται στο σύστημα, το οποίο συνδυάζοντας το αποτέλεσμα αυτό με τις υπόλοιπες εισόδους, προχωράει στη εξαγωγή χαρακτηριστικών και στους υπολογισμούς των σχέσεων ομοιότητας, με στόχο την τελική εξαγωγή των συστάδων.

Η *αναπαράσταση των προτύπων* αναφέρεται στο πλήθος των κλάσεων, το πλήθος των διαθέσιμων προτύπων και το πλήθος, τύπο και κλίμακα των χαρακτηριστικών που είναι διαθέσιμα στο συγκεκριμένο αλγόριθμο συσταδοποίησης. Η *επιλογή των χαρακτηριστικών* είναι η ταυτοποίηση του υποσυνόλου εκείνου των αρχικών δεδομένων το οποίο μπορεί να οδηγήσει στην πιο αποτελεσματική συσταδοποίηση. Η *εξαγωγή των χαρακτηριστικών* αναφέρεται στο μετασχηματισμό υπαρχόντων χαρακτηριστικών για την παραγωγή προεχόντων, περισσότερο κατατοπιστικών όσον αφορά τη διαδικασία συσταδοποίησης. Ένα παράδειγμα αναπαράστασης προτύπου με βάση τα χαρακτηριστικά *βάρος* και *χρώμα* είναι (*20, μαύρο)*, για ένα πρότυπο χρώματος μαύρου που ζυγίζει 20 μονάδες βάρους.

Η *γειτνίαση των προτύπων* συνήθως μετριέται με βάση μια συνάρτηση απόστασης που ορίζεται για ζεύγη προτύπων. Η πιο απλή συνάρτηση απόστασης είναι η Ευκλείδεια. Η συνάρτηση απόστασης η οποία επιλέγεται, αποτελεί κάθε φορά το μέτρο της ομοιότητας μεταξύ των προτύπων. Με βάση το μέτρο αυτό γίνεται η καταχώρησή τους στην ίδια ή σε διαφορετικές συστάδες.

Το βήμα της *συσταδοποίησης* μπορεί να πραγματοποιηθεί με πολλούς τρόπους (υπάρχουν δηλαδή πολλοί αλγόριθμοι συσταδοποίησης). Το αποτέλεσμά του μπορεί να είναι είτε αυστηρό (μια ακριβής αντιστοίχηση των προτύπων σε ομάδες, έτσι ώστε κάθε πρότυπο να ανήκει αυστηρά σε μια μόνο ομάδα), είτε ασαφής (κάθε πρότυπο έχει μια μεταβλητή ένδειξη ιδιότητας μέλους για κάθε μια από τις συστάδες εξόδου).

Το επόμενο στάδιο είναι η *συνοπτική περιγραφή των δεδομένων*. Τα δεδομένα δηλαδή που ανήκουν στην ίδια συστάδα, περιγράφονται με ένα αντιπροσωπευτικό στοιχείο της συστάδας. Συνήθως επιλέγεται ο μέσος όρος των στοιχείων αυτής για να την εκπροσωπήσει. Για παράδειγμα, αν σε μια συστάδα υπάρχουν τα σημεία (1,2,3,4,5) μπορούν αυτά να αναπαρασταθούν με το μέσο όρο τους, το 3.

Η *εγκυρότητα των συστάδων* είναι η εκτίμηση της αξιοπιστίας του αποτελέσματος ενός αλγορίθμου συσταδοποίησης. Συνήθως η ανάλυση αυτή χρησιμοποιεί ένα συγκεκριμένο κριτήριο βελτιστοποίησης. Πέραν αυτών όμως, αυτά τα κριτήρια συνήθως επιλέγονται ανάλογα με το συγκεκριμένο πρόβλημα. Μια συσταδοποίηση είναι έγκυρη αν με βάση τη

λογική δεν μπορεί έχει προκύψει τυχαία ή να αποτελεί σφάλμα ενός αλγορίθμου συσταδοποίησης.

Σήμερα είναι διαθέσιμη μια πληθώρα αλγορίθμων συσταδοποίησης, και για την επιλογή του καταλληλότερου ανάμεσά τους βασιζόμαστε στα παρακάτω:

1) Τον τρόπο με τον οποίο οι συστάδες δημιουργούνται
2) Τη δομή των δεδομένων και
3) Την ευαισθησία της τεχνικής συσταδοποίησης σε αλλαγές που δεν πρέπει να επηρεάζουν τη δομή των δεδομένων.

Φυσικά, δεν υπάρχει κάποια τεχνική συσταδοποίησης η οποία να είναι αποτελεσματική σε όλες τις περιπτώσεις ομαδοποίησης των δομών που είναι παρούσες σε πολυδιάστατα σύνολα δεδομένων. Η επιλογή μεταξύ των αλγορίθμων συσταδοποίησης επαφίεται στο χρήστη, ο οποίος εκτός του ότι θα πρέπει να είναι γνώστης του αλγορίθμου, θα πρέπει να γνωρίζει καλά τις συνθήκες κάτω από τις οποίες συλλέχθηκαν τα δεδομένα καθώς και το γενικότερο πλαίσιο του προβλήματος.

Από την πληθώρα των αλγορίθμων συσταδοποίησης που χρησιμοποιούνται σήμερα, ο K Means («αλγόριθμος K-μέσων») και ο Single Link (τεχνική «μονός σύνδεσμος») έχουν χρησιμοποιηθεί στην παρούσα διπλωματική εργασία για την εξαγωγή των ομάδων πλοηγήσεων των χρηστών.

K-Means

Ο αλγόριθμος K-Means είναι ένας αλγόριθμος ταξινόμησης προτύπων σε K ομάδες, βασισμένος σε συγκεκριμένες ιδιότητες αυτών. Η συσταδοποίηση γίνεται ελαττώνοντας το άθροισμα των τετραγώνων των αποστάσεων ανάμεσα στο κάθε πρότυπο και στο κέντρο της συστάδας στην οποία αυτό ανήκει [Tek04]. Σύμφωνα με την πηγή [Τζα02], ο αλγόριθμος ανήκει στην ευρύτερη κατηγορία των αλγορίθμων μάθησης χωρίς επίβλεψη. Χρησιμοποιεί ένα δείγμα διανυσμάτων ιδιοτήτων (προτύπων) $S=\{\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_q}\}$ από έναν πληθυσμό P, αλλά απαιτεί να είναι εκ των προτέρων γνωστός ο αριθμός K των ομάδων K<P. Η διαδικασία ξεκινά θεωρώντας ότι τα πρώτα K διανύσματα (πρότυπα) $\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_q}$ είναι τα κέντρα $\mathbf{z_1},\mathbf{z_2},\ldots,\mathbf{z_q}$ των K ομάδων, έτσι ώστε η σειρά των δειγμάτων να είναι τυχαία. Ο αλγόριθμος στη συνέχεια κατανέμει κάθε ένα από τα εναπομένοντα Q-K διανύσματα στην ομάδα από το κέντρο της οποίας απέχει τη μικρότερη απόσταση. Η απόσταση αυτή, D, πρέπει ικανοποιεί τα παρακάτω αξιώματα:

1. $D(x,y) = 0$
2. $D(x,y) = D(y,x)$

3. D(x,y)<= D(x,z) + D(z,y)

Πιο σύνηθες είναι να επιλέγεται η Ευκλείδεια απόσταση. Τότε, από τα πρότυπα κάθε ομάδας k, βρίσκεται ο μέσος όρος τους για να προσδιοριστεί ένα νέο, ανανεωμένο κέντρο $\mathbf{z_k}^*$ αυτής. Στη συνέχεια, κάθε ένα από τα Q πρότυπα που εξετάστηκαν ταξινομείται και πάλι στην ομάδα από το κέντρο της οποίας απέχει το λιγότερο. Κατόπι βρίσκονται ξανά τα νέα κέντρα με υπολογισμό των μέσων τιμών των προτύπων κάθε ομάδας και κατανομή εκ νέου των Q προτύπων στις K ομάδες με το κριτήριο της ελάχιστης απόστασης από τα νέα αυτά κέντρα. Η διαδικασία αυτή επαναλαμβάνεται μέχρις ότου καμία ομάδα να μην αλλάζει περαιτέρω, όπότε ο αλγόριθμος τερματίζει. Το μέσο τετραγωνικό σφάλμα σε κάθε σταθερή κατηγορία k είναι

$(\sigma_k)^2=\frac{1}{n(k)} \sum_{q:κατηγ.(q)=k} ||\mathbf{x}_q-\mathbf{z}_k||_2^2$, όπου n(k) είναι ο τρέχων αριθμός των δειγματικών προτύπων που αποδίδονται στην k κατηγορία. Το ολικό μέσο τετραγωνικό σφάλμα είναι

$(\sigma_{ολικό})^2=\sum_{k=1}^{K} (\sigma_k)^2$, και αποτελεί ένα μέτρο συνολικής συσταδοποίησης το οποίο πρέπει να ελαχιστοποιηθεί ως προς τον αριθμό K των ομάδων.

Για την εκχώρηση των προτύπων σε ομάδες, τροφοδοτείται από μία ακολουθία Q προτύπων ένα πρότυπο τη φορά $\mathbf{x}_q$, $1\leq q\leq Q$, και καταχωρείται σε μία από τις K ομάδες που έχουν επιλεχθεί ως αρχικές. Η εκχώρηση αυτή καταγράφεται με μια συνάρτηση δείκτη εκχώρησης κατηγορίας : «κατηγορία(q)=k», που δηλώνει ότι το $\mathbf{x}_q$ καταχωρήθηκε στην κατηγορία k.

Ο Αλγόριθμος των K μέσων

Είσοδοι: Τυχαία διατεταγμένα δείγματα διανυσμάτων (προτύπων) $S=\{\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_q}\}$ και πλήθος K των ομάδων.

Έξοδοι: -K ομάδες $C_1,C_2,\ldots C_K$ οριζόμενες με το δείκτη απόδοσης κατηγορίας «κατηγορία(q)=k»

- Αριθμός διανυσμάτων n(k) σε κάθε κατηγορία k.
- Κέντρο $\mathbf{z_k}$ της κάθε ομάδας.
- Το μέσο τετραγωνικό σφάλμα $(\sigma_k)^2$ της κάθε ομάδας k.
- Το ολικό μέσο τετραγωνικό σφάλμα, $(\sigma_{ολικό})^2$.

Βήματα του αλγορίθμου

Τα βήματα του αλγορίθμου περιγράφονται συνοπτικά

Βήμα 1:

Θέσε τα πρώτα K δειγματικά διανύσματα ως αρχικά κέντρα για k=1,2,…,K.

Θέσε $\mathbf{z_k} \leftarrow \mathbf{x_k}$, n(k)=0 /*Αρχικοποίηση των κέντρων των ομάδων*/

Βήμα 2:

Ταξινόμησε κάθε δειγματικό διάνυσμα στην ομάδα με το πλησιέστερο κέντρο.

Το βήμα αυτό περιλαμβάνει αρχικοποίηση της ελάχιστης απόστασης d_{min} σε έναν πολύ μεγάλο αριθμό. Εάν το πρότυπο $\mathbf{x}_q$ απέχει από το $\mathbf{z_k}$ απόσταση μικρότερη από d_{min}, τότε:

-Θέσε την απόσταση των $\mathbf{x}_q$ και $\mathbf{z_k}$ σαν τη νέα d_{min}

-k_{min}=k(Εύρεση ελάχιστης απόστασης από ένα κέντρο)

-κλάση(q)=k_{min}

-n(k_{min})= n(k_{min})+1

Βήμα 3:

Υπολόγισε τη νέα μέση τιμή και θέσε την ως το νέο κέντρο για την κάθε κατηγορία C_k.

Για k=1,2,…,K

- $\mathbf{z_k}^* \leftarrow \frac{1}{n(k)} \sum_{q:κατηγ.(q)=k} \mathbf{x}_q$ (Υπολογισμός K νέων κέντρων/ μέση τιμή για την k ομάδα)

-Υπολόγισε τα $(\sigma_k)^2$ και $(\sigma_{ολικό})^2$.

Βήμα 4:

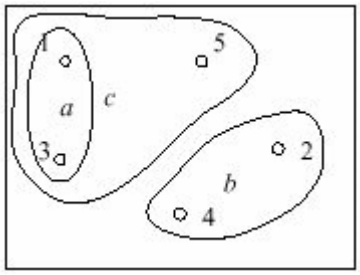
Αν κάποιο κέντρο έχει αλλάξει, τότε προχώρησε σε επόμενο κύκλο ομαδοποίησης. Διαφορετικά τερμάτισε τον αλγόριθμο.

Ένα παράδειγμα εφαρμογής του αλγορίθμου αυτού, δίνεται αμέσως παρακάτω:

<u>Παράδειγμα</u>

Όπως φαίνεται στο σχήμα, έχουμε 5 σημεία στο χώρο 1,2,3,4,5 και θέλουμε να τα κατατάξουμε σε k=2 συστάδες. Αρχικά επιλέγουμε τα σημεία 1,2 ως δυο κέντρα δυο διαφορετικών συστάδων .Ακολούθως κάνουμε τα εξής:

1. Εισαγωγή του 3 στη συστάδα με κέντρο το 1 μιας είναι πιο κοντά από το 2. Προσδιορισμός νέου κέντρου για τη συστάδα αυτή, το σημείο α.
2. Εισαγωγή του 4 στη συστάδα με κέντρο το 2 μιας και είναι πιο κοντά από το α. Προσδιορισμός νέου κέντρου για τη συστάδα αυτή, το σημείο b.
3. Εισαγωγή του 5 στη συστάδα με κέντρο το α, μιας και είναι πιο κοντά από το b. Προσδιορισμός νέου κέντρου για τη συστάδα αυτή, το σημείο c.



Σχήμα 3.2:Παράδειγμα K means αλγορίθμου

Ο αλγόριθμος επαναλαμβάνεται κατά τον ίδιο τρόπο μέχρι τα κέντρα να μην αλλάζουν πια.

Single Link

Ο αλγόριθμος αυτός ανήκει στη γενικότερη κατηγορία των ιεραρχικών αλγορίθμων (hierarchical clustering), δημιουργεί δηλαδή μια ιεραρχική αποσύνθεση του συνόλου δεδομένων βασιζόμενος σε κάποιο κριτήριο [Yeu05]. Συνήθως το κριτήριο αυτό είναι οι αποστάσεις μεταξύ των προτύπων, οι οποίες υποτίθεται ότι είναι καταχωρημένες σε έναν πίνακα αποστάσεων. Οι ιεραρχικές τεχνικές δεν απαιτούν σαν είσοδο τον αριθμό των συστάδων, k, αλλά απαιτούν μια συνθήκη τερματισμού, διαφορετικά θα εντάξουν όλα τα πρότυπα στην ίδια ομάδα. Στις τεχνικές αυτές, υπάρχουν πολλά επίπεδα, σε κάθε ένα από τα οποία δημιουργούνται σύνολα από συστάδες. Υπάρχουν δύο τρόποι εύρεσης των συστάδων με τους αλγορίθμους αυτούς: οι συσσωρευτικές (“agglomerative clustering”) και οι διαχωριστικές (“divisive clustering”). Η τεχνική «μονός σύνδεσμος» συνήθως υλοποιείται με συσσωρευτικές μεθόδους, γι’ αυτό και θα αναλυθούν μόνο αυτές στο παρόν έγγραφο.

Η γενική ιδέα των συσσωρευτικών αλγορίθμων έχει ως εξής:

-Αρχικά, κάθε πρότυπο αποτελεί από μόνο του μία συστάδα.

-Επαναληπτικά οι συστάδες ενώνονται μεταξύ τους.

-Είναι “bottom-up”, ξεκινούν δηλαδή από κάτω (τόσες ομάδες όσα και πρότυπα) και πηγαίνουν προς την κορυφή (τελικά καταλήγουμε σε μια μόνο συστάδα).

Οι ιεραρχικές τεχνικές οπτικοποιούνται με ένα δενδρόγραμμα, στο κάθε επίπεδο του οποίου βρίσκονται οι συστάδες αυτού του επιπέδου και στη ρίζα βρίσκεται μόνο μία συστάδα που περιέχει όλα τα πρότυπα.

Ο αλγόριθμος για την τεχνική «μονός σύνδεσμος», με χρήση συσσωρευτικής συσταδοποίησης, δίνεται αμέσως παρακάτω [Bor94]:

Αλγόριθμος:

Είσοδοι:Ένα σύνολο N στοιχείων(πρότυπα) και ένας πίνακας απόστασης ή ομοιότητας A.

Έξοδος:Ενα δενδρόγραμμα το οποίο αντιπροσωπεύεται ως ένα σύνολο από διατεταγμένες τριπλέτες.

Βήμα 1:

Ξεκίνα τοποθετώντας κάθε στοιχείο σε διαφορετική συστάδα. Έτσι, αν έχεις Ν αντικείμενα σαν είσοδο, στο βήμα αυτό έχεις Ν συστάδες. Θέσε τις αποστάσεις (ομοιότητες) μεταξύ των συστάδων ίσες με τις αποστάσεις (ομοιότητες) μεταξύ των στοιχείων που περιέχουν.

Βήμα 2:

Βρες το πιο κοντινό (πιο όμοιο) ζευγάρι συστάδων και ένωσε τις συστάδες αυτές σε μία. Τοποθέτησε τις νέες συστάδες στο επόμενο επίπεδο του δενδρογράμματος.

Βήμα 3:

Υπολόγισε τις αποστάσεις (ομοιότητες) μεταξύ των νέων συστάδων και κάθε μιας από τις παλιές συστάδες. Στην τεχνική «μονός σύνδεσμος», θεωρείται ότι η απόσταση μεταξύ δύο συστάδων είναι ίση με τη μικρότερη εκ των αποστάσεων μεταξύ όλων των μελών της μιας συστάδας από όλα τα μέλη της άλλης συστάδας. Αν στα δεδομένα υπάρχει πίνακας ομοιότητας, θεωρείται ότι η ομοιότητα μεταξύ δύο συστάδων είναι η μεγαλύτερη ομοιότητα μεταξύ όλων των μελών των δύο συστάδων ανά δύο.

Βήμα 4:

Επανέλαβε τα βήματα 2 και 3 μέχρι όλα τα πρότυπα να έχουν ομαδοποιηθεί σε μια μόνο συστάδα.

Το πλεονέκτημα της τεχνικής αυτής είναι ότι είναι πολύ γρήγορη. Αντίθετα, υστερεί έναντι άλλων μεθόδων συσταδοποίησης στο ότι δημιουργεί πιθανώς μεγάλες και αδύναμες συστάδες.

Ένα παράδειγμα όπου φαίνεται η υλοποίηση του αλγορίθμου αυτού είναι το παρακάτω:

<u>Παράδειγμα</u>

Έστω ότι δίνονται 5 πρότυπα και ο πίνακας των μεταξύ τους αποστάσεων είναι ο

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & & & & \\ 2 & 0 & & & \\ 6 & 3 & 0 & & \\ 10 & 9 & 7 & 0 & \\ 9 & 8 & 5 & 4 & 0 \end{bmatrix} \end{array}$$

Αρχικά τα πέντε πρότυπα είναι τοποθετημένα το καθένα σε διαφορετική συστάδα. Η μικρότερη απόσταση μεταξύ όλων των αποστάσεων των στοιχείων ανά δύο είναι 2 και αντιστοιχεί στα πρότυπα 1 και 2. Έτσι, τα πρότυπα αυτά συνενώνονται σε μία συστάδα και πλέον υπολογίζονται οι αποστάσεις της συστάδας αυτής από όλες τις υπόλοιπες.

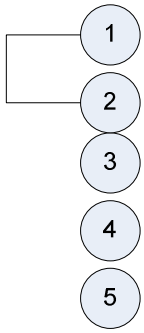
Ο υπολογισμός των νέων αυτών αποστάσεων, το δενδρόγραμμα στο επίπεδο αυτό, καθώς και ο νέος πίνακας που προκύπτει, δίνονται παρακάτω.

$d_{(1,2),3}=\min\{d_{1,3}, d_{2,3}\}=\min\{6,3\}=3$

$d_{(1,2),4}=\min\{d_{1,4}, d_{2,4}\}=\min\{10,9\}=9$

$d_{(1,2),5}=\min\{d_{1,5}, d_{2,5}\}=\min\{9,8\}=8$

	(1,2)	3	4	5
(1,2)	0			
3	3	0		
4	9	7	0	
5	8	5	4	0



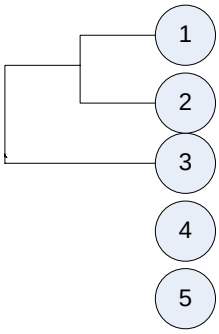
Σχήμα 3.3: Δενδρόγραμμα στο 1^{ο} επίπεδο

Στη συνέχεια συνενώνονται οι συστάδες με τα στοιχεία (1,2) και 3 σε μία συστάδα, γιατί αυτές έχουν τώρα τη μικρότερη απόσταση, ίση με 3. Οι υπολογισμοί των νέων αποστάσεων, ο νέος πίνακας και το νέο δενδρόγραμμα δίνονται παρακάτω:

$d_{(1,2,3),4}=\min\{d_{(1,2),4}, d_{3,4}\}=\min\{9,7\}=7$

$d_{(1,2,3),5}=\min\{d_{(1,2),5}, d_{3,5}\}=\min\{8,5\}=5$

	(1,2)	3	4	5
(1,2)	0			
3	3	0		
4	9	7	0	
5	8	5	4	0



Σχήμα 3.4: Δενδρόγραμμα στο 2^{ο} επίπεδο

Η διαδικασία συνεχίζεται μέχρι όλα τα στοιχεία να ομαδοποιηθούν σε μια συστάδα

3.4.2 Αλγόριθμος C-Index

Ο αλγόριθμος C-Index είναι ένας αλγόριθμος ο οποίος χρησιμοποιείται από την τεχνική «μονός σύνδεσμος» για η βελτιστοποίηση του αποτελέσματός της. Πιο συγκεκριμένα, η τεχνική «μονός σύνδεσμος», δέχεται σαν είσοδο ένα επίπεδο συσταδοποίησης, το οποίο δεν είναι τίποτε άλλο παρά ένας αριθμός ενδεικτικός του πότε σταματά η εκτέλεση του αλγορίθμου. Σημαίνει δηλαδή το σημείο στο οποίο ο αλγόριθμος θα σταματήσει τη συγχώνευση των συστάδων και την ανάπτυξη του δενδρογράμματος προς τα πάνω (υποτίθεται bottom-up υλοποίηση). Ο αλγόριθμος C-Index υπολογίζει το επίπεδο συσταδοποίησης για το οποίο η προκύπτουσα συσταδοποίηση είναι η βέλτιστη δυνατή. Έτσι, αποφεύγονται οι δοκιμές διαφόρων τιμών για το επίπεδο συσταδοποίησης, οι οποίες μπορεί τελικά και να μην καταλήξουν στην επιλογή του καταλληλότερου επιπέδου.

Η βάση για την ανάπτυξη του αλγορίθμου αυτού υπήρξε η αναφορά [DCS+]. Ο αλγόριθμος αυτός υπολογίζει τη βέλτιστη τιμή για το επίπεδο συσταδοποίησης (clustering level) που χρησιμοποιείται από την τεχνική «μονός σύνδεσμος» για τον προσδιορισμό των συστάδων.

Ως C-Index ορίζεται το διάνυσμα των ζευγαριών $((i_1,n_1),(i_2,n_2),(i_3,n_3),\ldots,(i_p,n_p))$. Τα ζεύγη αυτά είναι τόσα όσα είναι τα επίπεδα συσταδοποίησης $l_1,l_2,l_3,\ldots,l_p$ τα οποία εξετάζει ο αλγόριθμος. Ακόμα,

$\underline{i_i}$ είναι οι τιμή δείκτη που υπολογίζει η C-Index για το i-οστό επίπεδο συσταδοποίησης.

$\underline{n_i}$ είναι ο αριθμός των συστάδων σε κάθε επίπεδο συσταδοποίησης.

Για κάθε επίπεδο συσταδοποίησης l_i παράγονται N_i clusters (δηλαδή $n_i=N_i$). Τα clusters αυτά είναι τα $\{C_1,C_2,\ldots,C_{Ni}\}$,με αντίστοιχο πλήθος στοιχείων $\{c_1,c_2,\ldots,c_{Ni}\}$. Ορίζονται επίσης τα παρακάτω μεγέθη-μεταβλητές:

$\underline{n_d}=c_1*(c_1-1)/2+c_2*(c_2-1)/2+\ldots+c_{N1*}(c_{N1}-1)/2$, δηλαδή το πλήθος των αποστάσεων (ανά δύο στοιχεία) ανά συστάδα.

d_w= Sum(d_{w1}) + Sum(d_{w2}) +…+ Sum(d_{wN1}), όπου Sum(d_w) είναι το άθροισμα των αποστάσεων (ανά δύο στοιχεία) όλων των στοιχείων της συστάδας C_i, 1<=i<=n_1.

max (d_w) : το άθροισμα των n_d μεγαλύτερων αποστάσεων (ανά δύο στοιχεία) σε όλο το σύνολο των στοιχείων.

min (d_w): το άθροισμα των n_d μικρότερων αποστάσεων (ανά δύο στοιχεία) σε όλο το σύνολο των στοιχείων.

Όμοια υπολογίζονται όλες οι άλλες τιμές δείκτη του C-Index για τα υπόλοιπα p επίπεδα συσταδοποίησης, δημιουργώντας το διάνυσμα $((i_1,n_1),(i_2,n_2),(i_3,n_3),\ldots,(i_p,n_p))$. Το καταλληλότερο επίπεδο συσταδοποίησης είναι αυτό που δίνει αριθμό από συστάδες που αντιστοιχεί στη χαμηλότερη τιμή δείκτη στο διάνυσμα C-Index.

Ο αλγόριθμος έχει την εξής μορφή:

Είσοδος:Ο πίνακας A, ο οποίος είναι ο πίνακας των similarities μεταξύ όλων των paths της Βάσης Δεδομένων. Οι τιμές του είναι τιμές double.

Έξοδος:Ο δεκαδικός αριθμός μεταξύ 0 και 1 που αντιστοιχεί στο καλύτερο επίπεδο συσταδοποίησης για τον αλγόριθμο Single-Link.

Αλγόριθμος:

-Η double μεταβλητή step αντιστοιχεί στο επίπεδο συσταδοποίησης που εξετάζει ο αλγόριθμος σε κάθε επανάληψη. Αρχικοποιείται στην τιμή 0.01 και σε κάθε επανάληψη αυξάνεται κατά 0.01.

-Ο Vector Clusters1 περιέχει τα clusters που προκύπτουν μετά από την εφαρμογή του αλγορίθμου Single-Link για το κάθε επίπεδο συσταδοποίησης. Πιο συγκεκριμένα, είναι ένας vector από vectors, του οποίου στοιχεία είναι τα clusters που αντιστοιχούν στο συγκεκριμένο επίπεδο συσταδοποίησης.

-Ο Vector indexVec αποτελείται από δομές τύπου Index: Οι δομές αυτές έχουν δύο double πεδία. Το πρώτο αντιστοιχεί στην τιμή δείκτη για κάποιο επίπεδο συσταδοποίησης και το δεύτερο αντιστοιχεί στο επίπεδο αυτό.

-Οι μεταβλητές n_d, max(d_w), min(d_w) και Sum, είναι αυτές που ορίστηκαν παραπάνω και αρχικοποιούνται στην τιμή 0.

Ακολουθεί η περιγραφή του αλγορίθμου:

Όσο το επίπεδο συσταδοποίησης είναι μικρότερο του 1 (δηλαδή step<1){

Βρες τα clusters που αντιστοιχούν στο συγκεκριμένο επίπεδο συσταδοποίησης και αποθήκευσέ τα στο Vector Clusters1.

Για κάθε στοιχείο του Vector Clusters1(δηλαδή για κάθε cluster){

Βρες το μέγεθός του και αποθήκευσέ το στη μεταβλητή n_d.

}

Υπολόγισε τα min(d_w) και max(d_w) για τα n_d στοιχεία.

Για όλα τα clusters του τρέχοντος επιπέδου{

Υπολόγισε το Sum για το τρέχον επίπεδο

}

Αν max(d_w) είναι διάφορο του min(d_w) {

/*αποφεύγεται η περίπτωση διαίρεσης με το 0*/

Υπολογισμός του double ii, που είναι η τιμή δείκτη για το τρέχον επίπεδο συσταδοποίησης.

Τοποθέτηση της τιμής αυτής και του επιπέδου στο οποίο αντιστοιχεί στο Vector indexVec.

}

}

Εφαρμογή του αλγορίθμου Bubblesort στο vector indexVec, με βάση την τιμή δείκτη.

Επιστροφή της μικρότερης τιμής δείκτη σαν αποτέλεσμα του αλγορίθμου.

4

Ανάλυση και Σχεδίαση

Στο κεφάλαιο αυτό παρουσιάζεται η αναλυτική μελέτη του συστήματός NaviMoz. Αρχικά γίνεται η ανάλυση των λειτουργικών απαιτήσεων από το σύστημα με το διαχωρισμό του σε επιμέρους υποσυστήματα. Στη συνέχεια παρουσιάζονται αναλυτικά οι εφαρμογές που υλοποιούν το σύστημα.

4.1 Ανάλυση-Περιγραφή Αρχιτεκτονικής

Η ενότητα αυτή στόχο έχει να δώσει μια γενική εικόνα του τρόπου οργάνωσης του συστήματος με το διαχωρισμό του σε υποσυστήματα και παρουσίαση της αρχιτεκτονικής τους.

4.1.1 Διαχωρισμός υποσυστημάτων

Το σύστημα απαρτίζεται ουσιαστικά από τρία υποσυστήματα, το υποσύστημα του χρήστη, το υποσύστημα του διαχειριστή και το υποσύστημα της Βάσης Δεδομένων. Καθένα από τα δύο πρώτα υποσυστήματα αποτελείται από άλλα υποσυστήματα. Πιο συγκεκριμένα:

1) Υποσύστημα χρήστη
    1.1)Υποσύστημα εισόδου υπάρχοντος χρήστη στο σύστημα.
    1.2)Υποσύστημα δημιουργίας νέου χρήστη και εισαγωγή του στο σύστημα.
    1.3)Υποσύστημα διαπροσωπείας χρήστη.
2) Υποσύστημα διαχειριστή
    2.1) Υποσύστημα ερωτήσεων ταυτοποίησης χρηστών.
    2.2) Υποσύστημα ερωτήσεων εξόρυξης δεδομένων.
    2.3) Υποσύστημα ερωτήσεων ομαδοποίησης πλοηγήσεων και χρηστών.
    2.4) Υποσύστημα διαπροσωπείας διαχειριστή.

3) Υποσύστημα διαχείρισης Βάσης Δεδομένων. Αυτό είναι υπεύθυνο για τη σύνδεση με τη βάση, την αποθήκευση σε αυτήν καταλλήλων πληροφοριών και την άντληση πληροφοριών από αυτή.

Όπως θα εξηγηθεί και στη συνέχεια, μέσα από τις διαπροσωπείες, ο εκάστοτε χρήστης του συστήματος θα μπορεί να επιλέξει τη λειτουργία που θέλει να επιτελέσει και να εισάγει τα διάφορα δεδομένα, όπως για παράδειγμα το όνομά του (υποσύστημα χρήστη) ή κάποιο πρότυπο του οποίου θέλει να βρει τα όμοια (υποσύστημα διαχειριστή).

Στο υποσύστημα διαπροσωπείας χρήστη (1.3), οι δυνατές λειτουργίες που μπορεί να επιλέξει ο χρήστης του συστήματος είναι οι εξής:

- Εισαγωγή των στοιχείων Username και Password, αν πρόκειται για εγγεγραμμένο χρήστη του συστήματος.
- Εισαγωγή των προσωπικών του στοιχείων και εγγραφή του στο σύστημα αν πρόκειται για νέο χρήστη του συστήματος.

Στο υποσύστημα διαπροσωπείας διαχειριστή (2.4), οι δυνατές λειτουργίες που μπορεί να επιλέξει ο διαχειριστής του συστήματος είναι οι εξής:

- Εμφάνιση των πλοηγήσεων ενός χρήστη που επιλέγεται από λίστα.
- Εμφάνιση της διάρκειας των πλοηγήσεων ενός χρήστη που επιλέγεται από λίστα, μέσω της εμφάνισης των στιγμών εισόδου και εξόδου του χρήστη στο σύστημα.
- Εύρεση των πλοηγήσεων που αποτελούν υπερσύνολο μιας δοσμένης πλοήγησης.
- Εύρεση των πλοηγήσεων που είναι ταυτόσημες με μια δοσμένη πλοήγηση.
- Εύρεση των πλοηγήσεων που είναι κατά κάποιο βαθμό όμοιες με μια δοσμένη πλοήγηση.
- Εύρεση των πιο δημοφιλών πλοηγήσεων.
- Εύρεση των συστάδων των πλοηγήσεων και των χρηστών με τη μέθοδο *των Κ μέσων* (K Means)
- Εύρεση των συστάδων των πλοηγήσεων και των χρηστών με την τεχνική *μονός σύνδεσμος* (Single Link)
- Εύρεση των περισσότερο αναποφάσιστων χρηστών του συστήματος.

Σημειώνεται ότι σε όλες τις παραπάνω εργασίες, επιλέγεται και χρονικό διάστημα, οι πλοηγήσεις που πραγματοποιούνται εντός του οποίου εξετάζονται.

Το υποσύστημα χρήστη αντιστοιχεί στις λειτουργίες που επιτελούν οι χρήστες του NaviMoz οι οποίοι πραγματοποιούν τις πλοηγήσεις στην ιεραρχία του dmoz. Οι πλοηγήσεις τους αυτές αποθηκεύονται στη Βάση Δεδομένων και αποτελούν αντικείμενο επεξεργασίας από το

διαχειριστή του συστήματος. Το υποσύστημα διαχειριστή, αντιστοιχεί στις διεργασίες-ερωτήσεις εκείνες που εκτελεί ο διαχειριστής του συστήματος προκειμένου να ανακτήσει τις πληροφορίες που τον ενδιαφέρουν για τους χρήστες. Πιο συγκεκριμένα, το υποσύστημα ταυτοποίησης των χρηστών θέτει απλές ερωτήσεις για τους χρήστες. Το υποσύστημα ερωτήσεων εξόρυξης δεδομένων αναζητά πληροφορίες που σχετίζονται με κάποια συγκεκριμένη πλοήγηση. Τέλος, το υποσύστημα ερωτήσεων ομαδοποίησης πλοηγήσεων και χρηστών εκτελεί γενικές ομαδοποιήσεις χωρίς να βασίζεται σε κάποια συγκεκριμένη πλοήγηση. Το υποσύστημα χρήστη δεν μπορεί να επικοινωνήσει με το υποσύστημα του διαχειριστή, ο χρήστης δηλαδή δεν έχει γνώση των ενεργειών του διαχειριστή. Οι πλοηγήσεις όμως των χρηστών αποθηκεύονται στη Βάση Δεδομένων και μέσω αυτής μπορεί ο διαχειριστής να έχει πρόσβαση σε αυτές. Η πορεία αυτή (χρήστης→Βάση Δεδομένων→διαχειριστής) δεν είναι αμφίδρομή. Ο διαχειριστής απλά ανακτά δεδομένα από τη Βάση, χωρίς να αποθηκεύει τίποτα σε αυτή.

Στο σχήμα 4.1 παρουσιάζεται η αρχιτεκτονική του συστήματος.

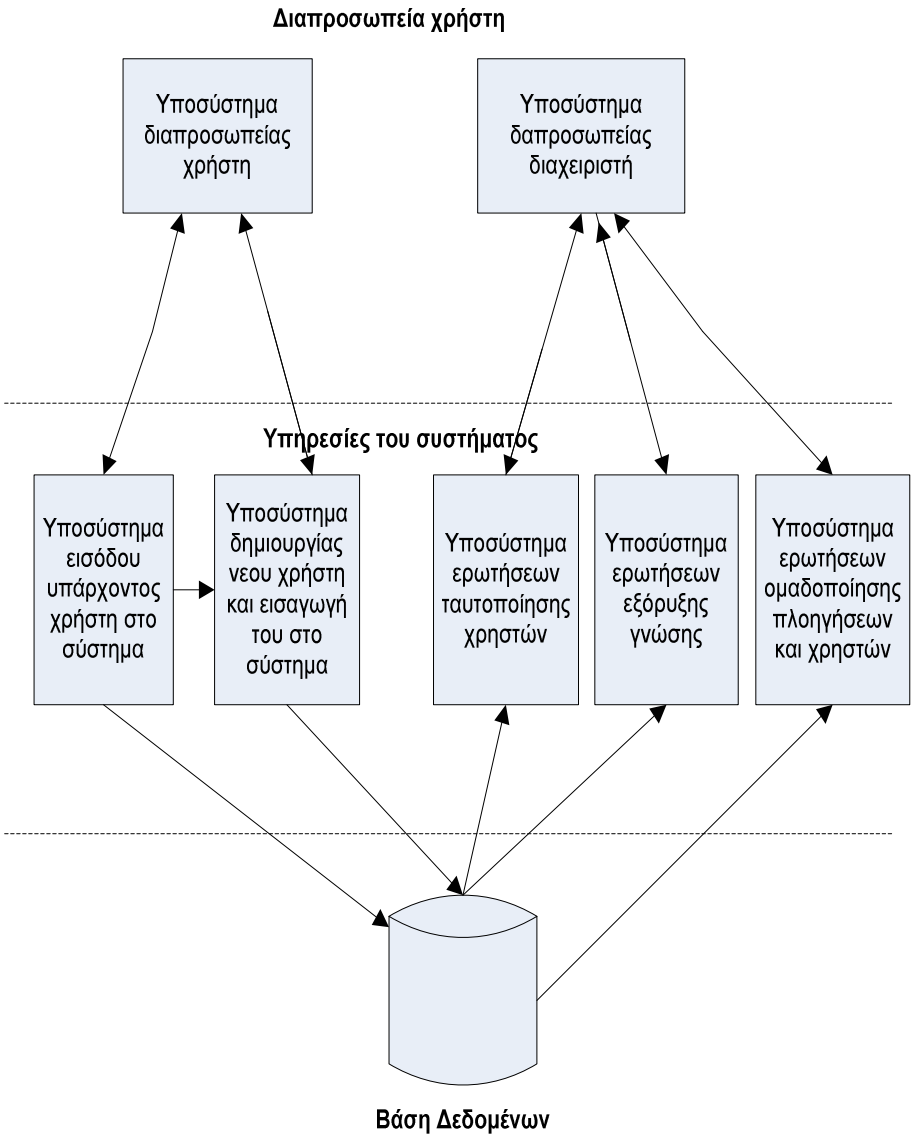
Όπως φαίνεται και στο σχήμα, έχουμε τρία στρώματα: Το πρώτο είναι η Βάση Δεδομένων, το μεσαίο είναι το στρώμα των υπηρεσιών του συστήματος το οποίο είναι αυτό που επικοινωνεί άμεσα με το στρώμα της διαπροσωπείας. Μέσω του μεσαίου στρώματος γίνεται η σύνδεση του τρίτου με το πρώτο επίπεδο, μέσα από κατάλληλες ερωτήσεις. Το τρίτο είναι το στρώμα διαπροσωπείας, το οποίο στέλνει τα αιτήματά του στο δεύτερο.

4.1.2 Περιγραφή υποσυστημάτων

Στην παράγραφο αυτή παρουσιάζονται αναλυτικά τα διάφορα υποσυστήματα και οι λειτουργίες που πρέπει να επιτελούν.

4.1.2.1 Υποσύστημα εισόδου υπάρχοντος χρήστη στο σύστημα

Το υποσύστημα αυτό πρέπει να είναι υπεύθυνο για την είσοδο ενός ήδη εγγεγραμμένου χρήστη του συστήματος σε αυτό. Όταν ο χρήστης δώσει τα αναγνωριστικά του στοιχεία (username και password), με κατάλληλο τρόπο θα εξετάζεται αν αντιστοιχούν σε εγγεγραμμένο στο σύστημα χρήστη. Αν ναι, θα εισάγεται στο NaviMoz και θα αρχίζει την πλοήγησή του στην ιεραρχία του dmoz. Επίσης πρέπει να προσφέρεται και η εναλλακτική, αν δεν είναι γραμμένος χρήστης, να ανοίξει τη φόρμα εγγραφής και να καταχωρηθεί στο σύστημα.



Σχήμα 4.1 Αρχιτεκτονική του συστήματος

4.1.2.2 Υποσύστημα δημιουργίας νέου χρήστη και εισαγωγή του στο σύστημα

Το υποσύστημα αυτό είναι άμεσα εξαρτώμενο από το υποσύστημα εισόδου υπάρχοντος χρήστη στο σύστημα. Πρέπει να ενημερώνει τη Βάση Δεδομένων με τα στοιχεία του νέου χρήστη και στη συνέχεια να τον εισάγει στο σύστημα, επιτρέποντάς του να πλοηγηθεί στην ιεραρχία του dmoz. Επίσης θα εξετάζει αν ο χρήστης δίνει όλα τα στοιχεία που του ζητούνται και θα είναι και υπεύθυνο για τον έλεγχο των στοιχείων του έτσι ώστε να μην υπάρχουν δύο ίδιοι χρήστες του συστήματος.

4.1.2.3 Υποσύστημα διαπροσωπείας χρήστη

Το υποσύστημα αυτό αποτελεί τη διεπαφή του συστήματος με το χρήστη. Το υποσύστημα αυτό πρέπει να είναι εύχρηστο και να καθοδηγεί με διάφορες επεξηγήσεις το χρήστη για τη σωστή συμπλήρωση των διαφόρων στοιχείων. Επίσης, σε περίπτωση λάθους ο χρήστης πρέπει να λαμβάνει κατάλληλα επεξηγηματικά μηνύματα. Οι λειτουργίες που θα μπορεί να επιτελέσει ο χρήστης μέσω του υποσυστήματος αυτού είναι οι:

- Εισαγωγή των στοιχείων Username και Password, αν πρόκειται για εγγεγραμμένο χρήστη του συστήματος.
- Εισαγωγή των προσωπικών του στοιχείων και εγγραφή του στο σύστημα αν πρόκειται για νέο χρήστη του συστήματος.

4.1.2.4 Υποσύστημα ερωτήσεων ταυτοποίησης χρηστών

Το υποσύστημα αυτό αντιστοιχεί στο υποσύστημα του διαχειριστή και εξυπηρετεί μία κατηγορία ενεργειών του. Οι ενέργειες αυτές είναι απλές ερωτήσεις ταυτοποίησης χρηστών. Θα αναζητούνται δηλαδή πληροφορίες που σχετίζονται με έναν χρήστη του συστήματος, όπως για παράδειγμα οι πλοηγήσεις του εντός ενός συγκεκριμένου χρονικού διαστήματος. Ο διαχειριστής δηλαδή θα μπορεί να επιλέγει ένα χρήστη του συστήματος, έστω το Μακρή Μανόλη, και να ζητά γι' αυτόν τις πλοηγήσεις που έχει πραγματοποιήσει μέσα σε μια συγκεκριμένη χρονική περίοδο. Τα αποτελέσματα θα πρέπει να επιστρέφονται σε κατανοητή μορφή στην αρχική φόρμα του συστήματος και να είναι τα /Arts/Music/Pop, /Arts/Radio/People/Young.

4.1.2.5 Υποσύστημα ερωτήσεων εξόρυξης δεδομένων

Το υποσύστημα αυτό εξυπηρετεί μια άλλη κατηγορία ενεργειών του διαχειριστή που σχετίζονται με κάποια συγκεκριμένη πλοήγηση. Ο διαχειριστής θα δίνει μία συγκεκριμένη πλοήγηση, η οποία θα αποτελεί το μοντέλο με βάση το οποίο εξετάζονται όλες οι υπόλοιπες πλοηγήσεις. Πρέπει δηλαδή να εξετάζονται οι ομοιότητες και οι διαφορές των υπολοίπων πλοηγήσεων με το μοντέλο αυτό, όπως για παράδειγμα η κατά ένα ποσοστό ομοιότητά τους, για συγκεκριμένο χρονικό διάστημα (εξόρυξη δεδομένων). Αν δηλαδή ο χρήστης δίνει την πλοήγηση /Sports/Olympics και ένα ποσοστό, έστω 60%, το σύστημα θα πρέπει να είναι σε θέση να του επιστρέψει τις πλοηγήσεις που είναι κατά 60% όμοιες με τη δοσμένη, με βάση το μέτρο της δομικής απόστασης, τροποποιημένης έτσι ώστε να λαμβάνει υπ' όψη της και το περιεχόμενο των συμβολοακολουθιών. Στο δεδομένο παράδειγμα, το σύστημα θα πρέπει να είναι σε θέση να επιστρέψει τις πλοηγήσεις /Sports/Events/Olympics/2004_-_Athens και /Sports/Events/Olympics/Paralympics

4.1.2.6 Υποσύστημα ερωτήσεων ομαδοποίησης δεδομένων και χρηστών

Το υποσύστημα αυτό είναι ίσως το σημαντικότερο υποσύστημα του συστήματος, από τη στιγμή που επιτελεί γενικές ομαδοποιήσεις χρηστών. Η σημαντική διαφορά του από το προηγούμενο υποσύστημα έγκειται στο ότι δε θα πρέπει να εξετάζει τις πλοηγήσεις με βάση κάποια συγκεκριμένη πλοήγηση-πρότυπο. Αντίθετα, θα εξετάζει όλες τις πλοηγήσεις μεταξύ τους και θα πραγματοποιεί ερωτήσεις πάνω σε αυτές προκειμένου να εξάγει κάποια γενικά στοιχεία ομαδοποίησης των χρηστών κατά τη διάρκεια μιας συγκεκριμένης χρονικής περιόδου. Ένα παράδειγμα εργασιών που πρέπει να επιτελεί το υποσύστημα αυτό είναι η συσταδοποίηση (clustering) των χρηστών με βάση το πόσο όμοιες πλοηγήσεις έχουν πραγματοποιήσει μεταξύ τους. Έτσι λοιπόν οι χρήστες που έχουν πραγματοποιήσει πλοηγήσεις στον κατάλογο Sports του dmoz θα πρέπει να καταχωρούνται σε διαφορετική ομάδα από αυτούς που έχουν πραγματοποιήσει πλοηγήσεις στον κατάλογο Arts.

4.1.2.7 Υποσύστημα διαπροσωπείας διαχειριστή

Το υποσύστημα αυτό αποτελεί τη διεπαφή του συστήματος NaviMoz με το διαχειριστή. Το βασικό του στοιχείο είναι να διατηρεί ομαδοποιημένες σε τρεις κατηγορίες, αντίστοιχες των υποσυστημάτων 4.1.2.4-4.1.2.6, τις εργασίες που μπορεί αυτός να επιλέξει. Οι εργασίες αυτές, οι οποίες αναλύονται σε επόμενη παράγραφο, είναι ονομαστικά οι εξής:

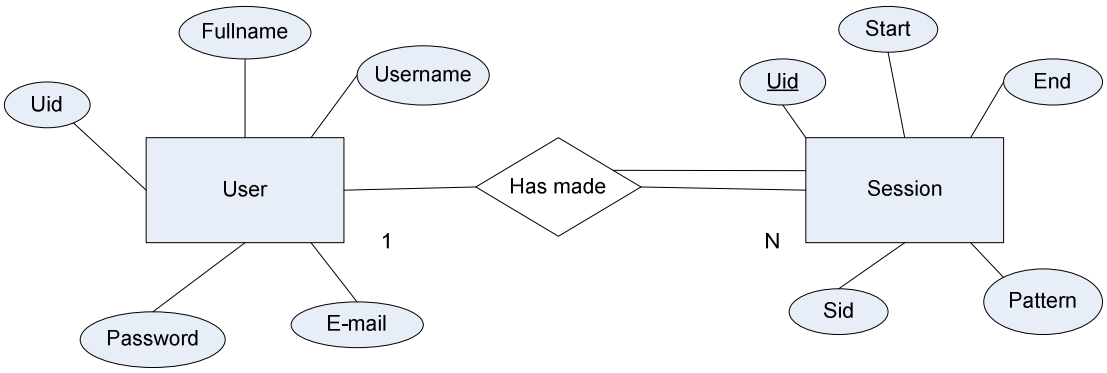
- Εμφάνιση των πλοηγήσεων ενός χρήστη που επιλέγεται από λίστα.
- Εμφάνιση της διάρκειας των πλοηγήσεων ενός χρήστη που επιλέγεται από λίστα.
- Εύρεση των πλοηγήσεων που αποτελούν υπερσύνολο μιας δοσμένης πλοήγησης.
- Εύρεση των πλοηγήσεων που είναι ταυτόσημες με μια δοσμένη πλοήγηση.
- Εύρεση των πλοηγήσεων που είναι κατά κάποιο βαθμό όμοιες με μια δοσμένη πλοήγηση.
- Εύρεση των πιο δημοφιλών πλοηγήσεων.
- Εύρεση των συστάδων των πλοηγήσεων και των χρηστών με τη μέθοδο *των Κ μέσων* (K Means)
- Εύρεση των συστάδων των πλοηγήσεων και των χρηστών με την τεχνική *μονός σύνδεσμος* (Single Link)
- Εύρεση των περισσότερο αναποφάσιστων χρηστών του συστήματος.

Και αυτό το υποσύστημα, όπως και το υποσύστημα διαπροσωπείας χρήστη, πρέπει να είναι φιλικό και να καθοδηγεί με κατάλληλες υποδείξεις και μηνύματα το διαχειριστή στη σωστή συμπλήρωση των πεδίων και των παραμέτρων στις διάφορες φόρμες. Ακόμα, πρέπει να δίνεται ιδιαίτερη προσοχή στις δυνατές επιλογές του διαχειριστή, έτσι ώστε να μην

παρουσιάζονται καν επιλογές που μπορούν να οδηγήσουν σε λάθος του συστήματος. Τέλος, πρέπει τα αποτελέσματα των αναζητήσεων να επιστρέφονται σε κατανοητή μορφή.

4.1.2.8 Υποσύστημα Βάσης Δεδομένων

Στο σημείο αυτό παρατίθενται οι λειτουργικές απαιτήσεις από τη Βάση Δεδομένων του συστήματος. Το σύστημα που κατασκευάστηκε δεν προβλέπει μεταβολή των δεδομένων, δηλαδή η Βάση Δεδομένων χρησιμοποιείται απλά για τη φύλαξη των χρηστών και των πλοηγήσεών τους, δεδομένα τα οποία μπορούν στη συνέχεια να ανακτηθούν από αυτή μέσω του υποσυστήματος του διαχειριστή. Άρα, μια απαίτηση είναι να υπάρχουν καταχωρημένοι στη Βάση Δεδομένων οι χρήστες του συστήματος κι επίσης οι πλοηγήσεις τους, με σωστή αντιστοίχηση χρήστη↔πλοήγησης. Οι απαιτήσεις από τη Βάση Δεδομένων μπορούν να κατανοηθούν καλύτερα με το παρακάτω διάγραμμα Οντοτήτων-Συσχετίσεων (E-R):



Οι οντότητες στην προκειμένη περίπτωση είναι ο χρήστης (user) και η πλοήγηση (session). Η συσχέτιση που τους συνδέει είναι η «έχει πραγματοποιήσει την», που σημαίνει ότι ένας χρήστης έχει πραγματοποιήσει μια συγκεκριμένη πλοήγηση. Ο λόγος πληθικότητας είναι 1:N, δηλαδή ένας χρήστης μπορεί να έχει πραγματοποιήσει πολλές πλοηγήσεις, όμως κάθε πλοήγηση έχει πραγματοποιηθεί από ένα μόνο χρήστη.

4.2 Σχεδίαση του συστήματος

Στην ενότητα αυτή παρουσιάζονται αναλυτικά οι εφαρμογές του συστήματος.

Για το υποσύστημα του χρήστη:

1. Εφαρμογή εισόδου υπάρχοντος χρήστη στο σύστημα
2. Εφαρμογή δημιουργίας νέου χρήστη και εισαγωγής του στο σύστημα.

3. Εφαρμογή πλοήγησης χρήστη.

Για το υποσύστημα του διαχειριστή:

4. Εφαρμογή εμφάνισης των πλοηγήσεων ενός χρήστη που επιλέγεται από λίστα.
5. Εφαρμογή εμφάνισης της διάρκειας των πλοηγήσεων ενός χρήστη που επιλέγεται από λίστα.
6. Εφαρμογή εύρεσης των πλοηγήσεων που αποτελούν υπερσύνολο μιας δοσμένης πλοήγησης.
7. Εφαρμογή εύρεσης των πλοηγήσεων που είναι ταυτόσημες με μια δοσμένη πλοήγηση.
8. Εφαρμογή εύρεσης των πλοηγήσεων που είναι κατά κάποιο βαθμό όμοιες με μια δοσμένη πλοήγηση.
9. Εφαρμογή εύρεσης των πιο δημοφιλών πλοηγήσεων.
10. Εφαρμογή εύρεσης των συστάδων των πλοηγήσεων και των χρηστών με τη μέθοδο *των Κ μέσων* (K Means)
11. Εφαρμογή εύρεσης των συστάδων των πλοηγήσεων και των χρηστών με την τεχνική *μονός σύνδεσμος* (Single Link)
12. Εφαρμογή εύρεσης των περισσότερο αναποφάσιστων χρηστών του συστήματος.

Και για το υποσύστημα της Βάσης Δεδομένων:

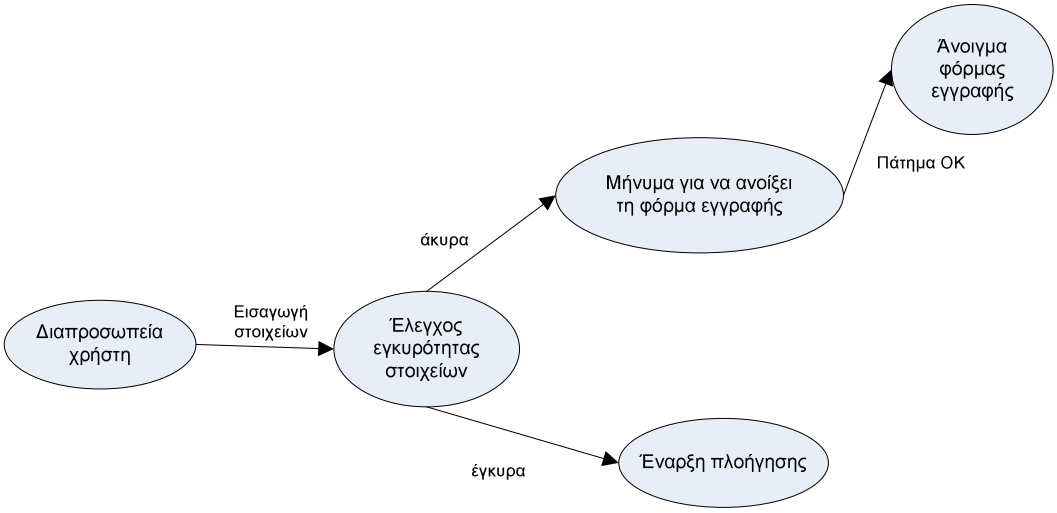
13. Εφαρμογή διαχείρισης της Βάσης Δεδομένων.

Παρακάτω παρουσιάζονται οι εφαρμογές αυτές σε συνάρτηση με τα τρία κύρια υποσυστήματα.

4.2.1 Υποσύστημα χρήστη

4.2.1.1 Εφαρμογή εισόδου υπάρχοντος χρήστη στο σύστημα

Η εφαρμογή αυτή είναι υπεύθυνη για την είσοδο ενός υπάρχοντος χρήστη στο σύστημα NaviMoz. Ζητείται από το χρήστη να εισάγει τα στοιχεία του (username και password) σε μια φόρμα. Στη συνέχεια πραγματοποιείται έλεγχος αν τα στοιχεία αυτά είναι σωστά, αν δηλαδή όντως αντιστοιχούν σε κάποιο χρήστη του συστήματος. Αν ναι, τότε επιτρέπεται στο χρήστη να πλοηγηθεί στην πύλη καταλόγου του dmoz, ανοίγει δηλαδή ένα νέο παράθυρο που είναι συνδεδεμένο με την πύλη αυτή. Αλλιώς εμφανίζεται μήνυμα που τον ενημερώνει ότι δεν είναι γραμμένος στο σύστημα και τον προτρέπει να ανοίξει τη φόρμα εγγραφής στο NaviMoz. Στο σχήμα 4.2 φαίνεται το διάγραμμα ροής της εφαρμογής αυτής:



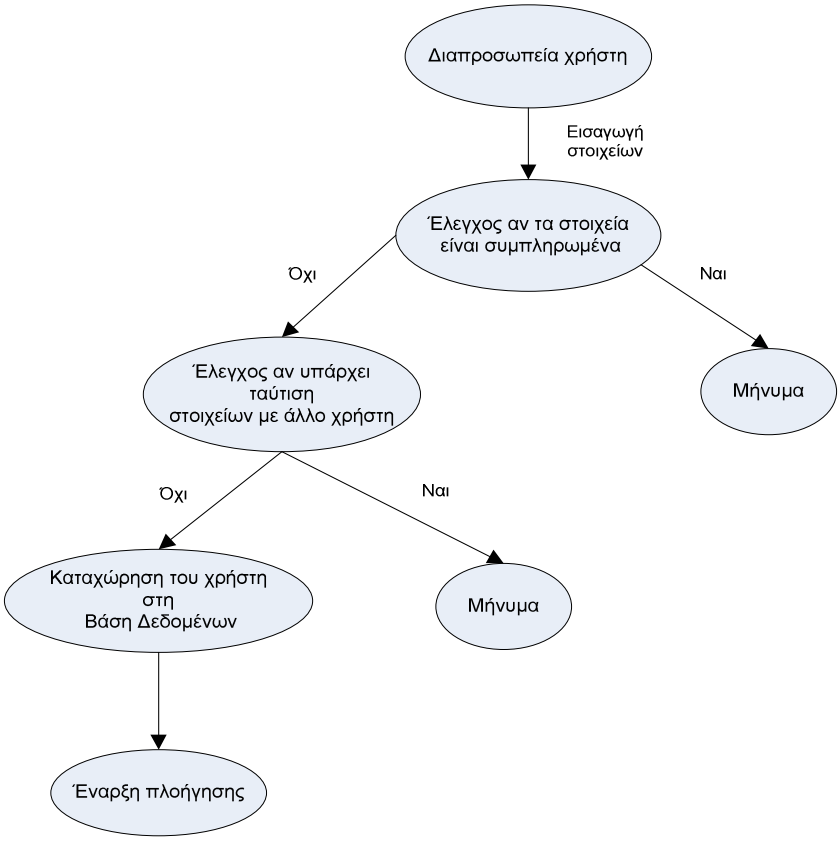
Σχήμα 4.2: Εφαρμογή εισόδου υπάρχοντος χρήστη στο σύστημα

4.2.1.2 Εφαρμογή δημιουργίας νέου χρήστη και εισαγωγής του στο σύστημα.

Η εφαρμογή αυτή είναι υπεύθυνη για τη δημιουργία νέου χρήστη και την εισαγωγή του στο σύστημα NaviMoz. Καλείται μέσω της προηγούμενης εφαρμογής(4.2.1.1). Ζητείται από το χρήστη να συμπληρώσει μια φόρμα εγγραφής η οποία περιλαμβάνει τα εξής πεδία: Όνομα, Επώνυμο, Όνομα χρήστη (Username), Κωδικός πρόσβασης (Password) και Ηλεκτρονική διεύθυνση (e-mail). Επειδή όλα τα πεδία πρέπει να είναι συμπληρωμένα προκειμένου να γραφτεί ο χρήστης στο σύστημα, σε περίπτωση που κάποιο πεδίο είναι κενό, η εφαρμογή προτρέπει το χρήστη να το συμπληρώσει. Βασικό επίσης κομμάτι της εφαρμογής αυτής, είναι ότι πραγματοποιεί έλεγχο για το αν υπάρχει κι άλλος χρήστης στο σύστημα με το ίδιο όνομα χρήστη ή τον ίδιο κωδικό πρόσβασης, επειδή αυτά είναι τα αναγνωριστικά στοιχεία του κάθε χρήστη (όπως περιγράφηκε στην εφαρμογή 4.2.1.1). Σε περίπτωση που βρεθεί ταύτιση στοιχείων ο χρήστης ειδοποιείται με κατάλληλο μήνυμα, το οποίο τον προτρέπει να χρησιμοποιήσει άλλα αναγνωριστικά στοιχεία. Το διάγραμμα ροής της εφαρμογής αυτής δίνεται στο σχήμα 4.3

4.2.1.3 Εφαρμογή πλοήγησης χρήστη

Η εφαρμογή αυτή είναι υπεύθυνη για την αποθήκευση της πλοήγησης του χρήστη στη Βάση Δεδομένων σε κατάλληλη μορφή. Πιο συγκεκριμένα, ανοίγει ένα νέο παράθυρο στην οθόνη του χρήστη στο οποίο πάνω φορτώνει την ιεραρχία του dmoz. Ανοίγει δηλαδή μπροστά του η αρχική σελίδα www.dmoz.org . Ο χρήστης μπορεί από τη στιγμή που εμφανίζεται το νέο παράθυρο να αλληλεπιδρά με αυτό, να επιλέγει δηλαδή την επόμενη κατηγορία της ιεραρχίας που επιθυμεί να επισκεφτεί και να οδηγείται σ’ αυτή.

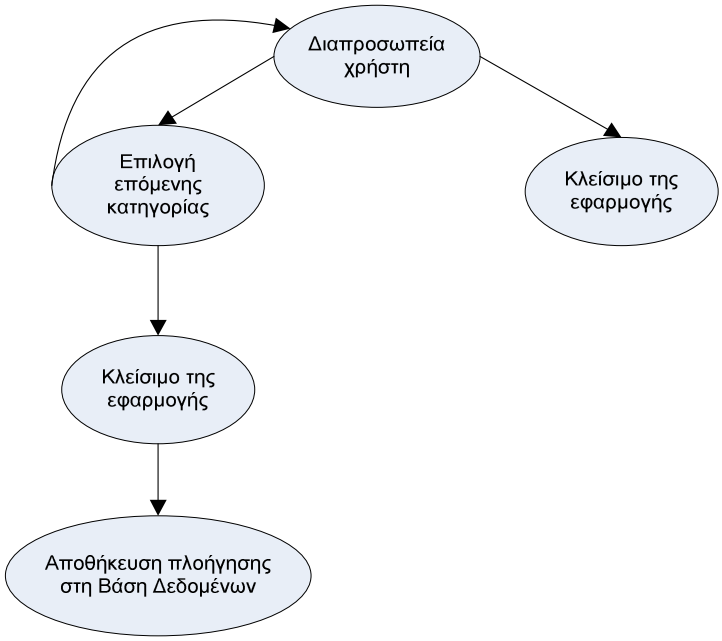


Σχήμα 4.3: Εφαρμογή δημιουργίας νέου χρήστη και εισαγωγής του στο σύστημα

Στο σημείο αυτό υπενθυμίζεται ότι η ιεραρχία του dmoz είναι οργανωμένη σε γράφο, πράγμα που σημαίνει ότι ο χρήστης μπορεί να βρεθεί στην ίδια κατηγορία έχοντας ακολουθήσει διαφορετικές διαδρομές. Στόχος της διπλωματικής αυτής εργασίας είναι η εξέταση των πλοηγήσεων των χρηστών ως ακολουθιών διαφορετικών κατηγοριών της ιεραρχίας που επισκέπτονται. Δεν εξετάζονται καθόλου οι ιστοσελίδες (sites) τις οποίες μπορεί να επισκεφτεί με υπερσύνδεσμο μέσω της ιεραρχίας, οι οποίες όμως δεν ανήκουν σε αυτή. Οι επισκέψεις τους αυτές μετατρέπονται σε Strings, τα οποία τελικά δημιουργούν το συνολικό String που καταχωρείται στη Βάση Δεδομένων. Η εφαρμογή φροντίζει ώστε να καταχωρείται στην πλοήγηση το τελευταίο στοιχείο του URL το οποίο αντιστοιχεί στη σελίδα που επισκέπτεται ο χρήστης. Για παράδειγμα, αν ο χρήστης, ξεκινώντας από την αρχική σελίδα www.dmoz.org βρεθεί στην κατηγορία Arts και από εκεί επιλέξει να επισκεφτεί την κατηγορία Photography, αυτό που αποθηκεύεται ως επόμενο στοιχείο της πλοήγησης είναι το Photography, και όχι το http://www.dmoz.org/Arts/Photography. Έτσι η πλοήγησή του μέχρι αυτή τη στιγμή θα είναι η www.dmoz.org/Arts/Photography.

Η εφαρμογή αυτή παρέχει όλες τις δυνατότητες ενός Browser, είναι δηλαδή ενεργοποιημένα τα πλήκτρα Back και Forward σε περίπτωση που ο χρήστης επιθυμεί να βρεθεί σε κάποια από τις προηγούμενες τοποθεσίες της επίσκεψής του. Επίσης το παράθυρο της εφαρμογής μπορεί

να μεγεθυνθεί, να σμικρυνθεί ή να κλείσει τελείως. Τέλος είναι ενεργοποιημένο το πλήκτρο Exit, το οποίο, μαζί με το πλήκτρο για κλείσιμο του παραθύρου (εικονίδιο X), κλείνει την εφαρμογή. Με το πάτημα των πλήκτρων αυτών, η εφαρμογή κλείνει και η πλοήγηση που έχει δημιουργηθεί ως εκείνη τη στιγμή αποθηκεύεται στη Βάση Δεδομένων. Αξίζει να σημειωθεί ότι η εφαρμογή πραγματοποιεί έλεγχο έτσι ώστε ο χρήστης να έχει επισκεφτεί τουλάχιστον μία κατηγορία της ιεραρχίας πριν αποθηκεύσει την πλοήγησή του. Έτσι αποφεύγεται το ενδεχόμενο καταχώρησης κενής (null) εγγραφής στη Βάση Δεδομένων. Η χρονική στιγμή έναρξης της πλοήγησης είναι αυτή κατά την οποία ο χρήστης επιλέγει την πρώτη κατηγορία που θα επισκεφτεί. Στο παρακάτω σχήμα (4.4) δίνεται το διάγραμμα ροής της εφαρμογής:



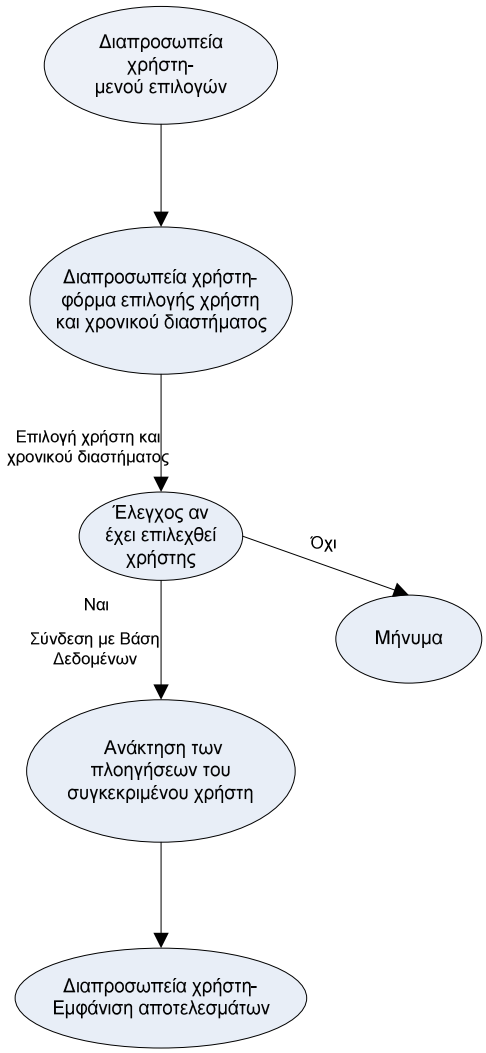
Σχήμα 4.4: Εφαρμογή πλοήγησης χρήστη

4.2.2 Υποσύστημα Διαχειριστή

4.2.2.1 Εφαρμογή εμφάνισης των πλοηγήσεων ενός χρήστη που επιλέγεται από λίστα.

Η εφαρμογή αυτή είναι υπεύθυνη για την εμφάνιση των πλοηγήσεων ενός συγκεκριμένου χρήστη του συστήματος εντός μιας συγκεκριμένης χρονικής περιόδου. Ανήκει στην κατηγορία εργασιών ομαδοποίησης χρηστών. Πιο αναλυτικά, η εργασία αυτή πραγματοποιείται από το σύστημα, όταν ο διαχειριστής την επιλέξει από την πρώτη κατηγορία εργασιών. Τότε παρουσιάζεται μια φόρμα η οποία τον προτρέπει να επιλέξει το όνομα του χρήστη του οποίου τις πλοηγήσεις θέλει να δει. Το όνομα του χρήστη επιλέγεται

από μια λίστα, έτσι ώστε να αποφεύγεται το ενδεχόμενο λάθος πληκτρολόγησης ενός ονόματος αλλά και για να μη χρειάζεται να γνωρίζει από πριν και να θυμάται όλους τους χρήστες του συστήματος. Επίσης, ο διαχειριστής καλείται να επιλέξει ή να πληκτρολογήσει ο ίδιος τις ημερομηνίες εντός των οποίων τις πλοηγήσεις θα ήθελε να γνωρίζει. Με το πάτημα του πλήκτρου ΟΚ του παρουσιάζονται τα αποτελέσματα στην αρχική φόρμα του συστήματος. Στο σημείο αυτό πραγματοποιείται έλεγχος αν έχει επιλεχθεί όνομα χρήστη και, στην περίπτωση που δεν έχει, εμφανίζεται κατάλληλο μήνυμα. Το διάγραμμα ροής της εφαρμογής δίνεται στο σχήμα 4.5:

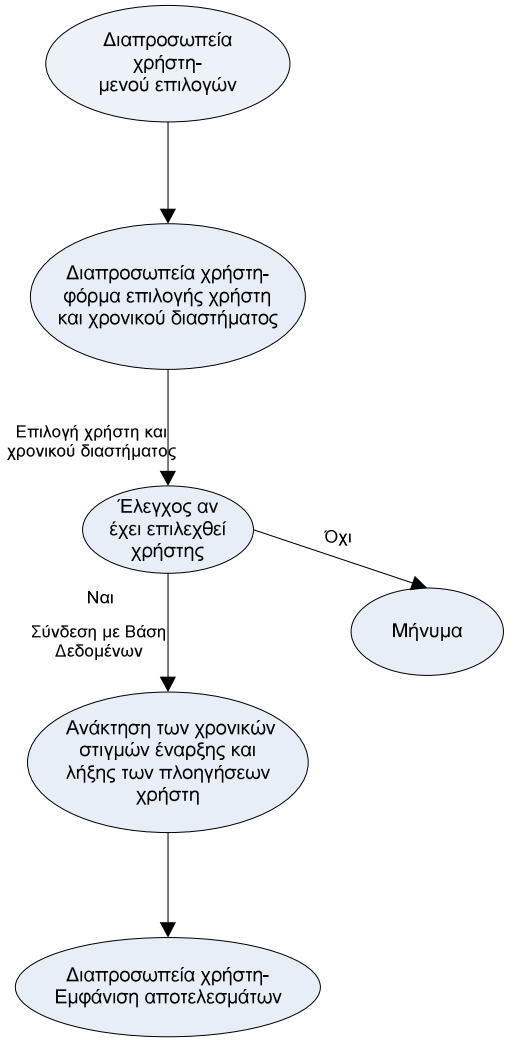


Σχήμα 4.5: Εφαρμογή εμφάνισης των πλοηγήσεων ενός χρήστη που επιλέγεται από λίστα.

4.2.2.2 Εφαρμογή εμφάνισης της διάρκειας των πλοηγήσεων ενός χρήστη που επιλέγεται από λίστα.

Η εφαρμογή αυτή είναι υπεύθυνη για την εμφάνιση της διάρκειας των πλοηγήσεων ενός συγκεκριμένου χρήστη του συστήματος. Ανήκει και αυτή στην κατηγορία εργασιών

ομαδοποίησης χρηστών. Η εργασία αυτή πραγματοποιείται από το σύστημα, όταν ο διαχειριστής την επιλέξει από την πρώτη κατηγορία εργασιών. Τότε παρουσιάζεται μια φόρμα η οποία τον προτρέπει να επιλέξει το όνομα του χρήστη και το χρονικό διάστημα ενδιαφέροντος. Οι επιλογές παρουσιάζονται με τρόπο όμοιο με αυτόν που περιγράφηκε στην εξήγηση της εφαρμογής 4.2.2.1. Με το πάτημα του πλήκτρου ΟΚ παρουσιάζονται τα αποτελέσματα με τη μορφή καταγραφής της ώρας και της ημερομηνίας έναρξης και λήξης της κάθε πλοήγησης. Το διάγραμμα ροής της εφαρμογής δίνεται στο σχήμα 4.6:



Σχήμα 4.6: Εφαρμογή εμφάνισης της διάρκειας των πλοηγήσεων ενός χρήστη που επιλέγεται από λίστα.

4.2.2.3 Εφαρμογή εύρεσης των πλοηγήσεων που αποτελούν υπερσύνολο μιας δοσμένης πλοήγησης

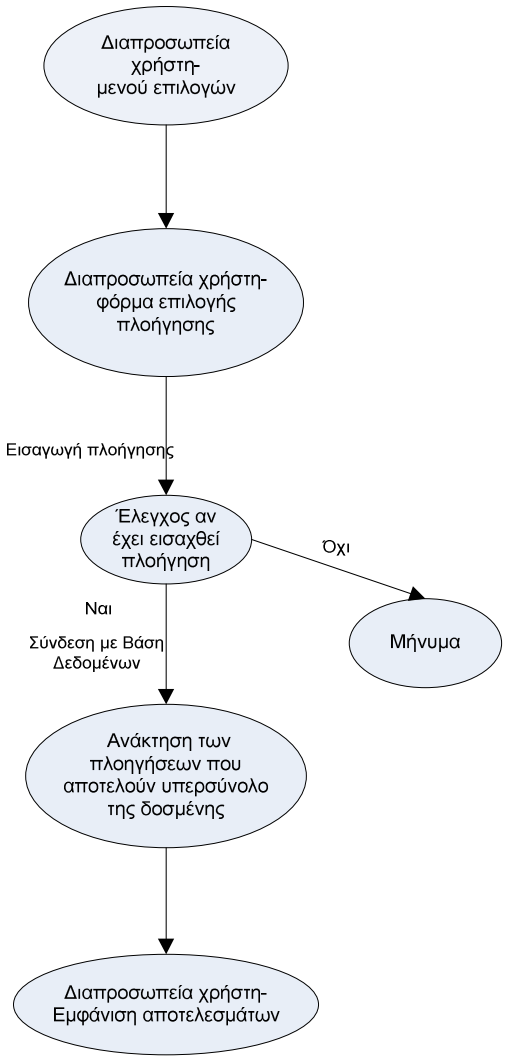
Η εφαρμογή αυτή ανήκει στην κατηγορία ερωτήσεων εξόρυξης δεδομένων. Το χαρακτηριστικό της κατηγορίας αυτής είναι το ότι ο διαχειριστής δίνει ως παράμετρο μια συγκεκριμένη πλοήγηση και το σύστημα ανακτά πληροφορίες που σχετίζονται με την πλοήγηση-πρότυπο αυτή. Η συγκεκριμένη εφαρμογή ανακτά τις πλοηγήσεις εκείνες οι οποίες αποτελούν υπερσύνολο της δοσμένης από το διαχειριστή πλοήγησης, εκείνες δηλαδή στις οποίες περιέχεται η δοσμένη πλοήγηση. Η εργασία αυτή πραγματοποιείται από το σύστημα, όταν ο διαχειριστής την επιλέξει από τη δεύτερη κατηγορία εργασιών. Τότε παρουσιάζεται μια φόρμα η οποία του ζητά να δώσει την πλοήγηση-πρότυπο και το χρονικό διάστημα ενδιαφέροντος. Η εφαρμογή δίνει τη δυνατότητα στο διαχειριστή να τυπώσει απευθείας την πλοήγηση ή να ανοίξει έναν Browser και να πλοηγηθεί στην ιεραρχία του dmoz, κατασκευάζοντας έτσι με οπτικό τρόπο την επιθυμητή πλοήγηση. Επίσης, του ζητείται να επιλέξει τις ημερομηνίες εντός των οποίων οι πλοηγήσεις θα εξεταστούν. Με το πάτημα του πλήκτρου OK τα αποτελέσματα εμφανίζονται στην αρχική φόρμα του συστήματος. Στο σημείο αυτό πρέπει να σημειωθεί ότι σε περίπτωση που ο διαχειριστής δεν έχει δώσει κάποια πλοήγηση στο κατάλληλο πεδίο, εμφανίζεται προειδοποιητικό μήνυμα που τον προτρέπει να συμπληρώσει το πεδίο αυτό. Το διάγραμμα ροής της εφαρμογής αυτής δίνεται στο σχήμα 4.7.

4.2.2.4 Εφαρμογή εύρεσης των πλοηγήσεων που είναι ταυτόσημες με μια δοσμένη πλοήγηση

Η εφαρμογή αυτή ανήκει στην κατηγορία ερωτήσεων εξόρυξης δεδομένων. Η περιγραφή της είναι ακριβώς η ίδια με την περιγραφή της προηγούμενης εφαρμογής (4.2.2.3) με τις μόνες διαφορές ότι αυτή τη φορά η αντίστοιχη φόρμα εμφανίζεται όταν ο διαχειριστής επιλέξει αυτή την εργασία από τη δεύτερη κατηγορία εργασιών, και ότι ανακτώνται από τη Βάση Δεδομένων οι πλοηγήσεις εκείνες που ταυτίζονται ακριβώς με την πλοήγηση που εισάγει ο διαχειριστής. Το διάγραμμα ροής της εφαρμογής αυτής δίνεται στο σχήμα 4.8.

4.2.2.5 Εφαρμογή εύρεσης των πλοηγήσεων που είναι κατά ένα βαθμό όμοιες με μια δοσμένη πλοήγηση

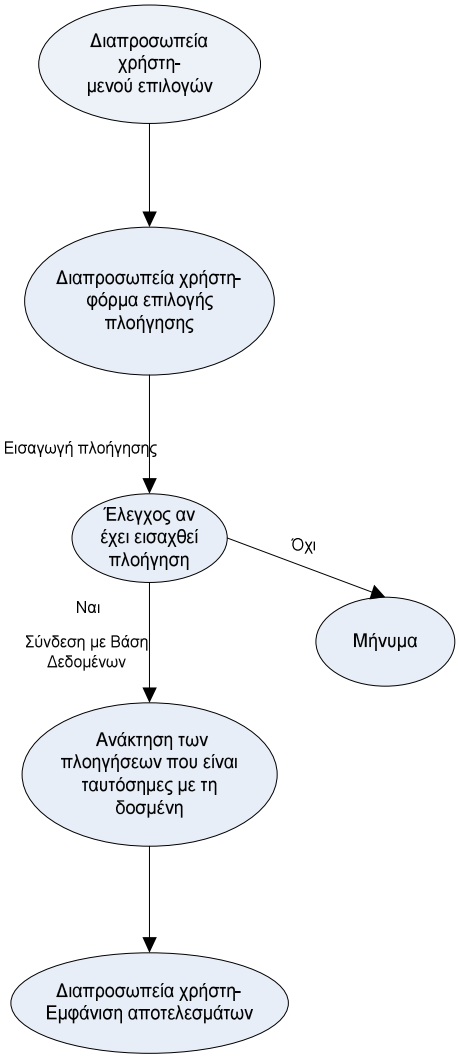
Και αυτή η εφαρμογή ανήκει στην κατηγορία ερωτήσεων εξόρυξης δεδομένων. Επιστρέφει τις πλοηγήσεις εκείνες οι οποίες είναι κατά τουλάχιστον ένα ποσοστό όμοιες με μία δοσμένη από το διαχειριστή πλοήγηση.



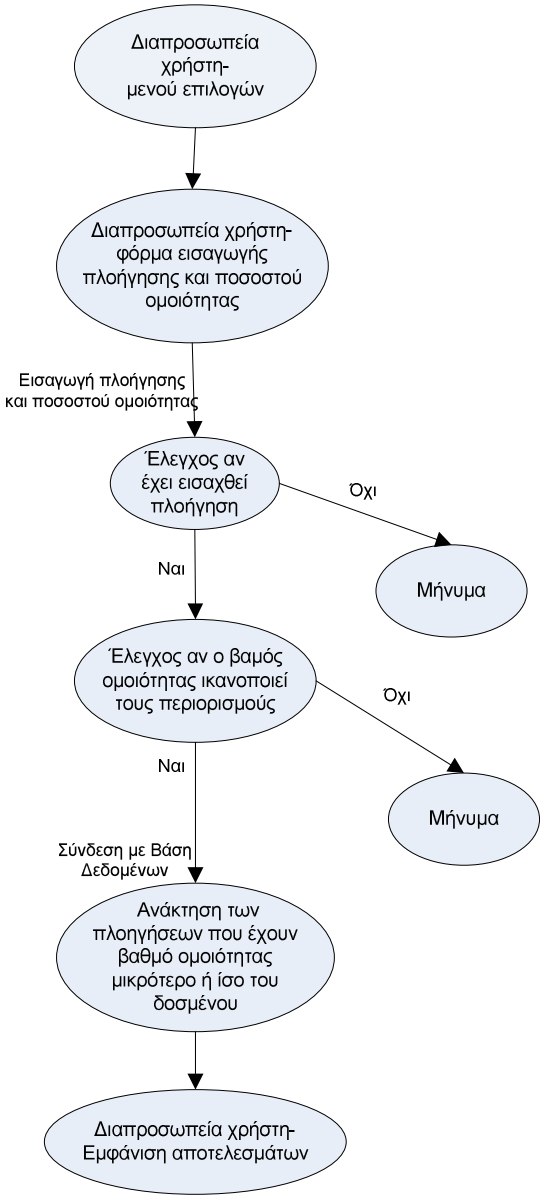
Σχήμα 4.7: Εφαρμογή εύρεσης των πλοηγήσεων που αποτελούν υπερσύνολο μιας δοσμένης πλοήγησης

Το ποσοστό ομοιότητας δίνεται επίσης από το διαχειριστή. Η αντίστοιχη φόρμα εμφανίζεται όταν ο χρήστης επιλέξει την εύρεση όμοιων πλοηγήσεων από τη δεύτερη κατηγορία εργασιών. Τότε ανοίγει μια φόρμα η οποία ζητά από το διαχειριστή να εισάγει μια συγκεκριμένη πλοήγηση-πρότυπο (είτε απευθείας, είτε μέσω του browser του dmoz) κι ένα ποσοστό αντίστοιχο της ομοιότητας μεταξύ της δοσμένης πλοήγησης και αυτών που θα ανακτηθούν. Το ποσοστό αυτό δίνεται σε μονάδες % , όπου το 100 δηλώνει απόλυτη ταύτιση των πλοηγήσεων και το 0 πλοηγήσεις εντελώς διαφορετικές μεταξύ τους. Η ομοιότητα των πλοηγήσεων υπολογίζεται με εύρεση της δομικής απόστασης (μέσω του αλγορίθμου Struct_Dist) και με κατάλληλους μετασχηματισμούς αυτής σε βαθμό ομοιότητας. Επίσης ζητείται από το διαχειριστή, όπως και σε όλες τις προηγούμενες φόρμες, να εισάγει το χρονικό διάστημα ενδιαφέροντος. Η εφαρμογή αυτή πραγματοποιεί τους εξής ελέγχους:

Όταν ο χρήστης πατήσει το πλήκτρο ΟΚ, αρχικά εξετάζεται αν έχει εισάγει κάποια πλοήγηση στο κατάλληλο πεδίο, και αν όχι εμφανίζεται προειδοποιητικό μήνυμα. Στη συνέχεια, εξετάζεται αν έχει εισάγει το ποσοστό ομοιότητας στη ζητούμενη μορφή. Εμφανίζεται μήνυμα αν ο αριθμός που έχει δώσει δεν είναι μεταξύ του 0 και του 100 ή αν έχει συμπληρώσει το πεδίο αυτό με άλλο χαρακτήρα εκτός από αριθμό. Αν όλοι οι έλεγχοι διεξαχθούν επιτυχώς, τα αποτελέσματα εμφανίζονται στην αρχική φόρμα του συστήματος. Το διάγραμμα ροής της εφαρμογής δίνεται στο σχήμα 4.9.



Σχήμα 4.8: Εφαρμογή εύρεσης των πλοηγήσεων που είναι ταυτόσημες με μια δοσμένη πλοήγηση

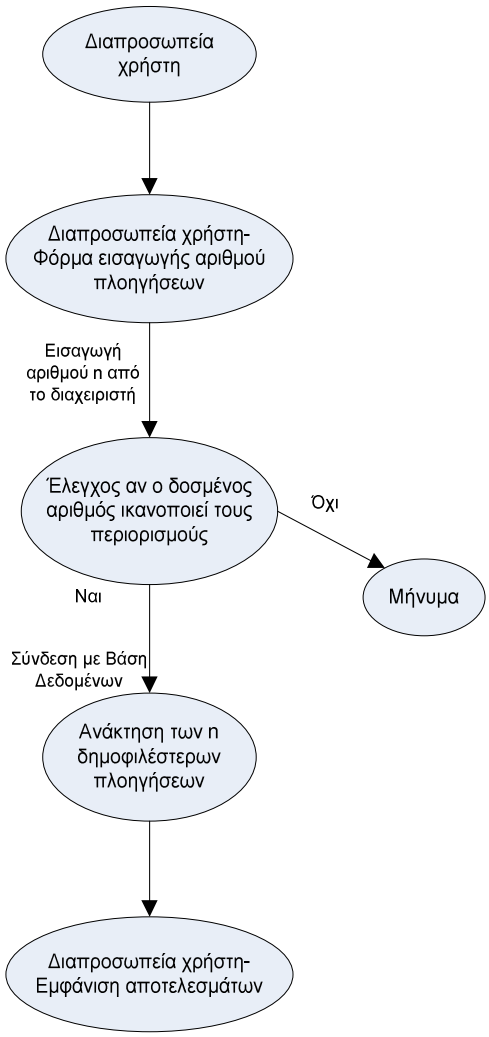


Σχήμα 4.9: Εφαρμογή εύρεσης των πλοηγήσεων που είναι κατά ένα βαθμό όμοιες με μια δοσμένη πλοήγηση

4.2.2.6 Εφαρμογή εύρεσης των πιο δημοφιλών πλοηγήσεων

Η εφαρμογή αυτή ανήκει στη γενικότερη κατηγορία ερωτήσεων ομαδοποίησης πλοηγήσεων και χρηστών. Το χαρακτηριστικό της κατηγορίας αυτής είναι ότι ο διαχειριστής δεν εισάγει κάποια συγκεκριμένη πλοήγηση βάσει της οποίας πραγματοποιείται η εξόρυξη δεδομένων. Αντίθετα, το σύστημα εξετάζει όλες τις πλοηγήσεις μεταξύ τους και τις ομαδοποιεί μα βάση το κριτήριο που επιλέγει ο διαχειριστής. Η συγκεκριμένη εφαρμογή ομαδοποιεί τις πλοηγήσεις με βάση το πόσο συχνά έχουν πραγματοποιηθεί. Ο διαχειριστής επιλέγει την εκτέλεση της συγκεκριμένης εργασίας από την τρίτη κατηγορία εργασιών. Τότε ανοίγει μια

φόρμα που του ζητά να δώσει έναν αριθμό, ο οποίος αντιστοιχεί στο πλήθος των πιο δημοφιλών πλοηγήσεων, καθώς και το χρονικό διάστημα ενδιαφέροντος. Με το πάτημα του πλήκτρου ΟΚ από το διαχειριστή, πράγμα που σημαίνει ότι θέλει να προχωρήσει στην εμφάνιση των αποτελεσμάτων, πραγματοποιείται έλεγχος αν ο αριθμός που έχει δώσει είναι ακέραιος μεγαλύτερος του 0 και αν είναι μεγαλύτερος από τον αριθμό των υπαρχόντων πλοηγήσεων. Και στις δύο περιπτώσεις προβάλλεται κατάλληλο μήνυμα. Ειδικά στη δεύτερη περίπτωση, το σύστημα προτρέπει το διαχειριστή να εισάγει έναν αριθμό μικρότερο ενός συγκεκριμένου που αντιστοιχείς στο πλήθος των πλοηγήσεων. Αν όλοι οι έλεγχοι διεξαχθούν επιτυχώς παρουσιάζονται τα αποτελέσματα στην αρχική φόρμα του συστήματος. Το διάγραμμα ροής δίνεται στο σχήμα 4.10:



Σχήμα 4.10: Εφαρμογή εύρεσης των πιο δημοφιλών πλοηγήσεων

4.2.2.7 Εφαρμογή εύρεσης των συστάδων των πλοηγήσεων και των χρηστών με τη μέθοδο των Κ μέσων (K Means)

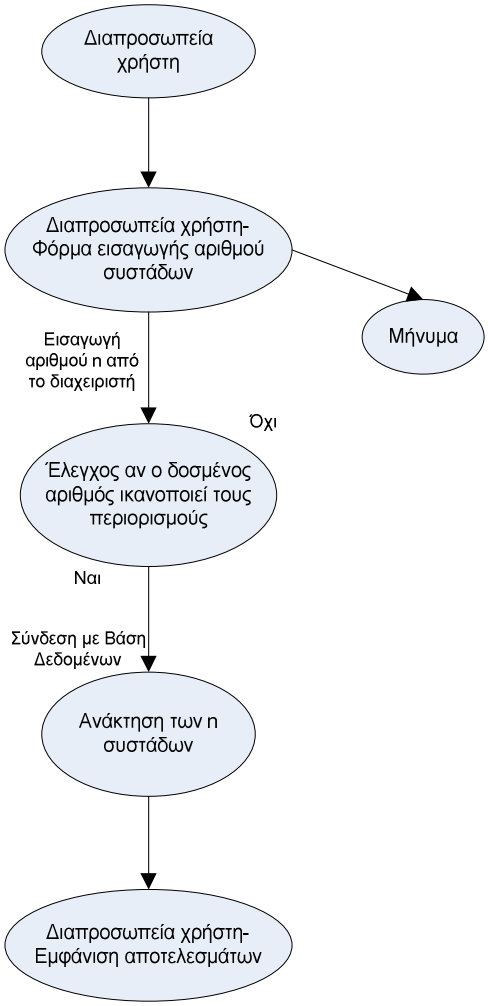
Η εφαρμογή αυτή αποτελεί μια από τις σημαντικότερες λειτουργίες του συστήματος. Ανήκει και αυτή στην κατηγορία ομαδοποίησης πλοηγήσεων και χρηστών, η οποία περιγράφηκε στην προηγούμενη εφαρμογή. Το επιπλέον εδώ έγκειται στο ότι το σύστημα, με βάση την ομαδοποίηση των πλοηγήσεων, και δεδομένου ότι έχει κρατήσει πληροφορία σχετικά με το ποιος χρήστης έχει πραγματοποιήσει την κάθε πλοήγηση, προχωρά σε ομαδοποίηση των χρηστών με βάση πάντα το δοσμένο κριτήριο. Η συγκεκριμένη εφαρμογή πραγματοποιεί συσταδοποίηση των πλοηγήσεων και των αντίστοιχων χρηστών με υλοποίηση του αλγορίθμου συσταδοποίησης Κ-Μέσων (K-Means). Όταν ο διαχειριστής επιλέγει την πραγματοποίηση της εργασίας αυτής από την τρίτη κατηγορία εργασιών, εμφανίζεται μια φόρμα η οποία του ζητά να συμπληρώσει πόσες συστάδες επιθυμεί να ανακτηθούν (Ο αλγόριθμος των Κ-Μέσων απαιτεί να γνωρίζει εκ των προτέρων τον αριθμό των συστάδων που θα δημιουργήσει), καθώς και το χρονικό διάστημα ενδιαφέροντος. Όταν ο διαχειριστής θελήσει να δει τα αποτελέσματα, πραγματοποιείται έλεγχος για το αν ο δοσμένος αριθμός είναι ακέραιος μεγαλύτερος του 0 και στην περίπτωση που δεν είναι προβάλλεται κατάλληλο μήνυμα. Αν ο έλεγχος διεξαχθεί σωστά παρουσιάζονται στο διαχειριστή οι συστάδες των πλοηγήσεων και των χρηστών που τις έχουν πραγματοποιήσει. Το διάγραμμα ροής της εφαρμογής δίνεται στο σχήμα 4.11.

4.2.2.8 Εφαρμογή εύρεσης των συστάδων των πλοηγήσεων και των χρηστών με την τεχνική «μονός σύνδεσμος» (“Single Link”)

Η εφαρμογή αυτή είναι επίσης μία πολύ σημαντική εφαρμογή του συστήματος. Αντίστοιχα με την προηγούμενη (4.2.2.7) πραγματοποιεί συσταδοποίηση των πλοηγήσεων και των αντιστοίχων χρηστών με βάσει της τεχνικής «μονός σύνδεσμος». Όταν ο διαχειριστής επιλέγει την πραγματοποίηση της εργασίας αυτής από την τρίτη κατηγορία εργασιών, εμφανίζεται μια φόρμα ή οποία τον προτρέπει να συμπληρώσει έναν αριθμό ενδεικτικό του επιπέδου συσταδοποίησης (Η τεχνική «μονός σύνδεσμος» απαιτεί να γνωρίζει από πριν το επίπεδο συσταδοποίησης το οποίο δεν είναι τίποτε άλλο παρά ένας αριθμός ενδεικτικός του πότε σταματά η εκτέλεση του αλγορίθμου), ο οποίος είναι δεκαδικός μεταξύ 0 και 1 και αντιστοιχεί στο σημείο στο οποίο ο αλγόριθμος θα σταματήσει τη συγχώνευση των συστάδων και την ανάπτυξη του δενδρογράμματος προς τα πάνω (υποτίθεται bottom-up υλοποίηση). Στο σημείο αυτό παρέχεται στο διαχειριστή η δυνατότητα, πατώντας ένα κουμπί, να δει το επίπεδο εκείνο που αντιστοιχεί στην καλύτερη συσταδοποίηση των δεδομένων, όπως αυτό υπολογίζεται από τον αλγόριθμο C-Index. Ο αλγόριθμος C-Index υπολογίζει το επίπεδο συσταδοποίησης για το οποίο η προκύπτουσα συσταδοποίηση είναι η βέλτιστη

δυνατή. Έτσι, αποφεύγονται οι δοκιμές διαφόρων τιμών για το επίπεδο συσταδοποίησης, οι οποίες μπορεί τελικά και να μην καταλήξουν στην επιλογή του καταλληλότερου επιπέδου.

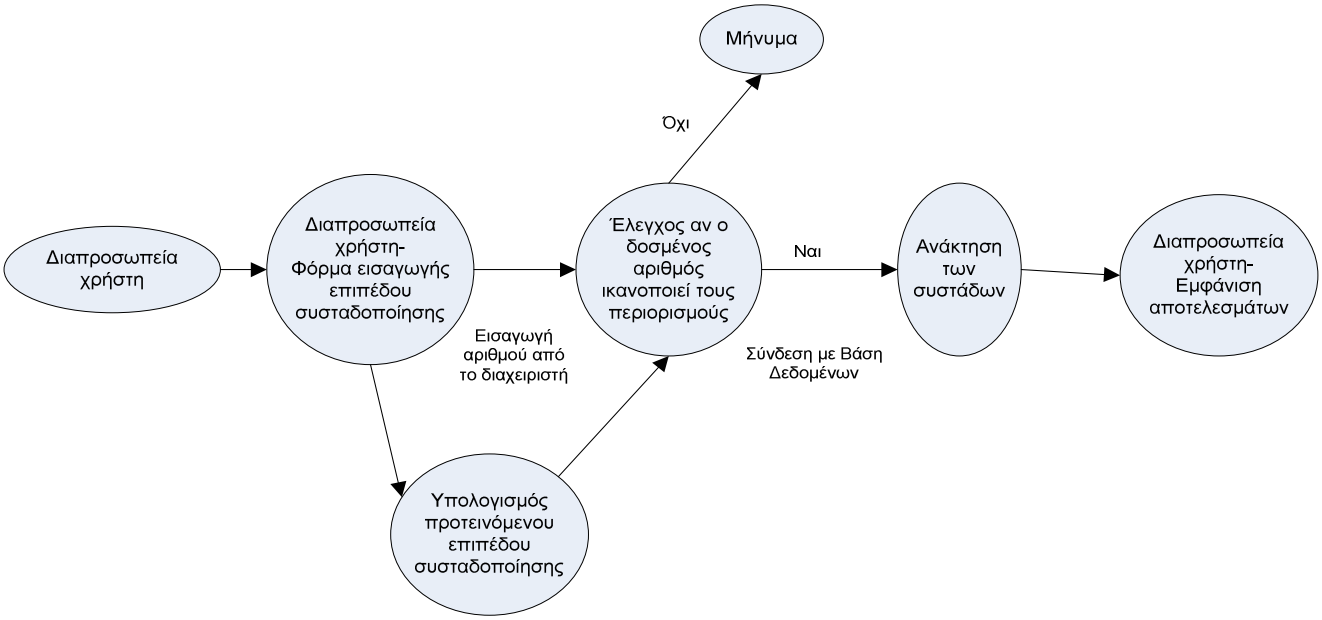
Ο χρήστης αφήνεται ελεύθερος να επιλέξει ανάμεσα στο προτεινόμενο επίπεδο συσταδοποίησης και σ' ένα άλλο. Επίσης του ζητείται να δοθεί το χρονικό διάστημα ενδιαφέροντος · σημειώνεται ότι το διάστημα αυτό πρέπει να έχει επιλεχθεί πριν από την εφαρμογή του αλγορίθμου C-Index γιατί αυτός πρέπει αν υλοποιηθεί με βάση τις πλοηγήσεις που έχουν πραγματοποιηθεί στο διάστημα αυτό. Με το πάτημα του πλήκτρου OK, από το διαχειριστή πραγματοποιείται έλεγχος για το αν ο δοσμένος αριθμός είναι δεκαδικός μεταξύ του 0 και του 1, και στην περίπτωση που δεν είναι προβάλλεται κατάλληλο μήνυμα. Αν ο έλεγχος διεξαχθεί σωστά παρουσιάζονται στο διαχειριστή οι συστάδες των πλοηγήσεων και των χρηστών που τις έχουν πραγματοποιήσει. Το διάγραμμα ροής της εφαρμογής δίνεται στο σχήμα 4.12.



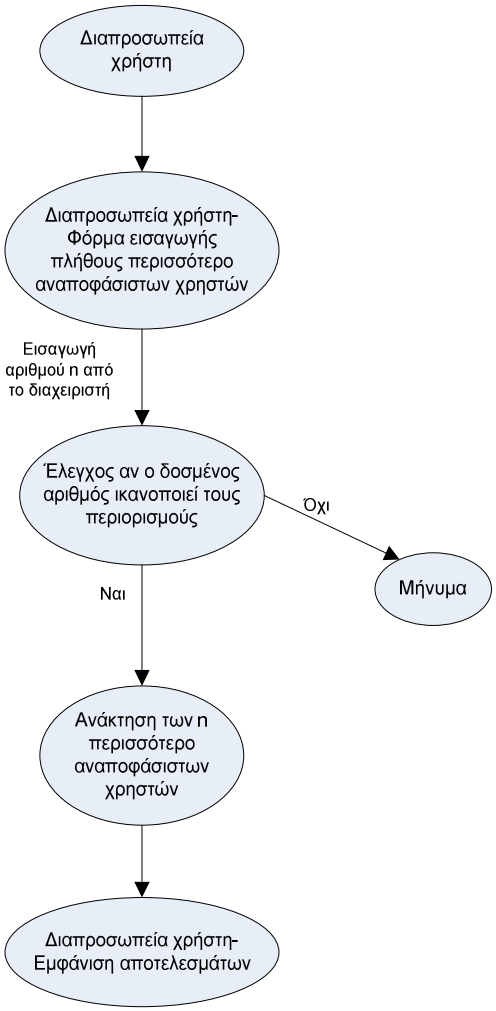
Σχήμα 4.11: Εφαρμογή εύρεσης των συστάδων με τη μέθοδο των K μέσων

4.2.2.9 Εφαρμογή εύρεσης των περισσότερο αναποφάσιστων χρηστών του συστήματος

Σημαντική εφαρμογή του συστήματος και αυτή, ανακτά τους πιο αναποφάσιστους χρήστες του συστήματος. Όπως περιγράφεται και σε επόμενο κεφάλαιο, η αναποφασιστικότητα των χρηστών μετράται βάσει του συνολικού αριθμού από Back και Forward που έχουν πραγματοποιήσει. Όπως θα εξηγηθεί και παρακάτω, δύο κόμβοι έχουν την ιδιότητα Back και Forward όταν ο δεύτερος είναι ίδιος με τον πρώτο και έχει προκύψει από πατήματα ενός από τα πλήκτρα Back ή Forward. Όταν δηλαδή ο χρήστης επισκεφθεί μια σελίδα που είχε επισκεφθεί και στο παρελθόν, και τη δεύτερη φορά καταλήγει σε αυτή μετά από ένα ή περισσότερα πατήματα των πλήκτρων Back ή Forward, οι δύο αυτές σελίδες χαρακτηρίζονται από την ιδιότητα Back και Forward. Έστω λοιπόν ότι ο χρήστης βρίσκεται στη σελίδα Arts/Photography/Photographers και θέλει να επιστρέψει στην Arts/Photography . Τότε θα πατήσει το πλήκτρό Back, και οι δύο αυτές επισκέψεις της ίδιας σελίδας Arts/Photography θα καταγραφούν από το σύστημα NaviMoz ως δύο κόμβοι Back και Forward. Έτσι λοιπόν, η εφαρμογή αυτή ουσιαστικά βρίσκει για κάθε πλοήγηση του κάθε χρήστη πόσα Back και Forward περιέχει και στη συνέχεια αθροίζει (για κάθε χρήστη) το πλήθος των Back και Forward που του αντιστοιχεί. Στη συνέχεια κατατάσσει τους χρήστες κατά φθίνοντα αριθμό πραγματοποιούμενων Back και Forward. Όταν ο διαχειριστής επιλέξει την πραγματοποίηση της εργασίας αυτής από την τρίτη ομάδα εργασιών, εμφανίζεται μια φόρμα ή οποία τον προτρέπει να συμπληρώσει έναν αριθμό n, ο οποίος αντιστοιχεί στο πλήθος των πιο αναποφάσιστων χρηστών, καθώς και το χρονικό διάστημα ενδιαφέροντος. . Με το πάτημα του πλήκτρου OK από το διαχειριστή, πραγματοποιείται έλεγχος αν ο αριθμός που έχει δώσει είναι ακέραιος μεγαλύτερος του 0 και αν είναι μεγαλύτερος από τον αριθμό των υπαρχόντων χρηστών του συστήματος. Και στις δύο περιπτώσεις προβάλλεται κατάλληλο μήνυμα. Ειδικά στη δεύτερη περίπτωση, το σύστημα προτρέπει το διαχειριστή να εισάγει έναν αριθμό μικρότερο ενός συγκεκριμένου που αντιστοιχεί στο πλήθος των χρηστών. Αν όλοι οι έλεγχοι διεξαχθούν σωστά παρουσιάζονται στο διαχειριστή οι n περισσότερο αναποφάσιστοι χρήστες και ο βαθμός αναποφασιστικότητάς τους. Το αντίστοιχο διάγραμμα ροής δίνεται στο σχήμα 4.13



Σχήμα 4.12: Εφαρμογή εύρεσης των συστάδων με την τεχνική «μονός σύνδεσμος»



Σχήμα 4.13: Εφαρμογή εύρεσης των περισσότερο αναποφάσιστων χρηστών

Στο σημείο αυτό πρέπει να αναφερθεί ότι για όλες τις εφαρμογές που υπάγονται στο υποσύστημα του διαχειριστή, το χρονικό διάστημα ενδιαφέροντος επιλέγεται με επιλογή της ημερομηνίας έναρξης και της ημερομηνίας λήξης από δύο αντίστοιχες λίστες. Οι λίστα της ημερομηνίας έναρξης αρχικοποιείται στην ημερομηνία 1/1/2000 και η αντίστοιχη της ημερομηνίας λήξης στην τρέχουσα. Κατ' αυτό τον τρόπο, ακόμα και αν ο διαχειριστής ξεχαστεί και παραβλέψει το βήμα αυτό, το σύστημα επιτελεί την εργασία για τις προεπιλεγμένες ημερομηνίες. Με λίγα λόγια δηλαδή, πραγματοποιεί την εργασία για όλες τις αποθηκευμένες πλοηγήσεις.

Τέλος, αν δεν υπάρχουν πλοηγήσεις που να ικανοποιούν τα εκάστοτε κριτήρια κάθε εργασίας στο χρονικό διάστημα ενδιαφέροντος, στη φόρμα αποτελεσμάτων τυπώνεται το αντίστοιχο μήνυμα.

4.2.3 Υποσύστημα Βάσης Δεδομένων

Εφαρμογή διαχείρισης της Βάσης Δεδομένων

Αυτό το υποσύστημα είναι υπεύθυνο κατ' αρχήν για την επικοινωνία με την Βάση Δεδομένων, δηλαδή για την δημιουργία και το κλείσιμο της σύνδεσης με αυτή. Στο σημείο αυτό χειρίζομαι κάποια Exceptions έτσι ώστε να είναι βέβαιο ότι η σύνδεση ανοίγει και κλείνει σωστά. Επίσης, το υποσύστημα είναι υπεύθυνο για την εκτέλεση των SQL ερωτήσεων και για το φόρτωμα των αποτελεσμάτων σε ειδικές δομές. Τα αποτελέσματα των queries επιστρέφονται στη δομή ResultSet η οποία έχει υλοποιηθεί στο πακέτο java.sql. Υλοποιεί έναν πίνακα, όπου οι γραμμές είναι τα records που επέστρεψαν από τη Βάση Δεδομένων, και οι στήλες είναι τα πεδία που ζητήθηκαν να επιστραφούν από το "select" της SQL ερώτησης που εφαρμόστηκε.

Επίσης το υποσύστημα αυτό ευθύνεται για την αποθήκευση του νέου χρήστη του συστήματος αλλά και για την αποθήκευση της πλοήγησής του σε μορφή String. Ακόμα, φροντίζει έτσι ώστε να υπάρχει σωστή αντιστοιχία πλοήγησης↔χρήστη. Το σχήμα της Βάσης Δεδομένων δίνεται στο σχήμα 4.14:

Σχήμα 4.14: Σχήμα της Βάσης Δεδομένων

5

Υλοποίηση

Στο κεφάλαιο αυτό αρχικά αναπτύσσονται οι σημαντικότεροι αλγόριθμοι του συστήματος. Στη συνέχεια δίνονται κάποιες πληροφορίες για τις λεπτομέρειες υλοποίησης του συστήματος με περιγραφή των κλάσεων αυτού. Τέλος, αναφέρονται τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη του συστήματος και δίνονται πληροφορίες για την εγκατάστασή του.

5.1 Βασικοί Αλγόριθμοι

Στην ενότητα αυτή αναπτύσσονται οι σημαντικότεροι αλγόριθμοι που αναπτύχθηκαν στη διπλωματική αυτή εργασία με τη μορφή ψευδοκώδικα. Οι αλγόριθμοι αυτοί είναι οι εξής:

- Ο αλγόριθμος για τον υπολογισμό της δομικής απόστασης μεταξύ δύο συμβολοακολουθιών, τροποποιημένος σε σχέση με αυτόν που περιγράφηκε στην Ενότητα 3.3.
- Ο αλγόριθμος K Μέσων (K Means) προσαρμοσμένος να υπολογίζει αποστάσεις μεταξύ δεδομένων σε μορφή String και όχι μεταξύ διανυσμάτων.
- Ο αλγόριθμος για συσταδοποίηση με βάση την τεχνική «μονός σύνδεσμος». Είναι τροποποιημένος σε σχέση με αυτόν που περιγράφεται στην υποενότητα 3.4.3 και υλοποιείται με χρήση δύο αλγορίθμων: i) Υπολογισμός ελαχίστου συνεκτικού δένδρου ενός γράφου και ii) Εύρεση των συνεκτικών συνιστωσών αυτού.
- Ο αλγόριθμος για υπολογισμό του CIndex
- Ο αλγόριθμος για την εύρεση των πιο αναποφάσιστων χρηστών του συστήματος

5.1.1 Τροποποιημένος αλγόριθμος για Structural Distance

Ο αλγόριθμος αυτός αποτελεί επέκταση του αλγορίθμου που παρουσιάστηκε στην Ενότητα 3.3, και εισάγει εκτός της συντακτικής απόστασης μεταξύ δύο Strings άλλο ένα αριθμητικό μέγεθος, χαρακτηριστικό της μεταξύ τους ομοιότητας, το οποίο σχετίζεται με την ομοιότητα

του περιεχομένου τους. Στη συγκεκριμένη διπλωματική εργασία, ο υπολογισμός της απόστασης μεταξύ δύο Strings είναι θεμελιώδης: Εκτός του ότι αποτελεί και ξεχωριστή εργασία του συστήματος NaviMoz (ο χρήστης δίνει μια πλοήγηση και έναν βαθμό ομοιότητας και το σύστημα επιστρέφει όλες εκείνες τις πλοηγήσεις, των οποίων η δομική απόσταση από τη δοσμένη είναι μικρότερη του βαθμού ομοιότητας-κατηγορία εργασιών 2), αποτελεί επίσης το κριτήριο με βάση το οποίο πραγματοποιούνται οι εργασίες clustering των πλοηγήσεων.

Το νέο αριθμητικό μέγεθος το οποίο εισάγεται εδώ αποτελεί μια ένδειξη του ποσοστού ομοιότητας των δύο συμβολοσειρών. Το ποσοστό αυτό υπολογίζεται ως εξής:

Καταμετρούνται οι εμφανίσεις ιδίων λέξεων (για την εφαρμογή μας, των ιδίων σελίδων) που εμφανίζονται στις δύο πλοηγήσεις (Strings) και το ποσό αυτό διαιρείται με το συνολικό μήκος και των δύο λέξεων. Στη συνέχεια αφαιρείται η τιμή που υπολογίστηκε από το 1, γιατί η σύμβασή μας είναι το 0 να δηλώνει τέλεια ομοιότητα και το 1 τέλεια ανομοιότητα.

Ο αλγόριθμος που παρουσιάζεται παρακάτω υπολογίζει το κόστος μετατροπής της μιας ακολουθίας στην άλλη, ως άθροισμα της κανονικοποιημένης συντακτικής απόστασης και του ποσοστού ομοιότητας, όπως περιγράφηκε παραπάνω. Σαν «λέξη» ορίζεται η κάθε σελίδα μέσα σε μία πλοήγηση. Ο αλγόριθμος εύρεσης της συντακτικής απόστασης υπάρχει στην υποενότητα 3.3 και για χάρη συντομίας παραλείπεται εδώ. Η μορφή του είναι η εξής:

Είσοδος:Δύο String ακολουθίες οι οποίες αντιστοιχούν στις πλοηγήσεις των οποίων θέλουμε να βρούμε το κόστος μετάβασης από τη μία στην άλλη.

Έξοδος:Ενας αριθμός δηλωτικός της ομοιότητας των δύο πλοηγήσεων. Ο αλγόριθμος επιστρέφει 0 αν οι πλοηγήσεις είναι ίδιες και 0 αν είναι τελείως διαφορετικές.

Αλγόριθμος:

-Ο ακέραιος similar είναι ένας μετρητής ο οποίος αρχικοποιείται στο 0 και αυξάνεται κάθε φορά που συναντιέται μια ίδια λέξη μέσα στις πλοηγήσεις.

-Ο ακέραιος total length περιέχει το συνολικό μήκος των δύο πλοηγήσεων(δηλαδή το άθροισμα των λέξεων που περιέχουν)

-Ο Vector v3 περιέχει τις ίδιες λέξεις μέσα στις πλοηγήσεις, από μια φορά την καθεμία.

-Οι Vectors v1, v2 έχουν σαν στοιχεία τους αντίστοιχα τις λέξεις που αποτελούν είσοδο στον αλγόριθμο, μετά τη διαγραφή των «/» μεταξύ των σελίδων.

Ο αλγόριθμος έχει ως εξής:

Υπολόγισε τη συντακτική απόσταση μεταξύ των δύο πλοηγήσεων σύμφωνα με τον αντίστοιχο αλγόριθμο της υποενότητας 3.3

Αποθήκευσε το αποτέλεσμά του στη μεταβλητή Number_of_steps

/*Κανονικοποίηση του αποτελέσματος*/

Διαίρεσε το Number_of_steps με την τιμή της total_length.

Αποθήκευσε το αποτέλεσμα στη μεταβλητή normalization.

/*Δημιουργία του Vector v3*/

Για όλα τα στοιχεία του v1{

Για όλα τα στοιχεία του v2{

Αν το στοιχείο-λέξη του v1 ισούται με το στοιχείο-λέξη του v2{

Αν ο v3 δεν περιέχει το στοιχείο αυτό

Πρόσθεσέ το στον v3

Αλλιώς συνέχισε

}

}

}

/*Υπολογισμός της τιμής similar*/

Για όλα τα στοιχεία του v3{

Για όλα τα στοιχεία του v1 (πρώτη πλοήγηση){

Αν το στοιχείο του v1 ισούται με το στοιχείο του v3 αύξησε την τιμή της μεταβλητής similar κατά 1.

}

Για όλα τα στοιχεία του v2 (δεύτερη πλοήγηση){

Αν το στοιχείο του v2 ισούται με το στοιχείο του v3 αύξησε την τιμή της μεταβλητής similar κατά 1.

}

}

/*Υπολογισμός του συνολικού βαθμού ομοιότητας*/

Διαίρεσε την τιμή της similar με το συνολικό μήκος και των δύο πλοηγήσεων.

Αφαίρεσε το προηγούμενο αποτέλεσμα από το 1.

Πρόσθεσε το αποτέλεσμα με την τιμή της normalization. Διαίρεσέ το με το 2 και αποθήκευσέ το στη μεταβλητή total_result.

Επέστρεψε το total_result.

Παράδειγμα

Ένα παράδειγμα εφαρμογής του αλγορίθμου αυτού δίνεται παρακάτω. Έστω ότι οι είσοδοι είναι οι πλοηγήσεις «/Health/Medicine/Fitness» και «/Health/Fitness/Running/Training/Training/Coaching/Training». Η συντακτική τους απόσταση είναι ίση με 6. Πράγματι, η πρώτη πλοήγηση μπορεί να μετατραπεί στη δεύτερη με τη διαγραφή της λέξης “Medicine” (διαγραφη-κόστος=1) και με την εισαγωγή στο τέλος της των λέξεων “Running”, “Training”, “Training”, “Coaching”, “Training” (5 εισαγωγές-κόστος 5). Οι πράξεις αυτές είναι και ο ελάχιστες δυνατές που μπορούν να γίνουν. Το συνολικό κόστος είναι 6. Η τιμή της normalization είναι 0.6, αφού το μέγεθος των πλοηγήσεων είναι 10. Στη συνέχεια προστίθενται στον v3 οι κοινές λέξεις μεταξύ των πλοηγήσεων, δηλαδή οι “Health” και “Fitness”. Ψάχνοντας για ταυτίσεις των λέξεων της πρώτης πλοήγησης με αυτές του v3 βρίσκουμε 2,και για της δεύτερης βρίσκουμε επίσης 2 (similar=2). Η τιμή λοιπόν που προστίθεται στη normalization είναι η 1-4/10=0.6. Το άθροισμά τους διαιρείται με το 2 κι έτσι τελικά αυτό που επιστρέφει ο αλγόριθμος είναι το 0.6

5.1.2 KMeans με χρήση δομικής απόστασης

Ο αλγόριθμος αυτός αποτελεί βασικό στοιχείο της συγκεκριμένης διπλωματικής εργασίας. Βασίζεται στον αλγόριθμο K μέσων που παρουσιάστηκε στην Ενότητα 3.3, είναι όμως προσαρμοσμένος στα δεδομένα μας (που τα αποτελούν οι πλοηγήσεις των χρηστών), έτσι ώστε να υπολογίζει αποστάσεις μεταξύ στοιχείων τύπου String και όχι μεταξύ σημείων. Οι αποστάσεις αυτές μεταξύ των στοιχείων τύπου String είναι ουσιαστικά οι δομικές τους αποστάσεις, όπως αυτές υπολογίζονται από τον αλγόριθμο Struct_Dist. Η τροποποίηση αυτή υλοποιείται ως εξής: Επιλέγω ως κέντρα των συστάδων ως τα πρότυπα-πλοηγήσεις εκείνα των οποίων η μέση απόσταση από τα υπόλοιπα πρότυπα της ίδιας συστάδας είναι η ελάχιστη. Άρα, το κέντρο μιας συστάδας είναι στην ουσία μία πλοήγηση που ανήκει στη συστάδα αυτή.

Ο αλγόριθμος υπολογίζει τις συστάδες στις οποίες κατηγοριοποιούνται οι πλοηγήσεις. Στο τέλος του εξετάζει αν η προηγούμενη ομαδοποίηση ήταν ίδια με τη νέα και, αν όχι, επαναλαμβάνεται μέχρι να γίνουν ίδιες. Το κριτήριο αυτό όμως δεν είναι και το μόνο κριτήριο τερματισμού του αλγορίθμου, καθώς είναι πολύ συχνή η περίπτωση τα κέντρα κάποιων συστάδων να εναλλάσσονται συνεχώς μεταξύ δύο τιμών, μη επιτρέποντας στον αλγόριθμο να δώσει αποτέλεσμα. Για το λόγο αυτό έχω θέσει ως μέγιστο δυνατό αριθμό επαναλήψεων τις 50. Ο αλγόριθμος έχει την εξής μορφή:

Είσοδος: Ο αριθμός K των συστάδων στις οποίες θα κατηγοριοποιηθούν οι πλοηγήσεις.

Έξοδος: Οι συστάδες των πλοηγήσεων των χρηστών.

Αλγόριθμος:

-Ο Vector patterns έχει ως στοιχεία τις πλοηγήσεις των χρηστών σε μορφή String. Το μέγεθός -του είναι Q, όσες δηλαδή και οι πλοηγήσεις.

-Η boolean μεταβλητή flag αποτελεί ένδειξη του αν ο αλγόριθμος θα επαναληφθεί ή όχι. Έχει την τιμή true αν πρέπει να επαναληφθεί και την τιμή false αν δεν πρέπει. Αρχικοποιείται με true.

-Ο πίνακας Z μεγέθους K αποθηκεύει τα παλιά κέντρα των συστάδων. Αρχικοποιείται με τα K πρώτα πρότυπα που περιέχει ο Vector patterns.

-Ο πίνακας Z_1 μεγέθους K αποθηκεύει τα νέα κέντρα των συστάδων.

-Ο πίνακας G_1 μεγέθους Q, φυλάσσει πληροφορία σχετικά με το σε ποια συστάδα ανήκει το q-οστό πρότυπο, $0 \leq q \leq Q$. Αρχικοποιείται έτσι ώστε το q-οστό στοιχείο του να ισούται με q, πράγμα που σημαίνει ότι αρχικά το q-οστό πρότυπο ανήκει στην q-οστή συστάδα, δηλαδή το κάθε πρότυπο αποτελεί από μόνο του συστάδα.

-Ο βοηθητικός πίνακας G μεγέθους Q, φυλάσσει την προηγούμενη πληροφορία σχετικά με το σε ποια συστάδα ανήκει το q-οστό πρότυπο, $0 \leq q \leq Q$

-Ο πίνακας N, μεγέθους K σε κάθε στοιχείο του k, $0 \leq k \leq K$, έχει πόσες πλοηγήσεις περιέχει η k-οστή συστάδα.

-Ο πίνακας D, μεγέθους K*Q, σε κάθε στοιχείο του D[k][q], $0 \leq k \leq K$, $0 \leq q \leq Q$, περιέχει την απόσταση του q-οστού προτύπου από το κέντρο της k-οστής συστάδας.

Ο αλγόριθμος έχει ως εξής:

Επιλογή των πλοηγήσεων από τη Βάση δεδομένων και αποθήκευσή τους στο Vector patterns.

Υπολογισμός του πίνακα δομικών αποστάσεων των πλοηγήσεων

Όσο υπάρχουν αλλαγές στα κέντρα των συστάδων (αν δηλαδή flag=true){

Για όλα τα πρότυπα q, $0 \leq q \leq Q$

Θέσε $G[q]=G_1[q]$

Αρχικοποίησε όλα τα στοιχεία του πίνακα D με τη μεγαλύτερη δυνατή δομική απόσταση, δηλαδή με την τιμή 1

Αρχικοποίησε όλα τα στοιχεία του πίνακα N με την τιμή 0.

/*Φάση ενημέρωσης του πίνακα D*/

Για όλες τις συστάδες k, $0 \leq k \leq K${

Για όλα τα πρότυπα q, $0 \leq q \leq Q${

Θέσε στο στοιχείο D[k][q] την απόσταση του q-οστού προτύπου από την k-οστή συστάδα.

}

}

Για όλα τα πρότυπα q, $0 \leq q \leq Q$ {

Βρες με την τεχνική Bubblesort τη συστάδα k_{min} εκείνη από την οποία απέχει την ελάχιστη απόσταση σε σχέση με τις υπόλοιπες.

/*Ενημέρωση ότι το τρέχον πρότυπο ανήκει στην k_{min} συστάδα*/

$G_1[q]=k_{min}$

$N[k_{min}]=N[k_{min}]+1$

}

/*Φάση υπολογισμού των νέων κέντρων*/

Για όλες τις συστάδες k, $0 \leq k \leq K$ {

Για όλα τα πρότυπα q, $0 \leq q \leq Q$

Βρες τα πρότυπα που ανήκουν στην k-οστή συστάδα

Για όλα τα πρότυπα q' της k-οστής συστάδας

Υπολόγισε το μέσο όρο των αποστάσεών του από τα υπόλοιπα πρότυπα της ίδιας συστάδας (για την εύρεση του μέσου όρου προστίθενται όλες οι αποστάσεις και διαιρούνται με την τιμή N[k])

Βρες με χρήση της τεχνικής Bubblesort τη μικρότερη απόσταση (δηλαδή το μικρότερο μέσο όρο) και το πρότυπο που αντιστοιχεί στην απόσταση αυτή

Θέσε το πρότυπο αυτό ως κέντρο της k-οστής συστάδας ($Z_1[k]=q'$).

}

/*Φάση καταχώρησης των Q προτύπων στις k συστάδες*/

Για όλα τα πρότυπα q, $0 \leq q \leq Q$ {

Αν ανήκουν στην ίδια συστάδα με την προηγούμενη επανάληψη, αν

Δηλαδή $G_1[q]=G[q]$, θέσε flag=false (οπότε και ο αλγόριθμος σταματά εδώ),

}

Αλλιώς {

/*Ενημέρωση των κέντρων των συστάδων*/

Θέσε τα νέα κέντρα ως παλιά, $Z[k]=Z_1[k]$

Επανέλαβε τη διαδικασία

}

}

/*Φάση ανάκτησης των K συστάδων*/

Για όλες τις συστάδες k, 0≤k≤K{

Για όλα τα πρότυπα q, 0≤q≤Q{

Αν $G_1[q]=k$

Τοποθέτησε σε Vector v1 το q (δημιουργώ νέο v1για κάθε συστάδα)

}

Πρόσθεσε στο Vector v το Vector v1 (Πρόσθεσε δηλαδή τη συστάδα που μόλις δημιούργησες)

}

Επέστρεψε το Vector v, δηλαδή το Vector εκείνο του οποίου στοιχεία είναι οι συστάδες

Παράδειγμα

Έστω ότι ζητείται η ταξινόμηση των παρακάτω πλοηγήσεων:

- /Arts/Photography/Music/Arts/Dancing
- /Arts/Photography/Music
- /Music/Expositions//Photography/ Expositions/Photography
- /Sculpture/ Expositions//Photography/ Expositions/Photography

Έστω ότι σαν είσοδος στον αλγόριθμο για το πλήθος των επιθυμητών συστάδων είναι 2. Ο αλγόριθμος πραγματοποιεί τρεις επαναλήψεις. Στην πρώτη επανάληψη οι συστάδες δημιουργούνται ως εξής:

1) [/Arts/Photography/Music/Arts/Dancing]
2) [/Arts/Photography/Music,
/Music/Expositions/Photography/Expositions/Photography,
/Sculpture/Expositions/Photography/Expositions/Photography]

Στη δεύτερη επανάληψη παρατηρείται η μετακίνηση της πλοήγησης /Arts/Photography/Music από τη δεύτερη συστάδα στην πρώτη:

1) [/Arts/Photography/Music/Arts/Dancing, /Arts/Photography/Music]
2) [/Music/Expositions/Photography/Expositions/Photography,
/Sculpture/Expositions/Photography/Expositions/Photography]

Τέλος, στην τρίτη επανάληψη οι τελικές συστάδες που λαμβάνουμε δε διαφέρουν από αυτές του δεύτερου επιπέδου:

1) [/Arts/Photography/Music/Arts/Dancing, /Arts/Photography/Music]
2) [/Music/Expositions/Photography/Expositions/Photography,
/Sculpture/Expositions/Photography/Expositions/Photography]

5.1.3 Αλγόριθμος υλοποίησης της τεχνικής «μονός σύνδεσμος»

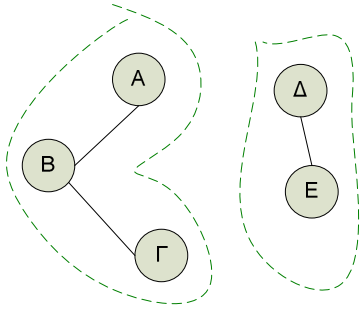
Στη διπλωματική αυτή εργασία, έχω υλοποιήσει τον αλγόριθμο για την εύρεση των συστάδων με την τεχνική «μονός σύνδεσμος» βασιζόμενη στη μεθοδολογία που προτείνει η Ellen Voorhees στη διδακτορική διατριβή της [Voo85]. Αρχικά υπολογίζω το Ελάχιστο Συνεκτικό Δένδρο (ΕΣΔ) του γράφου που αντιστοιχεί στις πλοηγήσεις των χρηστών και στη συνέχεια βρίσκω τις συνεκτικές συνιστώσες αυτού, δοθέντος κάποιου επιπέδου συσταδοποίησης, οι οποίες αποτελούν τις συστάδες για το δοσμένο επίπεδο.

Πιο αναλυτικά, στην εργασία [Voo85], ο αλγόριθμος που χρησιμοποιείται για να δημιουργήσει την ιεραρχία αποτελεί εφαρμογή του αλγορίθμου του Prim για την εύρεση του Μεγίστου Συνεκτικού Δένδρου (ΜΣΔ) ενός γράφου. Αυτό συμβαίνει γιατί οι υπολογισμοί πραγματοποιούνται πάνω σε έναν πίνακα ομοιοτήτων, οπότε, όσο μεγαλύτερη είναι η ομοιότητα μεταξύ δύο συστάδων, τόσο πιο «κοντινά» θεωρούνται και τείνουν να συνενωθούν σε ένα. Στη διπλωματική μου εργασία οι υπολογισμοί πραγματοποιούνται με βάση το N*N πίνακα δομικών αποστάσεων των πλοηγήσεων, όπως υπολογίζονται από τον τροποποιημένο αλγόριθμο για την εύρεση δομικής απόστασης (5.1.1). Για το λόγο αυτό, έχω τροποποιήσει τον αλγόριθμο της Voorhees έτσι ώστε να εφαρμόζεται στα δεδομένα μου. Βρίσκω λοιπόν το Ελάχιστο Συνεκτικό Δένδρο (ΕΣΔ) του γράφου G που αντιστοιχεί στις πλοηγήσεις των χρηστών. Αρχικά επιλέγεται τυχαία ένα δεδομένο από τα N και τοποθετείται στην ιεραρχία. Στη συνέχεια αρχικοποιείται ένας πίνακας με τις αποστάσεις του δεδομένου αυτού από τα υπόλοιπα N-1 δεδομένα. Αν κάποιο δεδομένο i δεν είναι στο ΕΣΔ, η i-οστή είσοδος του πίνακα θα περιέχει τις μικρότερες των αποστάσεων μεταξύ του δεδομένου i και όλων των δεδομένων του ΕΣΔ. Το δεδομένο d, από το οποίο η απόσταση του i-οστού είναι η μικρότερη δυνατή, προστίθεται στο ΕΣΔ και η i-οστή είσοδος του πίνακα ενημερώνεται με το ελάχιστο των εξής δύο τιμών: της απόστασης των δεδομένων i και d, και της τρέχουσας τιμής της. Η διαδικασία αυτή επαναλαμβάνεται μέχρι να τοποθετηθούν όλα τα δεδομένα στο ΕΣΔ.

Έχει αποδειχτεί [GR69] ότι ένα ΕΣΔ περιέχει όλη την απαραίτητη πληροφορία για να εκτελεστεί ο αλγόριθμος συσταδοποίησης με την τεχνική «μονός σύνδεσμος». Το ΕΣΔ όπως υπολογίστηκε παραπάνω, έχει τόσους κόμβους όσα είναι τα δεδομένα μου (πλοηγήσεις των χρηστών), οι οποίοι συνδέονται μεταξύ τους με ακμές με βάρη. Τα βάρη αυτά αντιστοιχούν στις ελάχιστες δυνατές αποστάσεις μεταξύ των κόμβων που συνδέουν. Δοθέντος ενός επιπέδου συσταδοποίησης l_1, οι συστάδες που προκύπτουν για το επίπεδο αυτό είναι οι συνεκτικές συνιστώσες του ΕΣΔ αν διαγραφούν όλες οι ακμές με βάρος $w \geq l_1$ [DCS+].

Έστω για παράδειγμα 5 σημεία, Α, Β, Γ, Δ, Ε και οι μεταξύ τους αποστάσεις στο ΕΣΔ:

(Α, Β)=1, (Β, Γ)=2, (Α, Δ)=4, (Δ, Ε)=1, (Γ, Ε)=3. Χρησιμοποιώντας ως επίπεδο συσταδοποίησης το l=2, κλαδεύονται όλες οι ακμές που έχουν βάρος μεγαλύτερο ή ίσο του 2. Έτσι, κλαδεύονται οι (Α, Δ)=4 και (Γ, Ε)=3. Ο γράφος που προκύπτει είναι ο εξής:



Σχήμα 5.1 Συνεκτικές συνιστώσες ενός Ελαχίστου Συνεκτικού Δένδρου (ΕΣΔ)

Το αποτέλεσμα αυτό μεταφράζεται ως εξής: Για το επίπεδο συσταδοποίησης 2, λαμβάνονται 2 συστάδες, οι (Α, Β, Γ) και η (Δ, Ε).

Στη συνέχεια παρατίθενται οι αλγόριθμοι MST και ConnectedComponents. Ο πρώτος υπολογίζει το ΕΣΔ ενός γράφου ενώ ο δεύτερος βρίσκει τις συνεκτικές συνιστώσες ενός ΕΣΔ βασιζόμενος σε αντίστοιχο αλγόριθμο που περιγράφεται στο [CLR+01].

MST

Είσοδος:Οι ημερομηνίες έναρξης και λήξης εντός των οποίων οι πλοηγήσεις θα εξεταστούν, οι πλοηγήσεις των χρηστών σε μορφή String για τις ημερομηνίες αυτές και ο πίνακας τιμών double των μεταξύ τους αποστάσεων.

Έξοδος: Το ΕΣΔ που αντιστοιχεί στις πλοηγήσεις αυτές.

Αλγόριθμος:

Στην περιγραφή του αλγορίθμου, με τον όρο «κόμβος» αναφέρομαι στην πλοήγηση του χρήστη.

-Ο Vector patterns περιέχει τις πλοηγήσεις των χρηστών σε μορφή String.

-Ο Vector hierarchy αντιστοιχεί στο ΕΣΔ. Το ΕΣΔ επιστρέφεται με τη μορφή του Vector αυτού, ο οποίος αποτελείται από τριπλέτες. Η κάθε τριπλέτα περιέχει στο πρώτο πεδίο έναν κόμβο-πλοήγηση, στο δεύτερο πεδίο τον «πατέρα» του κόμβου αυτού και στο τρίτο πεδίο τη μεταξύ τους απόσταση. Ο «πατέρας» ενός κόμβου k είναι αυτός ο κόμβος από τον οποίο απέχει τη μικρότερη απόσταση συγκριτικά με τους υπολοίπους, και εξαιτίας του οποίου ο k εισήχθηκε στο ΕΣΔ.

-Ο πίνακας τιμών double A, μεγέθους N*N, είναι ο πίνακας αποστάσεων μεταξύ των πλοηγήσεων.

-Ο πίνακας Entry αποτελείται από δομές τύπου Info. Κάθε τέτοια δομή έχει τέσσερα πεδία:

Το πρώτο πεδίο περιέχει την τιμή του κόμβου (τον αριθμό του προτύπου), το δεύτερο περιέχει την απόσταση sim που του αντιστοιχεί (δηλαδή την απόσταση του από κάποιον κόμβο-πατέρα με την οποία εισήχθηκε στο ΕΣΔ), το τρίτο είναι μία τιμή Boolean και είναι true αν ο κόμβος έχει εισαχθεί στο ΕΣΔ, ενώ αν όχι είναι false, και το τέταρτο πεδίο είναι ο κόμβος-πατέρας.

-Ο Current_Did είναι ο κόμβος που τοποθετείται στο ΕΣΔ (τρέχων κόμβος) .

-Ο Next_Did είναι ο επόμενος κόμβος που θα εισαχθεί στο ΕΣΔ.

-Η double μεταβλητή MinSim είναι η ελάχιστη απόσταση από το ΕΣΔ.

Ο αλγόριθμος έχει ως εξής:

Για μη μηδενική είσοδο πλοηγήσεων{

Αρχικοποίησε όλα τα στοιχεία του πίνακα Entry με τιμές: αριθμός προτύπου από 0 μέχρι N-1, 1, false, -1). Αρχικά δηλαδή κανένας κόμβος δεν ανήκει στο ΕΣΔ.

Current_Did=0

Όσο Current_Did≠-1{

Εισήγαγε στο ΕΣΔ τον τρέχοντα κόμβο.

Υπολόγισε τις αποστάσεις του από τους υπόλοιπους κόμβους.

Θέσε τη MinSim ίση με 1.

Για όλους τους κόμβους 0≤k≤N που δεν είναι στο ΕΣΔ{

Αν η απόσταση sims του κόμβου k από τον τρέχοντα κόμβο είναι μικρότερη από την τιμή στο αντίστοιχο πεδίο της απόστασης του πίνακα Entry για τον κόμβο αυτό (πεδίο sim){

Ενημέρωσε το sim του κόμβου k με την τιμή sims

Θέσε ως πατέρα του k-οστού κόμβου τον Current_Did

}

Αν η τιμή sim του κόμβου k είναι < MinSim{

Θέσε MinSim←sim[k]

Θέσε Next_Did←k

}

}

Εφόσον υπάρχει τιμή για το Next_Did{

/*Εισαγωγή του Next_Did στο ΕΣΔ με κλήση της InsertHierarchy()*/

Πρόσθεσε στο Vector hierarchy μια τριπλέτα της μορφής:

(Next_Did, ο πατέρας του Next_Did, MinSim)

}

}

Παράδειγμα

Έστω ότι ζητείται η εύρεση του Ελαχίστου Συνεκτικού Δένδρου των παρακάτω πλοηγήσεων:

- /Arts/Photography/Music/Arts/Dancing
- /Arts/Photography/Music
- /Music/Expositions//Photography/ Expositions/Photography
- /Sports/Multi-Sports/Triathlon/Riding

Αρχικά τοποθετείται στο δένδρο η πρώτη πλοήγηση /Arts/Photography/Music/Arts/Dancing. Υπολογίζονται οι δομικές αποστάσεις της από τις υπόλοιπες. Από τη δεύτερη πλοήγηση απέχει 0.1875, από την τρίτη απέχει 0.545 και από την τέταρτη 0.778. Έτσι πρώτη στο ΕΣΔ θα εισαχθεί η δεύτερη πλοήγηση /Arts/Photography/Music, επειδή αυτή απέχει λιγότερο από την πλοήγηση που ήδη υπάρχει στο δένδρο. Η αντίστοιχη τριπλέτα η οποία περιγράφει το ΕΣΔ είναι η [1, 0, 0.1875].

Στη συνέχεια, η τρέχουσα πλοήγηση από την οποία εξετάζονται οι δομικές αποστάσεις είναι αυτή που εισήχθηκε τελευταία στο δένδρο, δηλαδή η /Arts/Photography/Music. Εξετάζονται οι αποστάσεις της από τις υπόλοιπες δύο που δεν ανήκουν στο δένδρο. Από την πλοήγηση /Music/Expositions//Photography/ Expositions/Photography απέχει 0.4375 και από τη /Sports/Multi-Sports/Triathlon/Riding απέχει 0.7857. Έτσι, η επόμενη πλοήγηση που εισάγεται στο δένδρο είναι η /Music/Expositions//Photography/ Expositions/Photography, αφού αυτή απέχει λιγότερο. Η αντίστοιχη τριπλέτα η οποία περιγράφει το ΕΣΔ είναι η

[2, 1, 0.4375].

Τελευταία εισάγεται η πλοήγηση /Sports/Multi-Sports/Triathlon/Riding. Τώρα πια η τρέχουσα πλοήγηση από την οποία εξετάζονται οι αποστάσεις είναι η /Music/Expositions//Photography/ Expositions/Photography. Η /Sports/Multi-Sports/Triathlon/Riding απέχει από αυτή 0.778. Η αντίστοιχη τριπλέτα που εισάγεται στο ΕΣΔ είναι η [3, 0, 0.778].

ConnectedComponents

Είσοδος: :Οι ημερομηνίες έναρξης και λήξης εντός των οποίων οι πλοηγήσεις θα εξεταστούν και μια double μεταβλητή που αντιστοιχεί στο επίπεδο συσταδοποίησης.

Έξοδος: Οι συνεκτικές συνιστώσες ενός ΕΣΔ, ή αλλιώς οι συστάδες για το δοθέν επίπεδο συσταδοποίησης.

Αλγόριθμος:

-Ο Vector Initial_Clusters περιλαμβάνει τις αρχικές συστάδες. Αποτελείται από δομές τύπου Komvos δηλαδή από τριπλέτες της εξής μορφής:

Το πρώτο πεδίο (vertex) δηλώνει τον κόμβο και έχει τιμή int.

Το δεύτερο πεδίο (findset) δηλώνει έναν κόμβο με τον οποίο ο vertex (το πρώτο πεδίο) -βρίσκονται στην ίδια συνεκτική συνιστώσα.

Το τρίτο πεδίο (inset) είναι μια τιμή boolean και δηλώνει αν ο vertex έχει τοποθετηθεί σε κάποια συστάδα. Ο Vector Initial_Clusters αρχικοποιείται για όλους τους κόμβους k, 0≤k≤N, με τιμές (k, k, false).

-Ο Vector Clustered_Nodes είναι ουσιαστικά αυτό που επιστρέφει ο αλγόριθμος, δηλαδή οι συνεκτικές συνιστώσες του ΕΣΔ. Αποτελείται από Vectors, ο καθένας εκ των οποίων αντιστοιχεί στις συστάδες. Οι κόμβοι σε κάθε συστάδα είναι κωδικοποιημένοι ως int.

-Ο Vector hierarchy είναι το αποτέλεσμα του αλγορίθμου MST (το ΕΣΔ με τη μορφή τριπλετών)

Ο αλγόριθμος έχει ως εξής:

```
Για μη μηδενικό πλήθος προτύπων{
    /*Για όλες τις ακμές Ε του ΕΣΔ */
    Για όλα τα στοιχεία του Vector hierarchy{
        Αν η τιμή τους είναι μικρότερη από το επίπεδο συσταδοποίησης{
            Αν οι δύο κόμβοι που δημιουργούν την ακμή Ε έχουν διαφορετικό
            findset, θέσε την ίδια τιμή στο πεδίο findset
                                                        }
        Αλλιώς συνέχισε
                                                    }
    /*Τοποθέτηση των κόμβων με το ίδιο findset στην ίδια συστάδα*/
    Για όλα τα στοιχεία i, 0≤i≤N, του Vector Initial_Clusters {
        Αν δεν έχουν τοποθετηθεί σε κάποια συστάδα (inset=false){
            Για όλα τα στοιχεία j, 0≤j≤N, του Vector Initial_Clusters{
                Αν findset(i)=findset(j){
                    Τοποθέτησέ τους κόμβους i και j στην ίδια συστάδα
                    Πρόσθεσε τη συστάδα αυτή στο Vector
Clustered_Nodes
                                                    }
                Αλλιώς συνέχισε
```

```
                    }
                }
            }
        }
```

Επέστρεψε το Vector Clustered_Nodes.

Παράδειγμα

Χρησιμοποιώντας τις πλοηγήσεις :

- /Arts/Photography/Music/Arts/Dancing
- /Arts/Photography/Music
- /Music/Expositions//Photography/ Expositions/Photography
- /Sports/Multi-Sports/Triathlon/Riding,

με βάση τις οποίες δημιουργήθηκε το ΕΣΔ του προηγούμενου αλγορίθμου, και δίνοντας ως επίπεδο συσταδοποίησης τον αριθμό 0.7, ο παρών αλγόριθμος, «κόβοντας» τους κόμβους των οποίων η δομική απόσταση από το ΕΣΔ είναι μεγαλύτερη του 0.7, δημιουργεί τις εξής συστάδες:

[1, 2, 0] και [3] με τη μορφή κόμβων και

[/Arts/Photography/Music,/Music/Expositions//Photography/Expositions/Photography, /Arts/Photography/Music/Arts/Dancing] και [/Sports/Multi-Sports/Triathlon/Riding] με τη μορφή πλοηγήσεων.

5.1.4: Αλγόριθμος για αναποφάσιστους

Ο αλγόριθμος αυτός είναι ένας νέος αλγόριθμος ο οποίος προτείνεται σ' αυτή τη διπλωματική εργασία, γι' αυτό και αποτελεί βασικό στοιχείο αυτής. Κατατάσσεται στην τρίτη κατηγορία εργασιών του συστήματος (*ερωτήσεις εξόρυξης γνώσης και ομαδοποίησης των χρηστών)*. Στη διπλωματική εργασία έχω υποθέσει ότι ο αριθμός των Back και Forward που εκτελεί ένας χρήστης είναι μια ένδειξη της αναποφασιστικότητάς του. Για παράδειγμα, ο χρήστης που έχει πραγματοποιήσει την πλοήγηση /Arts/Music/Pop/Music/Pop/Concerts/Pop/Music/Rock/Concerts είναι πιο αναποφάσιστος από αυτόν που έχει πραγματοποιήσει /Arts/Music/Rock/Concerts. Έτσι υλοποίησα τον αλγόριθμο Undecided ο οποίος υπολογίζει πόσα Back και Forward εκτελεί ένας χρήστης ακολουθώντας κάποιο μονοπάτι (πλοήγηση).

Καταρχήν πρέπει να διευκρινιστεί τι εννοούμε όταν λέμε Back και Forward. Ο όρος αυτός περιγράφει μια ιδιότητα που έχουν δύο κόμβοι, όπου στην προκειμένη περίπτωση κόμβος είναι μία σελίδα που περιέχεται σε μια πλοήγηση χρήστη. Για παράδειγμα, στο μονοπάτι a/

/b/c/b/a/, οι δύο κόμβοι a και οι δύο κόμβοι b έχουν την ιδιότητα Back και Forward. Για να είναι λοιπόν δύο κόμβοι a και b Back και Forward πρέπει να διαθέτουν τα εξής χαρακτηριστικά:

1) label (a)=label(b). Πρέπει δηλαδή πρωτίστως οι ετικέτες τους να είναι ίδιες. Στην περίπτωσή μας πρέπει δύο σελίδες να είναι ίδιες.
2) Μεταξύ τους πρέπει να υπάρχει μονός αριθμός από κόμβους.
3) Τα labels των μεταξύ τους κόμβων πρέπει να είναι διάφορα από τα label(a) και label(b).
4) label(a+1)= label(b-1)

Με βάση τα παραπάνω, στοιχειώδες Back και Forward είναι αυτό που πραγματοποιείται μεταξύ δύο κόμβων οι οποίοι είναι ίδιοι και μεταξύ τους παρεμβάλλεται μόνο ένας κόμβος, διάφορος αυτών (έχουμε δηλαδή τη μορφή A/ B/ A). Γι' αυτό και ο αλγόριθμος αυτός βρίσκει πρώτα αυτά τα Back και Forward. Για να βρει όλα τα υπόλοιπα βασίζεται στην αναδρομή: Για να βρει αν δυο κόμβοι a, b είναι Back και Forward πρέπει οι δύο διπλανοί τους a+1, b-1 είτε να είναι Back και Forward, είτε να σχηματίζουν μια αλυσίδα από διαδοχικά Back και Forward. Ο αλγόριθμος έχει την εξής μορφή:

Είσοδος: Το μονοπάτι που ακολούθησε ο χρήστης με μορφή string.

Έξοδος: Ένας ακέραιος αριθμός που δηλώνει πόσοι κόμβοι με την ιδιότητα Back και Forward υπάρχουν στο μονοπάτι.

Αλγόριθμος:

-Ο Vector v έχει σαν στοιχεία του τις σελίδες της πλοήγησης.

-Ο πίνακας BF είναι ένας πίνακας μεγέθους όσο το πλήθος των λέξεων της πλοήγησης και το κάθε στοιχείο του περιέχει τον κόμβο με τον οποίο αυτό είναι Back και Forward, αλλιώς περιέχει το 0. Ο πίνακας αυτός αρχικοποιείται με όλα του τα στοιχεία 0.

-Ο Vector Possible περιέχει τους κόμβους που είναι υποψήφιοι για Back & Forward. Πιο συγκεκριμένα, αποτελείται από Vectors τριών στοιχείων που έχουν στην πρώτη θέση τον πρώτο κόμβο, στη δεύτερη το δεύτερο και στην τρίτη τη διαφορά των θέσεων των δύο αυτών κόμβων μέσα στην πλοήγηση.

-Ο int counter περιέχει τον αριθμό των Back και Forward.

Ο αλγόριθμος έχει ως εξής:

/*Δημιουργία ενός Hash table χαρακτηριστικού της πλοήγησης-κλήση της μεθόδου Hashes */

Για κάθε στοιχείο-κόμβο του v{

Αν δεν υπάρχει στο Hash table{

τοποθέτησέ τον και δημιούργησε τη λίστα του, valueVector, στην οποία τοποθέτησε τη θέση όπου τον συνάντησες, αλλιώς{

Αν υπάρχει στο Hash table τοποθέτησε στη λίστα του τη θέση όπου τον συνάντησες.

}

}

}

/*Εύρεση των στοιχειωδών Back και Forward-κλήση της μεθόδου FirstSearch*/

Για κάθε key του Hash table {

Αν το μέγεθος του valueVector είναι 1συνέχισε, αλλιώς

Αν το μέγεθος του valueVector είναι >1{

Αν δύο διαδοχικά στοιχεία του valueVector έχουν διαφορά 1 συνέχισε, αλλιώς{

Αν δύο διαδοχικά στοιχεία του valueVector έχουν διαφορά 2{

ενημέρωσε τον πίνακα BF για την εύρεση αυτού του Back και Forward, αλλιώς{

Αν δύο διαδοχικά στοιχεία του valueVector έχουν μεταξύ τους μονό αριθμό στοιχείων τοποθέτησέ τα στο Vector Possible, αλλιώς {

συνέχισε

}

}

}

}

}

Ταξινόμησε το Vector Possible με χρήση της μεθόδου Bubblesort , έτσι ώστε μπροστά-μπροστά να έχεις τους υποψήφιους για Back και Forward κόμβους με το μικρότερο αριθμό μεταξύ τους στοιχείων.

/* Εύρεση των υπολοίπων Back και Forward-κλήση της μεθόδου Algo*/

Για κάθε ζευγάρι κόμβων a, b του ταξινομημένου Vector Possible {

Έλεγξε αν οι διπλανοί του κόμβοι a+1, b-1αποτελούν ζεύγος στον πίνακα BF. Αν ναι, ενημέρωσε τον BF ότι και οι a, b είναι Back και Forward, αλλιώς

/*κλήση της μεθόδου Algo2-η διαδικασία είναι σχεδόν ίδια με της Algo αλλά εδώ έχουμε αναφορές μόνο στον πίνακα BF*/

Έλεγξε το στοιχείο με το οποίο αποτελεί Back και Forward ο κόμβος a+1{

Αν είναι το b-1 ενημέρωσε τον πίνακα BF, αλλιώς{

Έλεγξε το στοιχείο με το οποίο είναι Back και Forward το BF[a+1]

}

}

}

/*Υπολογισμός των Back και Forward*/

Διέτρεξε τον πίνακα BF. Για κάθε μη μηδενικό στοιχείο του, αύξησε τον counter κατά 1

Επέστρεψε τον counter.

Στη συνέχεια παρατίθεται ένα παράδειγμα εφαρμογής του αλγορίθμου αυτού. Έστω λοιπόν ότι δέχεται σαν είσοδο το String "r/s/m/d/m/o/m/s/f/s/f/w/f/w".

Ο Hashtable που θα δημιουργηθεί θα έχει τη μορφή `m→2,4,6`

`w→11,13`

`s→1,7,9`

`r→0`

`f→8,10,12`

`o→5`

`d→3`

Ο αλγόριθμος θα ενημερώσει μετά από εξέταση του Hashtable τον πίνακα BF για τα στοιχειώδη Back και Forward που όντως πραγματοποιούνται μεταξύ των στοιχείων m,w και f γιατί οι δείκτες τους απέχουν κατά 2 και πληρούνται οι προϋποθέσεις για Back και Forward.Έτσι, σε αυτό το στάδιο έχουμε τους εξής Back και Forward κόμβους: (2,4),(4,6),(11,13),(7,9),(8,10),(10,12). Επίσης τοποθετεί στο vector Possible τα πιθανά Back και Forward μεταξύ των στοιχείων s. Απορρίπτει τα στοιχεία r,o και d από πιθανά Back και Forward (που όντως έτσι είναι) αφού η λίστα τους έχει μόνο ένα στοιχείο.

Στη συνέχεια εξετάζει αν τα στοιχεία s στις θέσεις 1 και 7 αποτελούν Back και Forward. Για το λόγο αυτό, εξετάζει με χρήση του πίνακα BF αν τα στοιχεία στις θέσεις 2 και 6 αποτελούν Back και Forward. Επειδή δεν αποτελούν καλείται η μέθοδος Algo2, η οποία βρίσκει ότι το στοιχείο 2 είναι Back και Forward με το 4 και το 4 με τη σειρά του είναι Back και Forward με το 6. Μέσω αυτής της αλυσίδας από Back και Forward, ο αλγόριθμος βρίσκει ότι και τα δύο στοιχεία s αποτελούν Back και Forward, που όντως έτσι είναι.

5.2 Λεπτομέρειες Υλοποίησης

Στην ενότητα αυτή, αρχικά περιγράφεται ο τρόπος αρχικοποίησης του συστήματος, με τη δημιουργία της Βάσης Δεδομένων, και στη συνέχεια περιγράφονται οι κλάσεις με τα πεδία και τις μεθόδους τους.

5.2.1 Αρχικοποίηση συστήματος

Το αποθηκευτικό μέσο του συστήματος είναι η Βάση Δεδομένων NaviMoz στο σύστημα της MySQL. Όπως έχει εξηγηθεί αναλυτικά και σε προηγούμενο κεφάλαιο, για κάθε νέο χρήστη που εγγράφεται στο σύστημα κρατούνται η σειρά εγγραφής του (αύξων αριθμός χρήστη-uid), το όνομά του, το όνομα χρήστη (Username), ο κωδικός χρήστη (Password) και η ηλεκτρονική του διεύθυνση. Οι πληροφορίες αυτέ φυλάσσονται στον πίνακα users. Για κάθε πλοήγηση που πραγματοποιεί ένας χρήστης, εισάγεται στη Βάση Δεδομένων ο αύξων αριθμός της (sid), ο οποίος αποτελεί κλειδί για τον πίνακα session. Στον πίνακα αυτό επίσης φυλάσσονται οι χρονικές στιγμές έναρξης και λήξης της πλοήγησης καθώς και η πλοήγηση σε κατάλληλη μορφή, όπως έχει εξηγηθεί σε προηγούμενο κεφάλαιο. Τέλος, για την αντιστοίχηση μιας πλοήγησης με το χρήστη που την έχει πραγματοποιήσει, ο πίνακας session διαθέτει ένα ξένο κλειδί που αναφέρεται στον πίνακα users. Παρακάτω δίνεται ο SQL κώδικας που δημιουργεί τη Βάση Δεδομένων. Αυτή δημιουργείται μια φορά στην αρχή, από τη γραμμή εντολών.

```
CREATE DATABASE NaviMoz;
CREATE TABLE USERS(
      uid int primary key auto_increment,
      fullname VARCHAR(60) NOT NULL,
      username varchar (30) not null,
      password varchar (20) not null,
      email varchar (100) not null);
CREATE TABLE SESSION(
      uid int references users(uid),
      sid int primary key auto_increment,
      arxh timestamp default current_timestamp,
      telos timestamp,
      pattern varchar(10000));
```

5.2.2 Περιγραφή Κλάσεων

Στην ενότητα αυτή παρατίθενται οι κλάσεις που υλοποιούν τις εφαρμογές του συστήματος. Για κάθε κλάση παρουσιάζονται τα πεδία και οι μέθοδοί της, καθώς επίσης και μια σύντομη περιγραφή της λειτουργίας της και της λειτουργίας των μεθόδων της.

5.2.2.1 public Class UserLogin extends JFrame

Η κλάση αυτή είναι υπεύθυνη για την είσοδο ενός υπάρχοντος χρήστη στο σύστημα NaviMoz, μετά από έλεγχο εγκυρότητας των στοιχείων του.

Πεδία

- `private JLabel label1`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JLabel label11`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JLabel label2`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JLabel label3`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JLabel username`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JLabel password`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JButton ok`
  Το κουμπί OK
- `private JButton cancel`
  Το κουμπί Cancel
- `private JButton register`
  Το κουμπί Register Now
- `private JTextField textField1`
  Το πεδίο εισαγωγής του ονόματος χρήστη (username)
- `private JPasswordField pass`
  Το πεδίο εισαγωγής του κωδικού χρήστη (password)
- `private static boolean flag`
  Βοηθητική μεταβλητή η οποία αρχικοποιείται σε false και γίνεται true αν ο χρήστης είναι εγγεγραμμένος στο σύστημα.
- `private static ConnectSQL con`
  Αντικείμενο της κλάσης ConnectSQL που είναι υπεύθυνη για τη σύνδεση της εφαρμογής με τη Βάση Δεδομένων.
- `public static String a`
  Μεταβλητή στην οποία κρατείται το uid του χρήστη.
- `private class Handler3 implements ActionListener`

Κλάση υπεύθυνη για την ανταπόκριση στις ενέργειες του χρήστη

Μέθοδοι

- `void buildConstraints(GridBagConstraints gbc, int gx, int gy, int gw, int gh, int wx, int wy)`
  Μέθοδος για τη δημιουργία του πλέγματος GridBagLayout που χρησιμοποιείται
- `public void actionPerformed(ActionEvent event)`
  Μέθοδος για το χειρισμό των ενεργειών του χρήστη. Ανοίγει και κλείνει τη σύνδεση με τη Βάση Δεδομένων. Επίσης, στη μέθοδο αυτή ελέγχεται αν ο χρήστης είναι εγγεγραμμένος στο σύστημα.
- `public UserLogin()`
  Κατασκευαστής αντικειμένων της κλάσης

5.2.2.2 public Class NewMember extends JDialog

Η κλάση αυτή είναι υπεύθυνη για τη δημιουργία ενός νέου χρήστη και την εισαγωγή του στο σύστημα NaviMoz. Καλείται από την κλάση UserLogin γι' αυτό και είναι τύπου JDialog: Γιατί όταν καλείται «κλειδώνει» το παράθυρο που έχει ανοίξει η UserLogin, δηλαδή ο χρήστης δεν μπορεί να επιστρέψει στην προηγούμενη φόρμα (πράγμα λογικό, αφού δεν μπορεί να τη χρησιμοποιήσει αν πρώτα δε συμπληρώσει τη δεύτερη φόρμα ώστε να γραφτεί στο σύστημα).

Πεδία

- `private JLabel label1`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JLabel label2`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JLabel label3`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JLabel label4`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JLabel label5`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JLabel label6`
  Βοηθητική ετικέτα για την απεικόνιση πληροφοριακού κειμένου
- `private JButton ok`
  Το κουμπί OK
- `private JButton cancel`
  Το κουμπί Cancel
- `private static ConnectSQL con`
  Αντικείμενο της κλάσης ConnectSQL που είναι υπεύθυνη για τη σύνδεση της εφαρμογής με τη Βάση Δεδομένων.

- `private JTextField tname1`
  Το πεδίο εισαγωγής του μικρού ονόματος του χρήστη
- `private JTextField tname2`
  Το πεδίο εισαγωγής του επωνύμου του χρήστη
- `private JTextField tname3`
  Το πεδίο εισαγωγής του ονόματος χρήστη (username)
- `private JTextField tname4`
  Το πεδίο εισαγωγής του κωδικού χρήστη (password)
- `private JPasswordField tfpass`
  Το πεδίο εισαγωγής του κωδικού χρήστη (password)
- `public static String a`
  Μεταβλητή στην οποία κρατείται το uid του χρήστη.
- `private static ConnectSQL con`
  Αντικείμενο της κλάσης ConnectSQL που είναι υπεύθυνη για τη σύνδεση της εφαρμογής με τη Βάση Δεδομένων.
- `private static boolean flag`
  Βοηθητική μεταβλητή η οποία αρχικοποιείται σε false και γίνεται true αν το όνομα χρήστη ή ο κωδικός χρήστη χρησιμοποιούνται από κάποιον άλλο χρήστη του συστήματος.
- `private static UserLogin parent2`
  Δηλώνει τον πατέρα του JDialog τρέχοντος αντικειμένου, ο οποίος είναι στιγμιότυπο της κλάσης UserLogin.

Μέθοδοι

- `void buildConstraints(GridBagConstraints gbc, int gx, int gy, int gw, int gh, int wx, int wy)`
  Μέθοδος για τη δημιουργία του πλέγματος GridBagLayout που χρησιμοποιείται
- `public void actionPerformed(ActionEvent event)`
  Μέθοδος για το χειρισμό των ενεργειών του χρήστη. Ανοίγει και κλείνει τη σύνδεση με τη Βάση Δεδομένων. Επίσης, στη μέθοδο αυτή ελέγχεται αν ο χρήστης έχει συμπληρώσει όλα τα πεδία και αν το όνομα χρήστη ή ο κωδικός χρήστη που καταχωρεί χρησιμοποιούνται ήδη από κάποιον άλλο χρήστη.
- `public NewMember(UserLogin parent,String title,boolean modal)`
  Κατασκευαστής αντικειμένων της κλάσης. Το πεδίο parent δηλώνει τον πατέρα του Dialog Box που αντιστοιχεί στην κλάση NewMember, το title δηλώνει τον τίτλο του Dialog Box και η μεταβλητή modal δηλώνει αν το τρέχον παράθυρο κλειδώνει το παράθυρο που την κάλεσε. Στην εφαρμογή αυτή παίρνει την τιμή true (δηλαδή κλειδώνει).
- `protected void processWindowEvent(WindowEvent e)`
  Μέθοδος υπεύθυνη για το κλείσιμο του παραθύρου

5.2.2.3 public Class UserLoginBrowser

Η κλάση αυτή είναι υπεύθυνη για την πλοήγηση του ήδη εγγεγραμμένου στο σύστημα χρήστη και για την αποθήκευση της πλοήγησής του στη Βάση Δεδομένων, αφού πρώτα τη μετατρέψει στην επιθυμητή μορφή. Καλείται από την κλάση UserLogin. Η UserLoginBrowser περιέχει και εσωτερικές κλάσεις οι οποίες θα αναλυθούν παρακάτω.

Πεδία

- `private ConnectSQL con`
  Αντικείμενο της κλάσης ConnectSQL που είναι υπεύθυνη για τη σύνδεση της εφαρμογής με τη Βάση Δεδομένων.
- `private Date localtime`
  Η ημερομηνία που αντιστοιχεί στην έναρξη της πλοήγησης του χρήστη
- `private String localtimeDB`
  Η ημερομηνία έναρξης της πλοήγησης του χρήστη σε κατάλληλη μορφή ώστε να Αποθηκευτεί στη Βάση Δεδομένων.
- `private String currentURL`
  Η μεταβλητή αυτή περιέχει το URL που επισκέπτεται ο χρήστης όταν επιλέγει την επόμενη κατηγορία από την ιεραρχία σε μορφή String.
- `private String curl`
  Η μεταβλητή αυτή κρατά τη σελίδα που επισκέπτεται ο χρήστης όταν πατά το πλήκτρο Back.
- `private String curl1`
  Η τρέχουσα σελίδα, από την οποία ο χρήστης πατά το πλήκτρο Back, φυλάσσεται στην curl1.
- `private String curl2`
  Η σελίδα η οποία εμφανίζεται όταν ο χρήστης πατά το πλήκτρο Forward.
- `private Vector history`
  Κρατά τις σελίδες που επισκέπτεται ο χρήστης απευθείας από την ιεραρχία ή μετά το πάτημα του πλήκτρου Forward.
- `private Vector first_page`
  Κρατά τις σελίδες που επισκέπτεται ο χρήστης απευθείας από την ιεραρχία. Αυτός ο vector χρησιμοποιείται από την εφαρμογή για να ελέγξει αν ο χρήστης έχει επισκεφτεί τουλάχιστον μια κατηγορία προτού κλείσει τον browser.
- `private Vector back`
  Κρατά τις σελίδες που επισκέπτεται ο χρήστης με πάτημα του πλήκτρου Back.
- `private String navigation=new String("")`
  Το String που περιέχει την πλοήγηση του χρήστη στην επιθυμητή μορφή.
- `private String t=new String("")`
  Το String που περιέχει το επόμενο στοιχείο που θα προστεθεί στην πλοήγηση του χρήστη.

Εσωτερικές κλάσεις

1)class backButtonListener implements ActionListener

Η κλάση αυτή ανταποκρίνεται στο πάτημα του πλήκτρου Back από το χρήστη. Επίσης αποθηκεύει στη μεταβλητή navigation την πλοήγηση σε κατάλληλη για αποθήκευση στη Βάση Δεδομένων μορφή.

1.1) Πεδία

- `protected JEditorPane jep`
  Το παράθυρο της εφαρμογής.
- `protected JLabel label`
  Ετικέτα στο κάτω μέρος του παραθύρου, η οποία απεικονίζει το τρέχον URL
- `protected JButton backButton`
  Το κουμπί Back
- `protected JButton forwardButton`
  Το κουμπί Forward
- `protected Vector history`
  Ο Vector που περιγράφηκε παραπάνω (μέσα στην κλάση τίθεται `this.history=history`)

1.2) Μέθοδοι

- `public backButtonListener(JEditorPane jep, JButton backButton, JButton forwardButton,Vector history, JLabel label)`
  Ο κατασκευαστής της κλάσης.
- `public void actionPerformed(ActionEvent e)`
  Κλάση υπεύθυνη για την ανταπόκριση στο πάτημα του κουμπιού Back από το χρήστη. Μετατρέπει την πλοήγηση του χρήστη σε κατάλληλη για αποθήκευση μορφή. Πραγματοποιεί ελέγχους ενεργοποίησης και απενεργοποίησης των πλήκτρων Back και Forward.

2)class forwardButtonListener implements ActionListener

Η κλάση αυτή ανταποκρίνεται στο πάτημα του πλήκτρου Forward από το χρήστη. Επίσης αποθηκεύει στη μεταβλητή navigation την πλοήγηση σε κατάλληλη για αποθήκευση στη Βάση Δεδομένων μορφή.

2.1) Πεδία

- `protected JEditorPane jep`
  Το παράθυρο της εφαρμογής.
- `protected JLabel label`
  Ετικέτα στο κάτω μέρος του παραθύρου, η οποία απεικονίζει το τρέχον URL
- `protected JButton backButton`
  Το κουμπί Back
- `protected JButton forwardButton`
  Το κουμπί Forward
- `protected Vector history`

Ο Vector που περιγράφηκε παραπάνω (μέσα στην κλάση τίθεται `this.history=history`)

2.2) Μέθοδοι

- `public forwardButtonListener(JEditorPane jep, JButton backButton, JButton forwardButton,Vector history, JLabel label)`
  Ο κατασκευαστής της κλάσης.
- `public void actionPerformed(ActionEvent e)`
  Κλάση υπεύθυνη για την ανταπόκριση στο πάτημα του κουμπιού Forward από το χρήστη. Μετατρέπει την πλοήγηση του χρήστη σε κατάλληλη για αποθήκευση μορφή. Πραγματοποιεί ελέγχους ενεργοποίησης και απενεργοποίησης των πλήκτρων Back και Forward.

3) class LinkFollower implements HyperlinkListener

3.1) Πεδία

- `protected JEditorPane jep`
  Το παράθυρο της εφαρμογής.
- `protected JLabel label`
  Ετικέτα στο κάτω μέρος του παραθύρου, η οποία απεικονίζει το τρέχον URL
- `protected JButton backButton`
  Το κουμπί Back
- `protected JButton forwardButton`
  Το κουμπί Forward
- `protected Vector history`
  Ο Vector που περιγράφηκε παραπάνω (μέσα στην κλάση τίθεται `this.history=history`)

3.2) Μέθοδοι

- `public LinkFollower(JEditorPane jep, JButton backButton, JButton forwardButton, Vector history, JLabel label)`
  Ο κατασκευαστής της κλάσης.
- `public void hyperlinkUpdate(HyperlinkEvent evt)`
  Κλάση υπεύθυνη για την ανταπόκριση στην επιλογή κάποιας κατηγορίας από το χρήστη. Μετατρέπει την πλοήγηση του χρήστη σε κατάλληλη για αποθήκευση μορφή. Ενεργοποιεί το πλήκτρο Back. Θέτει ως χρονική στιγμή έναρξης της πλοήγησης τη στιγμή που ο χρήστης επιλέγει την πρώτη κατηγορία.

Μέθοδοι

- `public UserLoginBrowser(ConnectSQL connect,String initialPage)`
  Ο κατασκευαστής της κλάσης. Συνδέεται με τη Βάση Δεδομένων και ανοίγει τον browser στην καθορισμένη σελίδα (initial page). Ορίζει διαχειριστές ενέργειας για τα συστατικά κουμπί Exit και κουμπί κλεισίματος παραθύρου:

`exitButton.addActionListener(new ActionListener()`
`f.addWindowListener(new WindowAdapter(),`
οι οποίοι αντίστοιχα υλοποιούν τις μεθόδους
`public void actionPerformed(ActionEvent e)` και
`public void windowClosing(WindowEvent e)`, οι οποίες αποθηκεύουν την πλοήγηση του χρήστη στη Βάση Δεδομένων.

- `protected static void setPage(JEditorPane jep, String url)`
  Θέτει τη σελίδα που αντιστοιχεί στο URL στην οθόνη jep.

5.2.2.4 public Class NewMemberBrowser extends JDialog

Η κλάση αυτή είναι υπεύθυνη για τη σωστή πλοήγηση ενός νέου χρήστη στο σύστημα NaviMoz και για την αποθήκευση της πλοήγησής του στη Βάση Δεδομένων σε κατάλληλη μορφή. Καλείται από την κλάση NewMember και για το λόγο αυτό είναι υποκλάση της JDialog (επειδή και η NewMember είναι υποκλάση της JDialog-Διαφορετικά δε θα μπορούσε να αποκριθείς στις ενέργειες του χρήστη). Τα πεδία και οι μέθοδοι που χρησιμοποιεί η κλάση αυτή είναι ακριβώς όμοια με της κλάσης UserLoginBrowser. Η μόνη διαφορά είναι στον κατασκευαστή της κλάσης ο οποίος είναι ο

- `Public NewMemberBrowser (NewMember parent, String title, boolean modal, ConnectSQL connect, String initialPage)`

  Η κλάση καλείται από ένα αντικείμενο της κλάσης NewMember, με τίτλο παραθύρου title, τιμή true ή false στη μεταβλητή modal, ανάλογα με το αν είναι επιθυμητό να κλειδώνει ή όχι το παράθυρο της εφαρμογής του πατέρα. Επίσης, θέτει ως αρχική σελίδα την initialPage.

5.2.2.5 public class ConnectSQL

Η κλάση αυτή είναι υπεύθυνη για την πραγματοποίηση της σύνδεσης με τη Βάση Δεδομένων και για την υποβολή των queries.

Πεδία:

- `public Connection connection;`
  Η σύνδεση με τη Βάση Δεδομένων
- `private static ResultSet rs;`
  Η δομή στην οποία επιστρέφονται τα αποτελέσματα των queries
- `private String ResultString`
  Το String το οποίο περιέχει το αποτέλεσμα ενός select query στην περίπτωση που αυτό αποτελείται από μία μόνο εγγραφή.

Μέθοδοι:

- `public ConnectSQL()`
  Ο κατασκευαστής της κλάσης. Εδώ πραγματοποιείται η σύνδεση με τη Βάση Δεδομένων. Χειρίζεται τα κατάλληλα Exceptions προκειμένου η σύνδεση να πραγματοποιηθεί σωστά.

- `public void QueryUpdate(String q)`
  Μέθοδος υπεύθυνη για την ενημέρωση της Βάσης. Το αντίστοιχο query είναι το q.
- `public void QuerySelect(String q)`
  Μέθοδος υπεύθυνη για την εφαρμογή ενός query q τύπου select στη Βάση, με την προϋπόθεση ότι το αποτέλεσμά του αποτελείται από μία μόνο εγγραφή.
- `public Vector Query1(String q1,int selection)`
  Μέθοδος υπεύθυνη για την εφαρμογή ενός query q1 τύπου select στη Βάση. Ανάλογα με τον αριθμό selection (1ή 2), οι εγγραφές που επιστρέφει το query αποτελούνται από 1 ή 2 στήλες.
- `public String getResultString()`
  Η μέθοδος αυτή επιστρέφει το αποτέλεσμα της QuerySelect.

5.2.2.6 public class Menu extends JFrame

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση του μενού επιλογών στο διαχειριστή του συστήματος και για την αρχικοποίηση της φόρμας επιστροφής των αποτελεσμάτων.

Πεδία:

- `private static JMenuBar menuBar`
  Το μενού επιλογών
- `private static JTextArea js1=new JTextArea()`
  Η περιοχή εμφάνισης των αποτελεσμάτων
- `public JScrollPane scrollPane`
  Η γραμμή κύλισης που εφαρμόζεται στην περιοχή εμφάνισης των αποτελεσμάτων.

Μέθοδοι:

- `public Menu()`
  Ο κατασκευαστής της κλάσης. Τοποθετεί τα αντικείμενα στο μενού και αρχικοποιεί τη φόρμα επιστροφής των αποτελεσμάτων.
- `public static void main(String[] args)`
  Ξεκινά την εφαρμογή

5.2.2.7 public static SimpleTasks extends JDialog implements ActionListener

Η κλάση αυτή είναι υπεύθυνη για την εκτέλεση της πρώτης ομάδας εργασιών, και πιο ειδικά της εύρεσης των πλοηγήσεων του επιλεγμένου χρήστη και της εύρεσης της διάρκειας κάθε πλοήγησης του χρήστη, που είναι απλές εργασίες ταυτοποίησης χρηστών.

Πεδία:

- `private static DateFormat formatter=new SimpleDateFormat("yyyy-MM-dd")`
  Η μορφή "yyyy-MM-dd" εμφάνισης της ημερομηνίας

- `private DateFormat formatter1=new SimpleDateFormat("yyyy-MM-dd HH:mm:ss")`

  Η μορφή "yyyy-MM-dd HH:mm:ss" εμφάνισης της ημερομηνίας
- `private SimpleDateFormat formatter2=new SimpleDateFormat("dd/MM/yyyy")`

  Η μορφή "dd/MM/yyyy" εμφάνισης της ημερομηνίας
- `private DateFormat sdf=new SimpleDateFormat(" HH:mm, dd/MM/yyyy")`

  Η μορφή " HH:mm, dd/MM/yyyy" εμφάνισης της ημερομηνίας
- `private static Date start=null`

  Η ημερομηνία ενδιαφέροντος έναρξης
- `private static Date end=null`

  Η ημερομηνία ενδιαφέροντος λήξης
- `public Menu parent2`

  Ο πατέρας του τρέχοντος Dialog Box
- `private String name`

  Το όνομα και επώνυμο του χρήστη που επιλέγεται από τη λίστα
- `private static String st1`

  Η ημέρα από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st2`

  Ο μήνας από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st3`

  Το έτος από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st4`

  Η ημέρα από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static String st5`

  Ο μήνας από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static String nst6`

  Το έτος από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static JScrollPane myScrollPane`

  Ο κυλιστής που προσαρμόζεται στη φόρμα επιστροφής των αποτελεσμάτων
- `private JList myList`

  Η λίστα με τα ονόματα των χρηστών
- `private JLabel label1`

  Βοηθητική ετικέτα για την εισαγωγή πληροφοριακού κειμένου
- `private JLabel label2`

  Βοηθητική ετικέτα για την εισαγωγή πληροφοριακού κειμένου
- `private JLabel label22`

  Βοηθητική ετικέτα για την εισαγωγή πληροφοριακού κειμένου
- `private JLabel label23`

  Βοηθητική ετικέτα για την εισαγωγή πληροφοριακού κειμένου
- `private JLabel label24`

  Βοηθητική ετικέτα για την εισαγωγή πληροφοριακού κειμένου
- `private JLabel label3`

  Βοηθητική ετικέτα για την εισαγωγή πληροφοριακού κειμένου

- `private JLabel label33`
  Βοηθητική ετικέτα για την εισαγωγή πληροφοριακού κειμένου
- `private JLabel label34`
  Βοηθητική ετικέτα για την εισαγωγή πληροφοριακού κειμένου
- `private JLabel label35`
  Βοηθητική ετικέτα για την εισαγωγή πληροφοριακού κειμένου
- `private JComboBox yearbox1`
  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox yearbox2`
  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος λήξης
- `private JComboBox monthbox1`
  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox monthbox2`
  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος λήξης
- `private JComboBox daybox1`
  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox daybox1`
  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος λήξης
- `private JButton ok`
  Το κουμπί OK
- `private JButton cancel`
  Το κουμπί Cancel
- `private JTextArea t1`
  Η περιοχή εμφάνισης των αποτελεσμάτων
- `private JScrollPane scroll`
  Ο κυλιστής της περιοχής εμφάνισης των αποτελεσμάτων
- `private ConnectSQL cp`
  Η σύνδεση με τη Βάση Δεδομένων
- `private int flag1`
  Ακέραιος που δηλώνει αν πραγματοποιείται η πρώτη εργασία (flag=1), δηλαδή η εύρεση των πλοηγήσεων του επιλεγμένου χρήστη ή η δεύτερη (flag=2), δηλαδή η εύρεση της χρονικής διάρκειας κάθε πλοήγησης του επιλεγμένου χρήστη.

Μέθοδοι:

- `void buildConstraints(GridBagConstraints gbc, int gx, int gy, int gw, int gh, int wx, int wy)`
  Μέθοδος υπεύθυνη για τη δημιουργία του πλέγματος GridBagLayout που χρησιμοποιείται για τη διάταξη των συστατικών.
- `public SimpleTasks (Menu parent, String title, boolean modal, JTextArea textarea, int flag)`
  Ο κατασκευαστής της κλάσης. Καλείται από κάποιο αντικείμενο της κλάσης Menu, έχει τίτλο παραθύρου title, δυνατότητα κλειδώματος του παραθύρου που την κάλεσε (modal=true για κλείδωμα και modal=false για μη κλείδωμα), επιστρέφει τα αποτελέσματα στην περιοχή textarea και επιλέγει ποια από τις ετικέτες που αντιστοιχούν στις δύο εργασίες θα απεικονιστεί ανάλογα με την τιμή της flag.(Μέσα στον κατασκευαστή γίνεται flag=flag1).

- `private class Handler implements ActionListener`

  Η μέθοδος αυτή είναι υπεύθυνη για την απόκριση στο πάτημα των κουμπιών από το διαχειριστή. Ευθύνεται για τη σύνδεση με τη Βάση Δεδομένων, την εφαρμογή των queries, την ανάκτηση των αποτελεσμάτων, την παρουσίασή τους σε μορφή κατανοητή από το διαχειριστή και τέλος την αποσύνδεση από τη Βάση Δεδομένων. Ανάλογα με την τιμή της flag επιτελεί την πρώτη ή τη δεύτερη εργασία. Επίσης πραγματοποιεί τον έλεγχο αν έχει επιλεχθεί κάποιος χρήστης και, αν όχι, εμφανίζει κατάλληλο μήνυμα.
- `protected void processWindowEvent(WindowEvent e)`

  Μέθοδος που χειρίζεται το κλείσιμο του παραθύρου.
- `public static Date StartSession()`

  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος έναρξης
- `public static Date EndSession()`

  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος λήξης
- `public void actionPerformed (ActionEvent evt)`

  Η μέθοδος αυτή αποκρίνεται στην επιλογή των ημερομηνιών ενδιαφέροντος από το διαχειριστή και μετατρέπει σε Strings τις επιλογές του έτσι ώστε να είναι σε κατάλληλη για επεξεργασία μορφή από τις μεθόδους EndSession και StartSession.

5.2.2.8 public static class Task4 extends JDialog implements ActionListener

Η κλάση αυτή είναι υπεύθυνη για την εύρεση των πλοηγήσεων εκείνων που αποτελούν υπερσύνολο μιας δοσμένης από το διαχειριστή κι εκείνων που είναι ακριβώς ίδιες με μια δοσμένη

Πεδία:

- `private static DateFormat formatter=new SimpleDateFormat("yyyy-MM-dd")`

  Η μορφή "yyyy-MM-dd" εμφάνισης της ημερομηνίας
- `private SimpleDateFormat formatter2=new SimpleDateFormat("dd/MM/yyyy")`

  Η μορφή "dd/MM/yyyy" εμφάνισης της ημερομηνίας
- `private static Date start=null`

  Η ημερομηνία ενδιαφέροντος έναρξης
- `private static Date end=null`

  Η ημερομηνία ενδιαφέροντος λήξης
- `public Menu parent2`

  Ο πατέρας του τρέχοντος Dialog Box
- `private static String st1`

  Η ημέρα από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st2`

  Ο μήνας από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st3`

  Το έτος από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st4`

Η ημέρα από την ημερομηνία λήξης που επιλέγει ο διαχειριστής

- `private static String st5`

  Ο μήνας από την ημερομηνία λήξης που επιλέγει ο διαχειριστής

- `private static String nst6`

  Το έτος από την ημερομηνία λήξης που επιλέγει ο διαχειριστής

- `private JScrollPane myScrollPane`

  Ο κυλιστής που προσαρμόζεται στη φόρμα επιστροφής των αποτελεσμάτων

- `private JLabel label1, label2, label22, label23, label24, label3, label33, label34, label35, label1neo, label2neo`

  Βοηθητικές ετικέτες για την εισαγωγή πληροφοριακού κειμένου

- `private JComboBox yearbox1`

  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος έναρξης

- `private JComboBox yearbox2`

  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος λήξης

- `private JComboBox monthbox1`

  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος έναρξης

- `private JComboBox monthbox2`

  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος λήξης

- `private JComboBox daybox1`

  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος έναρξης

- `private JComboBox daybox1`

  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος λήξης

- `private  JButton ok`

  Το κουμπί OK

- `private JButton cancel`

  Το κουμπί Cancel

- `private JButton path2`

  Το κουμπί που ανοίγει τον browser του dmoz

- `private JTextArea copy_of`

  Η περιοχή εμφάνισης των αποτελεσμάτων

- `private JScrollPane scroll`

  Η γραμμή κύλισης της περιοχής εμφάνισης των αποτελεσμάτων

- `private ConnectSQL cp`

  Η σύνδεση με τη Βάση Δεδομένων

- `private int flag1`

  Ακέραιος που δηλώνει αν πραγματοποιείται η πρώτη εργασία (flag=1), δηλαδή η εύρεση των πλοηγήσεων εκείνων που αποτελούν υπερσύνολο της δοσμένης ή η δεύτερη (flag=2), δηλαδή η εύρεση των πλοηγήσεων εκείνων που είναι ακριβώς ίδιες με τη δοσμένη.

Μέθοδοι

- `void buildConstraints(GridBagConstraints gbc, int gx, int gy, int gw, int gh, int wx, int wy)`

  Μέθοδος υπεύθυνη για τη δημιουργία του πλέγματος GridBagLayout που χρησιμοποιείται για τη διάταξη των συστατικών.

- `public Task4(Menu parent, String title, boolean modal, JTextArea textarea, int flag)`

  Ο κατασκευαστής της κλάσης. Καλείται από κάποιο αντικείμενο της κλάσης Menu, έχει τίτλο παραθύρου title, δυνατότητα κλειδώματος του παραθύρου που την κάλεσε (modal=true για κλείδωμα και modal=false για μη κλείδωμα), επιστρέφει τα αποτελέσματα στην περιοχή textarea και επιλέγει ποια από τις ετικέτες που αντιστοιχούν στις δύο εργασίες θα απεικονιστεί ανάλογα με την τιμή της flag.(Μέσα στον κατασκευαστή γίνεται flag=flag1).

- `private class Handler implements ActionListener{`
  `public void actionPerformed( ActionEvent event )`

  Η μέθοδος αυτή είναι υπεύθυνη για την απόκριση στο πάτημα των κουμπιών από το διαχειριστή. Ευθύνεται για τη σύνδεση με τη Βάση Δεδομένων, την εφαρμογή των queries, την ανάκτηση των αποτελεσμάτων, την παρουσίασή τους σε μορφή κατανοητή από το διαχειριστή και τέλος την αποσύνδεση από τη Βάση Δεδομένων. Ανάλογα με την τιμή της flag επιτελεί την πρώτη ή τη δεύτερη εργασία. Επίσης πραγματοποιεί τον έλεγχο αν το πεδίο εισαγωγής της πλοήγησης-πρότυπο είναι ή όχι κενό και, αν όχι, εμφανίζει κατάλληλο μήνυμα.

- `private class Handler2 implements ActionListener`
  `public void actionPerformed( ActionEvent ev )`

  Μέθοδος που ανταποκρίνεται στο πάτημα του κουμπιού για το άνοιγμα του browser του dmoz.

- `protected void processWindowEvent(WindowEvent e)`

  Μέθοδος που χειρίζεται το κλείσιμο του παραθύρου.

- `public static Date StartSession()`

  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος έναρξης

- `public static Date EndSession()`

  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος λήξης

- `public void actionPerformed (ActionEvent evt)`

  Η μέθοδος αυτή αποκρίνεται στην επιλογή των ημερομηνιών ενδιαφέροντος από το διαχειριστή και μετατρέπει σε Strings τις επιλογές του έτσι ώστε να είναι σε κατάλληλη για επεξεργασία μορφή από τις μεθόδους EndSession και StartSession

5.2.2.9 public static class Task2 extends JDialog implements ActionListener

Η κλάση αυτή είναι υπεύθυνη για την ανάκτηση των πλοηγήσεων εκείνων που είναι τουλάχιστον κατά ένα συγκεκριμένο ποσοστό όμοιες με μια δοσμένη. Το ποσοστό αυτό καθώς και η πλοήγηση δίνονται από το διαχειριστή του συστήματος.

Πεδία:

- `private static DateFormat formatter=new SimpleDateFormat("yyyy-MM-dd")`

  Η μορφή "yyyy-MM-dd" εμφάνισης της ημερομηνίας

- `private DateFormat formatter2=new SimpleDateFormat("dd/MM/yyyy")`

  Η μορφή "dd/MM/yyyy" εμφάνισης της ημερομηνίας

- `private static Date start=null`

Η ημερομηνία ενδιαφέροντος έναρξης

- `private static Date end=null`

  Η ημερομηνία ενδιαφέροντος λήξης

- `public Menu parent2`

  Ο πατέρας του τρέχοντος Dialog Box

- `private static String st1`

  Η ημέρα από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής

- `private static String st2`

  Ο μήνας από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής

- `private static String st3`

  Το έτος από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής

- `private static String st4`

  Η ημέρα από την ημερομηνία λήξης που επιλέγει ο διαχειριστής

- `private static String st5`

  Ο μήνας από την ημερομηνία λήξης που επιλέγει ο διαχειριστής

- `private static String nst6`

  Το έτος από την ημερομηνία λήξης που επιλέγει ο διαχειριστής

- `private JScrollPane myScrollPane`

  Ο κυλιστής που προσαρμόζεται στη φόρμα επιστροφής των αποτελεσμάτων

- `private JLabel label1, label2, label22, label23, label24, label3, label33, label34, label35, label1neo, label2neo, label9`

  Βοηθητικές ετικέτες για την εισαγωγή πληροφοριακού κειμένου

- `private JComboBox yearbox1`

  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος έναρξης

- `private JComboBox yearbox2`

  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος λήξης

- `private JComboBox monthbox1`

  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος έναρξης

- `private JComboBox monthbox2`

  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος λήξης

- `private JComboBox daybox1`

  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος έναρξης

- `private JComboBox daybox1`

  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος λήξης

- `private JButton ok`

  Το κουμπί OK

- `private JButton cancel`

  Το κουμπί Cancel

- `private JButton path`

  Το κουμπί που ανοίγει τον browser του dmoz

- `private JTextArea path`

  Η περιοχή εμφάνισης των αποτελεσμάτων

- `private JScrollPane scroll`

  Η γραμμή κύλισης της περιοχής εμφάνισης των αποτελεσμάτων

- `private ConnectSQL cp`

  Η σύνδεση με τη Βάση Δεδομένων

- `private boolean flag`

  Μεταβλητή που γίνεται true όταν ο διαχειριστής πραγματοποιεί λάθος εισαγωγή στο πεδίο του ποσοστού και ξαναγίνεται false όταν μετά από προειδοποιητικό μήνυμα εισάγει σωστό. Η μεταβλητή αυτή δηλαδή χρησιμεύει για τον έλεγχο.

Μέθοδοι:

- `void buildConstraints(GridBagConstraints gbc, int gx, int gy, int gw, int gh, int wx, int wy)`

  Μέθοδος υπεύθυνη για τη δημιουργία του πλέγματος GridBagLayout που χρησιμοποιείται για τη διάταξη των συστατικών.

- `public Task2(Menu parent, String title, boolean modal, JTextArea textarea)`

  Ο κατασκευαστής της κλάσης. Καλείται από κάποιο αντικείμενο της κλάσης Menu, έχει τίτλο παραθύρου title, δυνατότητα κλειδώματος του παραθύρου που την κάλεσε (modal=true για κλείδωμα και modal=false για μη κλείδωμα) και επιστρέφει τα αποτελέσματα στην περιοχή textarea. Πραγματοποιεί επίσης τη σύνδεση με τη Βάση Δεδομένων.

- `private class Handler implements ActionListener`
  `public void actionPerformed( ActionEvent event )`

  Η μέθοδος αυτή είναι υπεύθυνη για την απόκριση στο πάτημα των κουμπιών από το διαχειριστή. Ευθύνεται για την εφαρμογή των queries, την ανάκτηση των αποτελεσμάτων, την παρουσίασή τους σε μορφή κατανοητή από το διαχειριστή και τέλος την αποσύνδεση από τη Βάση Δεδομένων.

  Διαθέτει επίσης δύο σημαντικά πεδία:

  1) `number`: Είναι αριθμός μεταξύ 0 και 1, και είναι ουσιαστικά το αποτέλεσμα της `transform(String arg1)`. Πρόκειται για το ποσοστό που εισάγει ο χρήστης αφού έχει μετατραπεί κατά τέτοιο τρόπο ώστε να είναι δεκαδικός μεταξύ 0 και 1 και να εκφράζει ομοιότητα.

  2) `result`: Είναι το αποτέλεσμα της `Struct_Dist`, η οποία καλείται τόσες φορές όσες και οι πλοηγήσεις της Βάσης Δεδομένων, και δηλώνει το βαθμό ομοιότητας της πλοήγησης που έδωσε ο διαχειριστής με κάθε μια από τις πλοηγήσεις των χρηστών.

  Η κλάση αυτή πραγματοποιεί έλεγχο αν η τιμή της `number` είναι μικρότερη από αυτή της `result` και, για τις πλοηγήσεις που συμβαίνει αυτό, παρουσιάζονται τα αποτελέσματα στη φόρμα των αποτελεσμάτων.

- `private class Handler2 implements ActionListener`
  `public void actionPerformed( ActionEvent ev )`

  Μέθοδος που ανταποκρίνεται στο πάτημα του κουμπιού για το άνοιγμα του browser του dmoz.

- `protected void processWindowEvent(WindowEvent e)`

  Μέθοδος που χειρίζεται το κλείσιμο του παραθύρου.

- `public static Date StartSession()`

  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος έναρξης

- `public static Date EndSession()`

  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος λήξης

- `public void actionPerformed (ActionEvent evt)`

Η μέθοδος αυτή αποκρίνεται στην επιλογή των ημερομηνιών ενδιαφέροντος από το διαχειριστή και μετατρέπει σε Strings τις επιλογές του έτσι ώστε να είναι σε κατάλληλη για επεξεργασία μορφή από τις μεθόδους EndSession και StartSession

- `private static double transform(String arg1)`

  Μέθοδος που μετατρέπει το όρισμα σε δεκαδικό διπλής ακρίβειας και στη συνέχεια τον διαιρεί με 100 και το αποτέλεσμα το αφαιρεί από το 1. Χρησιμοποιείται για να μετατρέψει το ποσοστό % που έδωσε ο διαχειριστής σε βαθμό ομοιότητας (αριθμός 0-1).

5.2.2.10 public class StructDist

Η κλάση αυτή χρησιμεύει για την εύρεση της δομικής απόστασης μεταξύ δύο συμβολοακολουθιών, με την τροποποίηση του βασικού αλγορίθμου (παράγραφος 3.3),όπως αυτή περιγράφεται στην 5.2.2. Χρησιμοποιείται από την κλάση Task2, για την εύρεση των πλοηγήσεων εκείνων που είναι κατά ένα ποσοστό όμοιες με μία δοσμένη από το διαχειριστή.

Πεδία:

- `private Vector v1`

  Διάνυσμα που περιέχει την πρώτη λέξη που αποτελεί είσοδο στον αλγόριθμο σε μορφή String, μετά τη διαγραφή των διαχωριστικών «/» μεταξύ των σελίδων.

- `private Vector v2`

  Διάνυσμα που περιέχει τη δεύτερη λέξη που αποτελεί είσοδο στον αλγόριθμο σε μορφή String, μετά τη διαγραφή των διαχωριστικών «/» μεταξύ των σελίδων.

- `private String[] word1`

  Πίνακας που σαν στοιχεία του έχει τις σελίδες της πρώτης πλοήγησης (τα στοιχεία δηλαδή του πρώτου String που αποτελεί είσοδο στον αλγόριθμο)

- `private String[] word2`

  Πίνακας που σαν στοιχεία του έχει τις σελίδες της δεύτερης πλοήγησης (τα στοιχεία δηλαδή του δεύτερου String που αποτελεί είσοδο στον αλγόριθμο)

- `private int similar`

  Αποτελεί μια ένδειξη που χρησιμεύει για την εξαγωγή του ποσοστού ομοιότητας των δύο συμβολοακολουθιών. Σχετίζεται με την απόσταση περιεχομένου τους.

- `private double normalization`

  Η κανονικοποιημένη δομική απόσταση μεταξύ των δύο συμβολοακολουθιών.

- `private double total_length`

  Το συνολικό μήκος των δύο πλοηγήσεων (δηλαδή το άθροισμα των λέξεων που περιέχουν)

- `private double Number_of_steps`

  Η συντακτική απόσταση μεταξύ των δύο πλοηγήσεων σύμφωνα με τον αντίστοιχο αλγόριθμο της υποενότητας 3.3 (χωρίς δηλαδή την τροποποίηση που παρουσιάζεται στην ενότητα….)

- `private double total_result`

  Η συνολική δομική απόσταση των δύο συμβολοακολουθιών, με βάση τον τροποποιημένο αλγόριθμο της ενότητας …..

- `private double percentage1`

Η τιμή της similar διαιρεμένη με την τιμή της total_length. Αντιστοιχεί σε όρους απόστασης.

- `private double percentage2`

  Ισούται με 1- percentage1. Αντιστοιχεί σε όρους ομοιότητας.

Μέθοδοι:

- `public StructDist ()`

  Ο κατασκευαστής της κλάσης

- `public double computation(String st1, String st2)`

  Η μέθοδος που υπολογίζει τη συνολική δομική απόσταση μεταξύ των δοσμένων συμβολοακολουθιών s1 και s2, σύμφωνα με τον αλγόριθμο που περιγράφηκε στην ενότητα

- `public static int change(Vector vector1, Vector vector2, int x, int y)`

  Υπολογίζει αν το y στοιχείο του vector2 προκύπτει από πράξη μετατροπής του στοιχείου x του vector1. Επιστρέφει 1 αν ναι, αλλιώς επιστρέφει 0.

- `public static int min(int x1, int x2, int x3)`

  Μέθοδος για τον υπολογισμό του μικρότερου από τα x1, x2 και x3.

5.2.2.11 public class PopularNavigations extends JDialog implements ActionListener

Η κλάση αυτή είναι υπεύθυνη για την εύρεση των πιο δημοφιλών πλοηγήσεων, το πλήθος των οποίων εισάγει ο διαχειριστής σε ειδική φόρμα, και για την εμφάνιση του αποτελέσματος στο διαχειριστή του συστήματος.

Πεδία:

- `private static DateFormat formatter=new SimpleDateFormat("yyyy-MM-dd")`

  Η μορφή "yyyy-MM-dd" εμφάνισης της ημερομηνίας

- `private static SimpleDateFormat formatter2=new SimpleDateFormat("dd/MM/yyyy")`

  Η μορφή "dd/MM/yyyy" εμφάνισης της ημερομηνίας

- `private static Date start=null`

  Η ημερομηνία ενδιαφέροντος έναρξης

- `private static Date end=null`

  Η ημερομηνία ενδιαφέροντος λήξης

- `public static Menu parent2`

  Ο πατέρας του τρέχοντος Dialog Box

- `private static String st1`

  Η ημέρα από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής

- `private static String st2`

  Ο μήνας από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής

- `private static String st3`

Το έτος από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής

- `private static String st4`
  Η ημέρα από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static String st5`
  Ο μήνας από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static String nst6`
  Το έτος από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static JScrollPane myScrollPane`
  Ο κυλιστής που προσαρμόζεται στη φόρμα επιστροφής των αποτελεσμάτων
- `private JLabel label1, label2, label22, label23, label24, label3, label33, label34, label35, label1neo, label2neo,mhn`
  Βοηθητικές ετικέτες για την εισαγωγή πληροφοριακού κειμένου
- `private JComboBox yearbox1`
  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox yearbox2`
  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος λήξης
- `private JComboBox monthbox1`
  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox monthbox2`
  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος λήξης
- `private JComboBox daybox1`
  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox daybox1`
  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος λήξης
- `private JButton ok`
  Το κουμπί OK
- `private JButton cancel`
  Το κουμπί Cancel
- `private JTextField data`
  Η περιοχή εισόδου του πλήθους των πιο δημοφιλών πλοηγήσεων
- `private JTextArea path`
  Η περιοχή εμφάνισης των αποτελεσμάτων
- `private JScrollPane scroll`
  Ο κυλιστής της περιοχής εμφάνισης των αποτελεσμάτων
- `private ConnectSQL cp`
  Η σύνδεση με τη Βάση Δεδομένων
- `private Vector userpat`
  Περιέχει ομαδοποιημένες τις ίδιες πλοηγήσεις και πόσες φορές έχει πραγματοποιηθεί η κάθε μια.
- `private int number`
  Το νούμερο των δημοφιλέστερων πλοηγήσεων που εισάγει ο διαχειριστής.
- `private String no`
  Ο αριθμός που δίνει ο χρήστης σε μορφή String, πριν δηλαδή μετατραπεί σε ακέραιο.

Μέθοδοι:

- `void buildConstraints(GridBagConstraints gbc, int gx, int gy, int gw, int gh, int wx, int wy)`

  Μέθοδος υπεύθυνη για τη δημιουργία του πλέγματος GridBagLayout που χρησιμοποιείται για τη διάταξη των συστατικών.

- `public PopularNavigations(Menu parent, String title, boolean modal, JTextArea textarea)`

  Ο κατασκευαστής της κλάσης. Καλείται από κάποιο αντικείμενο της κλάσης Menu, έχει τίτλο παραθύρου title, δυνατότητα κλειδώματος του παραθύρου που την κάλεσε (modal=true για κλείδωμα και modal=false για μη κλείδωμα) και επιστρέφει τα αποτελέσματα στην περιοχή textarea. Επίσης πραγματοποιεί τη σύνδεση με τη Βάση Δεδομένων.

- `private class Handler implements ActionListener`
  `public void actionPerformed( ActionEvent event )`

  Η μέθοδος αυτή είναι υπεύθυνη για την απόκριση στο πάτημα των κουμπιών από το διαχειριστή. Ευθύνεται για την εφαρμογή των queries, την ανάκτηση των αποτελεσμάτων, την παρουσίασή τους σε μορφή κατανοητή από το διαχειριστή και τέλος την αποσύνδεση από τη Βάση Δεδομένων. Επίσης πραγματοποιεί τους απαραίτητους ελέγχους αυτού που εισάγει ο διαχειριστής και εμφανίζει κατάλληλα μηνύματα.

- `protected void processWindowEvent(WindowEvent e)`

  Μέθοδος που χειρίζεται το κλείσιμο του παραθύρου.

- `public static Date StartSession()`

  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος έναρξης

- `public static Date EndSession()`

  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος λήξης

- `public void actionPerformed (ActionEvent evt)`

  Η μέθοδος αυτή αποκρίνεται στην επιλογή των ημερομηνιών ενδιαφέροντος από το διαχειριστή και μετατρέπει σε Strings τις επιλογές του έτσι ώστε να είναι σε κατάλληλη για επεξεργασία μορφή από τις μεθόδους EndSession και StartSession

5.2.2.12 public class Clustering extends JDialog implements ActionListener

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση στο διαχειριστή της δυνατότητας επιλογής ανάμεσα στις δύο διαθέσιμες από το σύστημα μεθόδους συσταδοποίησης: K-Means και Single Link.

Πεδία:

- `private ConnectSQL cp`

  Η σύνδεση με τη Βάση δεδομένων

- `private JLabel label1`

  Βοηθητική ετικέτα για την εμφάνιση πληροφοριακού κειμένου

- `private JRadioButton KMeans`

  Το κουμπί επιλογής του αλγορίθμου K-Means.

- `private JRadioButton SLink`

  Το κουμπί επιλογής του αλγορίθμου Single Link.

- `private ButtonGroup radioGroup`

  Τοποθετεί τα δύο προηγούμενα Radio Buttons στην ίδια ομάδα, έτσι ώστε να είναι αλληλοαποκλειόμενα.
- `private JButton button1`

  Πλήκτρο που αν πατηθεί, ανοίγει ο Internet Explorer σε σελίδα με πληροφορίες για τον αλγόριθμο K-Means.
- `private JButton button2`

  Πλήκτρο που αν πατηθεί, ανοίγει ο Internet Explorer σε σελίδα με πληροφορίες για τον αλγόριθμο Single Link.
- `private JButton cancel`

  Το κουμπί Cancel.
- `public Menu father`

  Η κλάση που κάλεσε αυτή τη JDialog (ο πατέρας της)
- `private JTextArea path1`

  Η περιοχή εμφάνισης των αποτελεσμάτων.

Μέθοδοι:

- `void buildConstraints(GridBagConstraints gbc, int gx, int gy, int gw, int gh, int wx, int wy)`

  Μέθοδος υπεύθυνη για τη δημιουργία του πλέγματος GridBagLayout που χρησιμοποιείται για τη διάταξη των συστατικών.
- `public Clustering(Menu parent, String title, boolean modal, JTextArea textarea)`

  Ο κατασκευαστής της κλάσης. Καλείται από κάποιο αντικείμενο της κλάσης Menu, έχει τίτλο παραθύρου title, δυνατότητα κλειδώματος του παραθύρου που την κάλεσε (modal=true για κλείδωμα και modal=false για μη κλείδωμα) και επιστρέφει τα αποτελέσματα στην περιοχή textarea. Επίσης πραγματοποιεί τη σύνδεση με τη Βάση Δεδομένων.
- `protected void processWindowEvent(WindowEvent e)`

  Μέθοδος που χειρίζεται το κλείσιμο του παραθύρου.
- `public void actionPerformed (ActionEvent evt)`

  Μέθοδος που αποκρίνεται στις ενέργειες του διαχειριστή. Με την επιλογή ενός αλγορίθμου συσταδοποίησης, καλεί την αντίστοιχη κλάση υλοποίησής του. Οι μεταβλητές father, cp και path περνούν σαν ορίσματα στην κλήση της κλάσης αυτής.

5.2.2.13 public class SLinkUI extends JDialog implements ActionListener

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση στο χρήστη των επιλογών που του παρέχει ο αλγόριθμος «μονός σύνδεσμος». Καλεί τις κλάσεις υλοποίησης της τεχνικής αυτής. Τα αποτελέσματα επιστρέφονται από την κλάση αυτή μέσω της διαπροσωπείας χρήστη.

Πεδία:

- `private static DateFormat formatter=new SimpleDateFormat("yyyy-MM-dd")`

Η μορφή "yyyy-MM-dd" εμφάνισης της ημερομηνίας

- `private SimpleDateFormat formatter2=new SimpleDateFormat("dd/MM/yyyy")`

  Η μορφή "dd/MM/yyyy" εμφάνισης της ημερομηνίας
- `private static Date start=null`

  Η ημερομηνία ενδιαφέροντος έναρξης
- `private static Date end=null`

  Η ημερομηνία ενδιαφέροντος λήξης
- `public Menu parent2`

  Ο πατέρας του τρέχοντος Dialog Box
- `private static String st1`

  Η ημέρα από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st2`

  Ο μήνας από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st3`

  Το έτος από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st4`

  Η ημέρα από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static String st5`

  Ο μήνας από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static String nst6`

  Το έτος από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private JScrollPane myScrollPane`

  Ο κυλιστής που προσαρμόζεται στη φόρμα επιστροφής των αποτελεσμάτων
- `private JLabel label1, label2, label22, label23, label24, label3, label33, label34, label35,mhn, cindex1,cindex2`

  Βοηθητικές ετικέτες για την εισαγωγή πληροφοριακού κειμένου
- `private static JComboBox yearbox1`

  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος έναρξης
- `private static JComboBox yearbox2`

  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος λήξης
- `private JComboBox monthbox1`

  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox monthbox2`

  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος λήξης
- `private JComboBox daybox1`

  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox daybox1`

  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος λήξης
- `private JButton ok`

  Το κουμπί OK
- `private JButton cancel`

  Το κουμπί Cancel
- `private JButton button`

Το κουμπί που πραγματοποιεί τη σύνδεση με την κλάση CIndex και εφαρμόζει τον αλγόριθμο για τα τρέχοντα δεδομένα.

- `private JTextArea path`

  Η περιοχή εμφάνισης των αποτελεσμάτων
- `private JScrollPane scroll`

  Ο κυλιστής της περιοχής εμφάνισης των αποτελεσμάτων
- `public Menu father`

  Ο πατέρας της κλάσης που κάλεσε την τρέχουσα
- `private JDialog parent2`

  Ο πατέρας της κλάσης αυτής
- `private JTextField data`

  Το πεδίο όπου ο διαχειριστής εισάγει το επίπεδο συσταδοποίησης
- `private ConnectSQL cp`

  Η σύνδεση με τη Βάση Δεδομένων
- `private double number`

  Το επίπεδο συσταδοποίησης που εισάγει ο διαχειριστής
- `private ConnectedComponents concomp`

  Αντικείμενο της κλάσης ConnectedComponents
- `private String no`

  Ο αριθμός που δίνει ο χρήστης σε μορφή String, πριν δηλαδή μετατραπεί σε δεκαδικό διπλής ακρίβειας.
- `private Vector clustered_patterns`

  Οι συστάδες των πλοηγήσεων
- `private Vector names`

  Τα ονόματα των χρηστών που έχουν πραγματοποιήσει τις πλοηγήσεις, από μια φορά το καθένα.
- `private Vector namesPlai`

  Τα ονόματα των χρηστών που έχουν πραγματοποιήσει τις πλοηγήσεις σε ένα προς ένα αντιστοιχία με αυτές.
- `private Vector patterns`

  Το σύνολο των πλοηγήσεων για τη δεδομένη χρονική περίοδο
- `private Dist_Matrix dist`

  Στιγμιότυπο της κλάσης Dist_Matrix. Χρησιμεύει για την επιλογή των πλοηγήσεων που βρίσκονται μέσα στο χρονικό διάστημα που ενδιαφέρει το διαχειριστή.
- `private CIndex cindex`

  Στιγμιότυπο της κλάσης CIndex.

Μέθοδοι:

- `void buildConstraints(GridBagConstraints gbc, int gx, int gy, int gw, int gh, int wx, int wy)`

  Μέθοδος υπεύθυνη για τη δημιουργία του πλέγματος GridBagLayout που χρησιμοποιείται για τη διάταξη των συστατικών.
- `public SLinkUI(JDialog parent, Menu newFrame, String title, boolean modal, JTextArea textarea, ConnectSQL con)`

  Ο κατασκευαστής της κλάσης. Είναι υπεύθυνος για τη διάταξη των συστατικών στο παράθυρο της διαπροσωπείας χρήστη. Ο πατέρας της κλάσης αυτής είναι αντικείμενο

της κλάσης Clustering που περιγράφηκε παραπάνω. Επίσης χρειάζεται να κρατηθεί πληροφορία και για τον πατέρα newFrame της κλάσης Clustering, που είναι τύπου Menu, γιατί εκεί θα επιστραφούν τα αποτελέσματα. Η σύνδεση με τη Βάση Δεδομένων είναι η con, ο τίτλος του παραθύρου είναι title, η μεταβλητή modal είναι true ή false ανάλογα με το αν κλειδώνει ή όχι το παράθυρο-πατέρας και, τέλος, η περιοχή εμφάνισης των αποτελεσμάτων είναι η textarea.

- `private class Handler implements ActionListener{`
  `public void actionPerformed( ActionEvent event )`
  Η μέθοδος αυτή είναι υπεύθυνη για την απόκριση στο πάτημα των κουμπιών OK και Cancel από το διαχειριστή. Συνδέεται με την κλάση ConnectedComponents για την εφαρμογή του αλγορίθμου «μονός σύνδεσμος» και επιστρέφει τα αποτελέσματα στο διαχειριστή. Είναι υπεύθυνη και για την εμφάνιση των ονομάτων των χρηστών που έχουν πραγματοποιήσει τις πλοηγήσεις σε κάθε συστάδα Πραγματοποιεί όλους τους απαραίτητους ελέγχους του επιπέδου συσταδοποίησης που εισάγει ο διαχειριστής. Επίσης κλείνει τη σύνδεση με τη Βάση Δεδομένων.
- `private class Handler1 implements ActionListener{`
  `public void actionPerformed( ActionEvent event )`
  Η μέθοδος αυτή είναι υπεύθυνη για την απόκριση στο πάτημα του κουμπιού που δίνει το αποτέλεσμα της εφαρμογής του αλγορίθμου C-Index. Συνδέεται με την κλάση αυτή. Παρουσιάζει στο διαχειριστή το αποτέλεσμα της εκτέλεσής της.
- `protected void processWindowEvent(WindowEvent e)`
  Μέθοδος που χειρίζεται το κλείσιμο του παραθύρου.
- `public static Date StartSession()`
  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος έναρξης
- `public static Date EndSession()`
  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος λήξης
- `public void actionPerformed (ActionEvent evt)`
  Η μέθοδος αυτή αποκρίνεται στην επιλογή των ημερομηνιών ενδιαφέροντος από το διαχειριστή και μετατρέπει σε Strings τις επιλογές του έτσι ώστε να είναι σε κατάλληλη για επεξεργασία μορφή από τις μεθόδους EndSession και StartSession

5.2.2.14 public class Dist_Matrix

Η κλάση αυτή υπολογίζει τις αποστάσεις ανά δύο όλων των προτύπων στο χρονικό διάστημα ενδιαφέροντος και τις καταχωρεί στον πίνακα αποστάσεων.

Πεδία:

- `private static SimpleDateFormat formatter=new SimpleDateFormat("yyyy-MM-dd")`
  Η μορφή "yyyy-MM-dd" εμφάνισης της ημερομηνίας
- `public double[][] A`
  Ο πίνακας αποστάσεων των πλοηγήσεων
- `public Vector patterns`
  Οι πλοηγήσεις εντός του χρονικού διαστήματος ενδιαφέροντος
- `private ConnectSQL con`
  Η σύνδεση με τη Βάση Δεδομένων
- `public Date StartSession, EndSession`

Οι χρονικές στιγμές έναρξης και λήξης, οι πλοηγήσεις που βρίσκονται εντός των οποίων εξετάζονται

Μέθοδοι:

`public Dist_Matrix(Date from, Date to)`

Ο κατασκευαστής της κλάσης αυτής. Οι ημερομηνίες εντός των οποίων τις πλοηγήσεις θα εξετάσει είναι οι from (για την έναρξη της πλοήγησης) και to(για τη λήξη της πλοήγησης). Υπολογίζει τις αποστάσεις των πλοηγήσεων, αφού πρώτα διαγράψει τις null πλοηγήσεις, με βάση τη δομική απόσταση, καλώντας τη μέθοδο Struct_Dist και τις καταχωρεί σε πίνακα.

5.2.2.15 public class MSTree

Η κλάση αυτή είναι από τις σημαντικότερες του συστήματος. Υλοποιεί τον τροποποιημένο αλγόριθμο της Voorhees [Voo85], και αναπτύχθηκε αναλυτικά σε προηγούμενη παράγραφο. Ουσιαστικά, ο αλγόριθμος αυτός βρίσκει το ελάχιστο συνεκτικό δένδρο (ΕΣΔ) ενός γράφου. Στη συγκεκριμένη διπλωματική εργασία, ο γράφος αυτός έχει ως κόμβους τις πλοηγήσεις των χρηστών και ως ακμές τις μεταξύ τους δομικές αποστάσεις.

Πεδία:

- `private ConnectSQL con`
  Η σύνδεση με τη Βάση Δεδομένων
- `public Vector hierarchy`
  Το ΕΣΔ που επιστρέφει ο αλγόριθμος.
- `public Vector patterns`
  Οι πλοηγήσεις που αποτελούν το γράφο, του οποίου το ΕΣΔ υπολογίζεται.
- `public Date start`
  Η ημερομηνία ενδιαφέροντος έναρξης.
- `public Date end`
  Η ημερομηνία ενδιαφέροντος λήξης.
- `private int NextDid`
  Ο επόμενος κόμβος που εισάγεται στο ΕΣΔ.
- `private int CurrentDid`
  Ο τρέχον κόμβος του ΕΣΔ που εξετάζεται. Από αυτόν υπολογίζονται οι αποστάσεις από όλους τους υπόλοιπους που δεν έχουν ακόμα εισαχθεί στο ΕΣΔ.
- `private double MinSim`
  Η ελάχιστη απόσταση μεταξύ των κόμβων-πλοηγήσεων. Αρχικοποιείται με 1.
- `final int Undef=-1`
  Η ένδειξη του ότι κάποιος κόμβος δεν έχει εισαχθεί στο ΕΣΔ.
- `private double[] sims`
  Ο πίνακας των αποστάσεων των υπολοίπων κόμβων του ΕΣΔ από τον τρέχοντα κόμβο.
- `private Dist_Matrix dist`

Στιγμιότυπο της κλάσης Dist_Matrix. Χρησιμεύει για την επιλογή των πλοηγήσεων που βρίσκονται μέσα στο χρονικό διάστημα που ενδιαφέρει το διαχειριστή.

- `private Info[] Entry`

  Ο πίνακας από δομές τύπου Info, που περιέχει τις απαραίτητες πληροφορίες για έναν κόμβο: Τον αύξοντα αριθμό του, την απόστασή του από το ΕΣΔ (δηλαδή την ελάχιστη απόσταση από κάποιον από τους κόμβους του), το αν έχει εισαχθεί ή όχι στο ΕΣΔ και τον κόμβο-πατέρα του, δηλαδή τον κόμβο του ΕΣΔ από τον οποίον απέχει την ελάχιστη απόσταση.

- `private SubjectSL Triple`

  Η τριπλέτα η οποία αντιστοιχεί σε μία εισαγωγή στο ΕΣΔ. Υλοποιείται με μία δομή τύπου SubjectSL.

Μέθοδοι:

- `public MSTree(Date from, Date to)`

  Ο κατασκευαστής της κλάσης. Ορίζει τις ημερομηνίες εντός των οποίων οι πλοηγήσεις θα εξεταστούν.

- `public Vector Algo1(Vector patterns, double[][] A)`

  Πραγματοποιεί τη σύνδεση με τη Βάση Δεδομένων. Καθορίζει ποιος θα είναι ο επόμενος κόμβος-πλοήγηση από τα patterns που θα εισαχθεί στο ΕΣΔ. Ο A είναι ο πίνακας αποστάσεων των πλοηγήσεων.

- `public double[] ComputeSims(int i, int size, double[][] AA)`

  Επιστρέφει τις αποστάσεις του κόμβου i από τους υπόλοιπους κόμβους, πλήθους size του γράφου, με βάση τον πίνακα αποστάσεων AA.

- `public void InsertHierarchy(int ND, int nnND, double MS)`

  Εισάγει στο ΕΣΔ τον κόμβο ND, ο οποίος έχει πατέρα τον nnND και η ελάχιστη απόστασή του από το ΕΣΔ είναι MS. Η μορφή εισαγωγής είναι μορφή Triple.

5.2.2.16 public class SubjectSL

Η κλάση αυτή ορίζει μία δομή η οποία αποτελεί τη μορφή με την οποία εισάγονται οι κόμβοι στο ΕΣΔ. Χρησιμοποιείται από την κλάση MSTree.

Πεδία:

- `int x`

  Ο κόμβος που εισάγεται στο ΕΣΔ.

- `int y`

  Ο πατέρας του κόμβου x. Είναι αυτός ο κόμβος του ΕΣΔ από τον οποίο ο x απέχει τη μικρότερη απόσταση.

- `double value`

  Η απόσταση από το ΕΣΔ του κόμβου x.

Μέθοδοι:

`public SubjectSL (int xi, int yi, double v)`

Ο κατασκευαστής της κλάσης

5.2.2.17 public class ConnectedComponents

Η μέθοδος αυτή, όπως έχει αναφερθεί σε προηγούμενο κεφάλαιο[ ], βρίσκει τις συνεκτικές συνιστώσες ενός ΕΣΔ. Μαζί με τις Dist_Matrix και MSTree, υλοποιεί την τεχνική «μονός σύνδεσμος».

Πεδία:

- `private ConnectSQL con`
  Η σύνδεση με τη Βάση Δεδομένων
- `private static Vector collection`
  Περιλαμβάνει τα στοιχεία μιας συστάδας.
- `private static Vector Clusters`
  Οι συστάδες που επιστρέφει ο αλγόριθμος
- `private static Vector Clustered_Nodes`
  Οι συστάδες που βρίσκει ο αλγόριθμος, οι πλοηγήσεις όμως εδώ αντιπροσωπεύονται από κόμβους
- `private static Vector Names`
  Τα ονόματα των χρηστών που πραγματοποίησαν τις πλοηγήσεις μιας συστάδας, χωρίς επαναλήψεις.
- `private static Vector NamesInDetail`
  Τα ονόματα των χρηστών που πραγματοποίησαν τις πλοηγήσεις μιας συστάδας, με δυνατότητα επαναλήψεων. Υπάρχει δηλαδή ένα προς ένα αντιστοιχία πλοήγησης-χρήστη.
- `public double k`
  Το επίπεδο συσταδοποίησης
- `public Date StartSession`
  Η ημερομηνία ενδιαφέροντος έναρξης
- `public Date EndSession`
  Η ημερομηνία ενδιαφέροντος λήξης
- `private Dist_Matrix dist;`
  Στιγμιότυπο της κλάσης Dist_Matrix. Χρησιμεύει για την επιλογή των πλοηγήσεων που βρίσκονται μέσα στο χρονικό διάστημα που ενδιαφέρει το διαχειριστή.
- `private Komvos Vertex`
  Ο Vertex είναι μια δομή τύπου Komvos. Περιέχει για έναν κόμβο τις εξής πληροφορίες:τον αριθμό που του αντιστοιχεί, τον κόμβο (τον αριθμό του) με τον οποίο βρίσκεται στην ίδια συνεκτική συνιστώσα, και μια ένδειξη του αν έχει εισαχθεί σε κάποια συνεκτική συνιστώσα ή όχι.

Μέθοδοι:

- `public  Vector getClustered_Nodes()`
  Επιστρέφει τα Clustered_Nodes
- `public Vector getNames()`
  Επιστρέφει τα Names
- `public  Vector getNamesInDetail()`

Επιστρέφει τα NamesInDetail

- `public ConnectedComponents(double kappa, Date from, Date to)`

  Ο κατασκευαστής της κλάσης. Δέχεται ως παραμέτρους το επίπεδο συσταδοποίησης kappa, την ημερομηνία ενδιαφέροντος έναρξης from και την ημερομηνία ενδιαφέροντος λήξης to.
- `public Vector Connected()`

  Η κύρια μέθοδος της κλάσης. Εξάγει από το ΕΣΔ τις συνεκτικές συνιστώσες και επιστρέφει τις συστάδες που αντιστοιχούν στις πλοηγήσεις.

5.2.2.18 public class Info

Δομή που χρησιμοποιείται από την κλάση MSTree για την αποθήκευση κατάλληλης πληροφορίας, όπως έχει προαναφερθεί.

Πεδία:

- `int i`
- `double sim;`
- `boolean InHierarchy;`
- `int nn;`

Μέθοδοι:

`public Info(int ii, double s, boolean InH, int n)`

Ο κατασκευαστής της κλάσης

5.2.2.19 public class Komvos

Δομή που χρησιμοποιείται από την κλάση ConnectedComponents για την αποθήκευση κατάλληλης πληροφορίας, όπως έχει προαναφερθεί.

Πεδία:

- `int vertex`

  Ο εκάστοτε κόμβος
- `int findset`

  Κόμβος με τον οποίο ο vertex βρίσκεται στην ίδια συστάδα, και ειδικότερα αυτός από τον οποίον απέχει τη λιγότερη απόσταση (ο πατέρας του-αυτός εξαιτίας του οποίου ο vertex τοποθετήθηκε στη συγκεκριμένη συστάδα)
- `boolean inset`

  Ένδειξη του αν ο vertex έχει ήδη καταχωρηθεί σε κάποια συστάδα.

Μέθοδοι:

`public Komvos(int v, int fs, boolean in)`

Ο κατασκευαστής της κλάσης

5.2.2.20 public class CIndex

Η κλάση αυτή εφαρμόζει τον αλγόριθμο CIndex στα δεδομένα της Βάσης και υπολογίζει το βέλτιστο επίπεδο συσταδοποίησης αυτών με βάση την τεχνική «μονός σύνδεσμος».

Πεδία:

- `private ConnectSQL con`
  Η σύνδεση με τη Βάση Δεδομένων
- `private int nd`
  Το πλήθος των αποστάσεων (ανά δύο στοιχεία) ανά συστάδα της κάθε συσταδοποίησης
- `private double Sum`
  Το άθροισμα των αποστάσεων (ανά δύο στοιχεία) όλων των στοιχείων της κάθε συστάδας.
- `private double step`
  Το επίπεδο συσταδοποίησης που εξετάζει ο αλγόριθμος σε κάθε επανάληψη
- `private double mindw`
  Το άθροισμα των nd μικρότερων αποστάσεων (ανά δύο στοιχεία) σε όλο το σύνολο των στοιχείων.
- `private double maxdw`
  Το άθροισμα των nd μεγαλύτερων αποστάσεων (ανά δύο στοιχεία) σε όλο το σύνολο των στοιχείων.
- `public Date StartSession`
  Η ημερομηνία ενδιαφέροντος έναρξης
- `public Date EndSession`
  Η ημερομηνία ενδιαφέροντος λήξης
- `private double[] min`
  Ο πίνακας με τα nd μικρότερα στοιχεία
- `private double[] max`
  Ο πίνακας με τα nd μεγαλύτερα στοιχεία
- `private double[][] A`
  Ο πίνακας δομικών αποστάσεων μεταξύ όλων των πλοηγήσεων
- `private double[] a`
  Πίνακας που περιέχει όλα τα στοιχεία του A, χωρίς επαναλήψεις (δηλαδή δεν περιέχεται και το στοιχείο A[1][2] και το A[2][1], τα οποία, λόγω συμμετρίας του πίνακα A, είναι ίδια.
- `private Vector Clusters`
  Οι συστάδες που επιστρέφονται από τον αλγόριθμο ConnectedComponents
- `private Vector indexVec`
  Διάνυσμα που περιέχει δομές τύπου Index. Σε κάθε δομή περιέχεται το επίπεδο συσταδοποίησης και η τιμή δείκτη που αντιστοιχεί στο επίπεδο αυτό.
- `private Vector C`
  Περιέχει τα στοιχεία της κάθε συστάδας.

Μέθοδοι

- `public CIndex(Date from, Date to)`
  Ο κατασκευαστής της κλάσης. Ο αλγόριθμος εξετάζει τις πλοηγήσεις που βρίσκονται μεταξύ των ημερομηνιών from και to.
- `public double Prog ()`
  Ο αλγόριθμος C-Index, όπως επεξηγήθηκε στην ενότητα 5.2.2.3

5.2.2.21 public class Index

Η κλάση αυτή υλοποιεί μια δομή που αποθηκεύει την τιμή δείκτη του διανύσματος CIndex που αντιστοιχεί στο επίπεδο συσταδοποίησης. Χρησιμοποιείται από τον αλγόριθμο CIndex για τον υπολογισμό του βέλτιστου επιπέδου συσταδοποίησης.

Πεδία:

- `double k`
  Το επίπεδο συσταδοποίησης
- `double i`
  Η τιμή δείκτη που αντιστοιχεί στο επίπεδο συσταδοποίησης

Μέθοδοι:

`public Index (double ind, double kappa)`
Ο κατασκευαστής της κλάσης

5.2.2.22 public class BubblesortIndex

Η κλάση αυτή πραγματοποιεί ταξινόμηση των τιμών δείκτη του διανύσματος CIndex κατά αύξουσα διάταξη. Η ταξινόμηση πραγματοποιείται πάνω σε δομές τύπου CIndex, οι οποίες είναι αποθηκευμένες σε ένα διάνυσμα, με βάση την τιμή του πεδίου δείκτη (πεδίο i). Επιστρέφει το επίπεδο συσταδοποίησης που αντιστοιχεί σε αυτήν την τιμή δείκτη, το οποίο είναι και το προτεινόμενο στο διαχειριστή επίπεδο συσταδοποίησης από τον αλγόριθμο CIndex.

Πεδία:

- `private static int n`
  Το μέγεθος του διανύσματος προς ταξινόμηση.

Μέθοδοι:

- `public  Index bubblesort (Vector a)`
  Η μέθοδος αυτή πραγματοποιεί ταξινόμηση των δομών τύπου Index που αποτελούν το διάνυσμα a, με βάση την τιμή δείκτη (πεδίο i).

5.2.2.23 public class KMeansUI extends JDialog implements ActionListener

Η κλάση αυτή είναι υπεύθυνη για την παρουσίαση στο χρήστη της φόρμας επιλογών που του παρέχονται με τον αλγόριθμο συσταδοποίησης K Means. Καλεί τις κλάσεις υλοποίησης του αλγορίθμου αυτού και παρουσιάζει τα αποτελέσματα στο διαχειριστή μέσω της διαπροσωπείας.

Πεδία:

- `private static DateFormat formatter=new SimpleDateFormat("yyyy-MM-dd")`
  Η μορφή "yyyy-MM-dd" εμφάνισης της ημερομηνίας
- `private SimpleDateFormat formatter2=new SimpleDateFormat("dd/MM/yyyy")`
  Η μορφή "dd/MM/yyyy" εμφάνισης της ημερομηνίας
- `private static Date start=null`
  Η ημερομηνία ενδιαφέροντος έναρξης
- `private static Date end=null`
  Η ημερομηνία ενδιαφέροντος λήξης
- `public static Menu parent2`
  Ο πατέρας του τρέχοντος Dialog Box
- `private static String st1`
  Η ημέρα από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st2`
  Ο μήνας από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st3`
  Το έτος από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st4`
  Η ημέρα από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static String st5`
  Ο μήνας από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static String nst6`
  Το έτος από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private JLabel label1, label2, label22, label23, label24, label3, label33, label34, label35,mhn`
  Βοηθητικές ετικέτες για την εισαγωγή πληροφοριακού κειμένου
- `private JComboBox yearbox1`
  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox yearbox2`
  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος λήξης
- `private JComboBox monthbox1`
  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox monthbox2`
  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος λήξης
- `private JComboBox daybox1`

JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος έναρξης

- `private JComboBox daybox1`

  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος λήξης
- `private JButton ok`

  Το κουμπί OK
- `private JButton cancel`

  Το κουμπί Cancel
- `private static JTextArea path`

  Η περιοχή εμφάνισης των αποτελεσμάτων
- `private JScrollPane scroll`

  Η γραμμή κύλισης της περιοχής εμφάνισης των αποτελεσμάτων
- `public static Menu father`

  Ο πατέρας της κλάσης που κάλεσε την τρέχουσα
- `private static JDialog parent2`

  Ο πατέρας της κλάσης αυτής
- `private JTextField data`

  Το πεδίο όπου ο διαχειριστής εισάγει το επίπεδο συσταδοποίησης
- `private static ConnectSQL cp`

  Η σύνδεση με τη Βάση Δεδομένων
- `private double number`

  Το πλήθος των συστάδων που εισάγει ο χρήστης
- `private KMeans kmeans`

  Αντικείμενο της κλάσης KMeans
- `private String no`

  Ο αριθμός που δίνει ο χρήστης σε μορφή String, πριν δηλαδή μετατραπεί σε ακέραιο.
- `private Vector clustered_patterns`

  Οι συστάδες των πλοηγήσεων
- `private Vector names`

  Τα ονόματα των χρηστών που έχουν πραγματοποιήσει τις πλοηγήσεις, από μια φορά το καθένα.
- `private Vector namesNext`

  Τα ονόματα των χρηστών που έχουν πραγματοποιήσει τις πλοηγήσεις σε ένα προς ένα αντιστοιχία με αυτές.

Μέθοδοι:

- `void buildConstraints(GridBagConstraints gbc, int gx, int gy, int gw, int gh, int wx, int wy)`

  Μέθοδος υπεύθυνη για τη δημιουργία του πλέγματος GridBagLayout που χρησιμοποιείται για τη διάταξη των συστατικών.
- `public SLinkUI(JDialog parent, Menu newFrame, String title, boolean modal, JTextArea textarea, ConnectSQL con)`

  Ο κατασκευαστής της κλάσης. Είναι υπεύθυνος για τη διάταξη των συστατικών στο παράθυρο της διαπροσωπείας χρήστη. Ο πατέρας της κλάσης αυτής είναι αντικείμενο της κλάσης Clustering που περιγράφηκε παραπάνω. Επίσης χρειάζεται να κρατηθεί πληροφορία και για τον πατέρα newFrame της κλάσης Clustering, που είναι τύπου Menu, γιατί εκεί θα επιστραφούν τα αποτελέσματα. Η σύνδεση με τη Βάση

Δεδομένων είναι η con, ο τίτλος του παραθύρου είναι title, η μεταβλητή modal είναι true ή false ανάλογα με το αν κλειδώνει ή όχι το παράθυρο-πατέρας και, τέλος, η περιοχή εμφάνισης των αποτελεσμάτων είναι η textarea.

- `private class Handler implements ActionListener{`
  `public void actionPerformed( ActionEvent event )`
  Η μέθοδος αυτή είναι υπεύθυνη για την απόκριση στο πάτημα των κουμπιών OK και Cancel από το διαχειριστή. Συνδέεται με την κλάση KMeans για την εφαρμογή του αλγορίθμου των K-Μέσων και επιστρέφει τα αποτελέσματα στο διαχειριστή. Είναι υπεύθυνη και για την εμφάνιση των ονομάτων των χρηστών που έχουν πραγματοποιήσει τις πλοηγήσεις σε κάθε συστάδα. Πραγματοποιεί όλους τους απαραίτητους ελέγχους του πλήθους των συστάδων που εισάγει ο διαχειριστής. Επίσης κλείνει τη σύνδεση με τη Βάση Δεδομένων.
- `protected void processWindowEvent(WindowEvent e)`
  Μέθοδος που χειρίζεται το κλείσιμο του παραθύρου.
- `public static Date StartSession()`
  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος έναρξης
- `public static Date EndSession()`
  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος λήξης
- `public void actionPerformed (ActionEvent evt)`
  Η μέθοδος αυτή αποκρίνεται στην επιλογή των ημερομηνιών ενδιαφέροντος από το διαχειριστή και μετατρέπει σε Strings τις επιλογές του έτσι ώστε να είναι σε κατάλληλη για επεξεργασία μορφή από τις μεθόδους EndSession και StartSession

5.2.2.24 public class KMeans

Η κλάση αυτή είναι υπεύθυνη για την εφαρμογή του αλγορίθμου των K Μέσων για τη συσταδοποίηση των πλοηγήσεων, ο οποίος είναι τροποποιημένος για να υπολογίζει αποστάσεις μεταξύ Strings και όχι μεταξύ διανυσμάτων. Επιστρέφει τις συστάδες των πλοηγήσεων.

Πεδία:

- `private ConnectSQL con`
  Η σύνδεση με τη Βάση Δεδομένων
- `private int K`
  Το πλήθος των εξαγόμενων συστάδων, αριθμός που δίνεται από το χρήστη.
- `public Date StartSession`
  Η ημερομηνία έναρξης
- `public Date EndSession`
  Η ημερομηνία λήξης
- `private Vector v`
  Οι συστάδες που επιστρέφει ο αλγόριθμος
- `private Vector Names`
  Τα ονόματα των χρηστών που έχουν πραγματοποιήσει τις πλοηγήσεις, από μια φορά το καθένα.
- `private Vector NamesNext`

Τα ονόματα των χρηστών που έχουν πραγματοποιήσει τις πλοηγήσεις, σε ένα προς ένα αντιστοιχία με τις πλοηγήσεις (με δυνατότητα επαναλήψεων δηλαδή).

- `private Vector patterns`

  Οι πλοηγήσεις που εξετάζει ο αλγόριθμος.

- `private double[][] A`

  Ο πίνακας δομικών αποστάσεων όλων των πλοηγήσεων

- `private double[][] D`

  Ο πίνακας που περιέχει τις αποστάσεις όλων των προς εξέταση πλοηγήσεων από το κέντρο της κάθε συστάδας

- `private int[] G`

  Φυλάσσει την προηγούμενη πληροφορία σχετικά με το σε ποια συστάδα ανήκει η κάθε πλοήγηση

- `private int[] G1`

  Φυλάσσει τη νέα πληροφορία σχετικά με το σε ποια συστάδα ανήκει η κάθε πλοήγηση

- `private int[] Z`

  Αποθηκεύει τα παλιά κέντρα των συστάδων

- `private int[] Z1`

  Αποθηκεύει τα νέα κέντρα των συστάδων

- `private int[] N`

  Αποθηκεύει το πλήθος των πλοηγήσεων που περιέχει η κάθε συστάδα (έχει μέγεθος όσες και οι συστάδες)

- `private Subject[] Entry`

  Πίνακας δομών τύπου Subject. Αποθηκεύει πληροφορία σχετική με την απόσταση της κάθε πλοήγησης από την κάθε συστάδα και τη συστάδα αυτή. Στη συνέχεια ταξινομείται με χρήση της Bubblesort (και συγκεκριμένα από το στιγμιότυπο bubble αυτής-παρουσιάζεται παρακάτω)

- `private Subject[] Means`

  Πίνακας δομών τύπου Subject. Αποθηκεύει του μέσους όρους των αποστάσεων όλων των πλοηγήσεων που ανήκουν σε κάθε συστάδα από την κάθε πλοήγηση αυτής. Στη συνέχεια ταξινομείται με χρήση της Bubblesort (και συγκεκριμένα από το στιγμιότυπο bubble2 αυτής-παρουσιάζεται παρακάτω) για να βρεθεί το νέο κέντρο της κάθε συστάδας.

- `private Bubblesort bubble`

  Στιγμιότυπο της κλάσης Bubblesort. Χρησιμεύει για να βρεθεί η συστάδα εκείνη από την οποία η κάθε πλοήγηση απέχει λιγότερο.

- `private Bubblesort bubble2`

  Στιγμιότυπο της κλάσης Bubblesort. Χρησιμεύει για την εύρεση της πλοήγησης εκείνης που θα αποτελέσει το κέντρο της κάθε συστάδας.

- `private Dist_Matrix dist`

  Στιγμιότυπο της κλάσης Dist_Matrix. Χρησιμεύει για την επιλογή των πλοηγήσεων που βρίσκονται μέσα στο χρονικό διάστημα που ενδιαφέρει το διαχειριστή.

Μέθοδοι:

- `public Vector getNames()`

  Μέθοδος για την επιστροφή των Names

- `public Vector getNamesNext()`

Μέθοδος για την επιστροφή των NamesNext

- `public KMeans(int kappa, Date from Date to)`
  Ο κατασκευαστής της κλάσης. Δέχεται σαν όρισμα το πλήθος kappa των συστάδων και τις ημερομηνίες έναρξης (from) και λήξης (to)
- `public Vector Algo()`
  Η κλάση που υλοποιεί τον αλγόριθμο των Κ Μέσων

5.2.2.25 public class Subject

Δομή που χρησιμοποιείται από την κλάση KMeans για την αποθήκευση κατάλληλης πληροφορίας όπως έχει προαναφερθεί.

Πεδία:

- double value
  Η απόσταση της πλοήγησης από τη συστάδα cluster.
- int cluster
  Η συστάδα

Μέθοδοι:

public Subject (double v, int c)
Ο κατασκευαστής αντικειμένων της κλάσης

5.2.3.26 public class Bubblesort

Η κλάση αυτή βασίζεται στη μέθοδο Bubblesort για την ταξινόμηση ενός πίνακα από δομές τύπου Subject.

Πεδία:

private int n
Το μέγεθος του πίνακα από δομές τύπου Subject

Μέθοδοι:

public int bubblesort (Subject [] a)
Η μέθοδος αυτή εξετάζει το πεδίο value όλων των δομών Subject που αποτελούν τον πίνακα a και με βάση αυτό ταξινομεί τις δομές αυτές κατά αύξοντα τιμή της value. Επιστρέφει τη συστάδα στην οποία αντιστοιχεί η μικρότερη τιμή value.

5.2.2.27 public class UndecidedUI extends JDialog implements ActionListener

Η κλάση αυτή είναι υπεύθυνη για την εμφάνιση στο χρήστη της φόρμας επιλογής του πλήθους των αναποφάσιστων χρηστών του συστήματος. Επιστρέφει τα αποτελέσματα σε κατάλληλη φόρμα, μέσω της διαπροσωπείας του διαχειριστή.

Πεδία:

- `private static DateFormat formatter=new SimpleDateFormat("yyyy-MM-dd")`

  Η μορφή "yyyy-MM-dd" εμφάνισης της ημερομηνίας
- `private SimpleDateFormat formatter2=new SimpleDateFormat("dd/MM/yyyy")`

  Η μορφή "dd/MM/yyyy" εμφάνισης της ημερομηνίας
- `private static Date start=null`

  Η ημερομηνία ενδιαφέροντος έναρξης
- `private static Date end=null`

  Η ημερομηνία ενδιαφέροντος λήξης
- `public Menu parent2`

  Ο πατέρας του τρέχοντος Dialog Box
- `private static String st1`

  Η ημέρα από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st2`

  Ο μήνας από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st3`

  Το έτος από την ημερομηνία έναρξης που επιλέγει ο διαχειριστής
- `private static String st4`

  Η ημέρα από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static String st5`

  Ο μήνας από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private static String st6`

  Το έτος από την ημερομηνία λήξης που επιλέγει ο διαχειριστής
- `private JScrollPane myScrollPane`

  Ο κυλιστής που προσαρμόζεται στη φόρμα επιστροφής των αποτελεσμάτων
- `private JLabel label1, label2, label22, label23, label24, label3, label33, label34, label35, label1neo, label2neo`

  Βοηθητικές ετικέτες για την εισαγωγή πληροφοριακού κειμένου
- `private JComboBox yearbox1`

  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox yearbox2`

  JComboBox για την επιλογή του έτους στην ημερομηνία ενδιαφέροντος λήξης
- `private JComboBox monthbox1`

  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox monthbox2`

  JComboBox για την επιλογή του μήνα στην ημερομηνία ενδιαφέροντος λήξης
- `private JComboBox daybox1`

  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος έναρξης
- `private JComboBox daybox1`

  JComboBox για την επιλογή της ημέρας στην ημερομηνία ενδιαφέροντος λήξης
- `private JButton ok`

Το κουμπί OK

- `private JButton cancel`

  Το κουμπί Cancel

- `private JTextArea path`

  Η περιοχή εμφάνισης των αποτελεσμάτων

- `private JScrollPane scroll`

  Ο κυλιστής της περιοχής εμφάνισης των αποτελεσμάτων

- `private ConnectSQL cp`

  Η σύνδεση με τη Βάση Δεδομένων

- `private boolean flag`

  Μεταβλητή που γίνεται true όταν το νούμερο που εισάγει ο διαχειριστής στο πεδίο εισαγωγής του πλήθους των αναποφάσιστων χρηστών δεν είναι σωστό.

- `private int number`

  Το νούμερο των αναποφάσιστων χρηστών που εισάγει ο διαχειριστής.

- `private String no`

  Το νούμερο που εισάγει ο διαχειριστής πριν ακόμα μετατραπεί σε ακέραιο.

- `private Vector result`

  Οι περισσότερο αναποφάσιστοι χρήστες του συστήματος και το πλήθος των Back και Forward που έχουν πραγματοποιήσει.

- `private User[] us`

  Πίνακας από δομές τύπου User.

- `private Task6 task6`

  Στιγμιότυπο της κλάσης Task6.

- `private Bubblesort6 bubble6`

  Αντικείμενο της κλάσης Bubblesort6. Πραγματοποιεί την ταξινόμηση των χρηστών.

Μέθοδοι:

- `void buildConstraints(GridBagConstraints gbc, int gx, int gy, int gw, int gh, int wx, int wy)`

  Μέθοδος υπεύθυνη για τη δημιουργία του πλέγματος GridBagLayout που χρησιμοποιείται για τη διάταξη των συστατικών.

- `public UndecidedUI(Menu parent, String title, boolean modal, JTextArea textarea)`

  Ο κατασκευαστής της κλάσης. Είναι υπεύθυνος για τη διάταξη των συστατικών στο παράθυρο της διαπροσωπείας χρήστη. Ο πατέρας της κλάσης αυτής είναι αντικείμενο της κλάσης Menu που περιγράφηκε παραπάνω. Ο τίτλος του παραθύρου είναι title, η μεταβλητή modal είναι true ή false ανάλογα με το αν κλειδώνει ή όχι το παράθυρο-πατέρας και, τέλος, η περιοχή εμφάνισης των αποτελεσμάτων είναι η textarea.

- `private class Handler implements ActionListener{`
  `public void actionPerformed (ActionEvent event)`

  Η μέθοδος αυτή είναι υπεύθυνη για την απόκριση στο πάτημα των κουμπιών OK και Cancel από το διαχειριστή. Συνδέεται με την κλάση Task6 για τον υπολογισμό των πιο αναποφάσιστων χρηστών του συστήματος. Είναι επίσης υπεύθυνη και για την εμφάνιση των αποτελεσμάτων στην κατάλληλη φόρμα. Πραγματοποιεί όλους τους απαραίτητους ελέγχους του αριθμού που εισάγει ο διαχειριστής. Τέλος, κλείνει τη σύνδεση με τη Βάση Δεδομένων.

- `protected void processWindowEvent(WindowEvent e)`

Μέθοδος που χειρίζεται το κλείσιμο του παραθύρου.

- `public static Date StartSession()`

  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος έναρξης

- `public static Date EndSession()`

  Η μέθοδος αυτή επιστρέφει την ημερομηνία ενδιαφέροντος λήξης

- `public void actionPerformed (ActionEvent evt)`

  Η μέθοδος αυτή αποκρίνεται στην επιλογή των ημερομηνιών ενδιαφέροντος από το διαχειριστή και μετατρέπει σε Strings τις επιλογές του έτσι ώστε να είναι σε κατάλληλη για επεξεργασία μορφή από τις μεθόδους EndSession και StartSession

5.2.2.28 public class Task6

Η κλάση αυτή βρίσκει για κάθε χρήστη πόσα Back και Forward έχει πραγματοποιήσει μέσα στο χρονικό διάστημα ενδιαφέροντος.

Πεδία:

- `private static ConnectSQL con`

  Η σύνδεση με τη Βάση Δεδομένων

- `private static DateFormat formatter=new SimpleDateFormat("yyyy-MM-dd")`

  Η μορφή εμφάνισης της ημερομηνίας "yyyy-MM-dd"

- `private static Date StartSession`

  Η ημερομηνία έναρξης

- `private static EndSession`

  Η ημερομηνία λήξης.

- `private static Vector patterns`

  Οι πλοηγήσει ομαδοποιημένες (δηλαδή χωρίς επαναλήψεις) που έχουν πραγματοποιηθεί στη ζητούμενη χρονική περίοδο.

- `private static Vector userpat`

  Οι χρήστες που έχουν πραγματοποιήσει μια συγκεκριμένη πλοήγηση.

- `private static Vector usernames`

  Τα ονόματα όλων των χρηστών του συστήματος.

- `private static User[] Users`

  Πίνακας δομών τύπου User.

- `private static Undecided undec`

  Αντικείμενο της κλάσης Undecided.

Μέθοδοι:

- `public Task6(ConnectSQL cp, Date from, Date to)`

  Ο κατασκευαστής της κλάσης.

- `public static User[] basic()`

  Η μέθοδος αυτή εξετάζει όλες τις πλοηγήσεις εντός του ζητούμενου χρονικού διαστήματος και βρίσκει για κάθε μία από αυτές πόσους κόμβους Back και Forward περιέχει. Στη συνέχεια, για κάθε πλοήγηση βρίσκει ποιος χρήστης την έχει πραγματοποιήσει και αυξάνει ανάλογα το πλήθος των Back και Forward που έχει

επιτελέσει. Επιστρέφει έναν πίνακα από δομές τύπου User, οι οποίες περιέχουν το χρήστη και το πλήθος από τα Back και Forward που αυτός έχει πραγματοποιήσει.

5.2.2.29 public class User

Η κλάση αυτή χρησιμοποιείται βοηθητικά από τις κλάσεις Task6 και Undecided. Αποθηκεύει για κάθε χρήστη του συστήματος το πλήθος των Back και Forward που έχει πραγματοποιήσει.

Πεδία:

- `String fullname`
  Το ονοματεπώνυμο του χρήστη
- `int count`
  Το πλήθος των Back και Forward που έχει πραγματοποιήσει

Μέθοδοι:

public User (String fn, int co)
Ο κατασκευαστής της κλάσης.

5.2.2.30 public class Undecided

Η κλάση αυτή ευθύνεται για την εύρεση των κόμβων Back και Forward που υπάρχουν σε μια πλοήγηση.

Πεδία:

- `public ConnectSQL con`
  Η σύνδεση με τη Βάση Δεδομένων.
- `private String path`
  Η πλοήγηση όπως λαμβάνεται από τη Βάση Δεδομένων
- `private User[] Patterns`
  Χρησιμοποιείται η δομή User για να αποθηκεύσει τη σελίδα που συναντάται μέσα στην πλοήγηση και τον αύξοντα αριθμό αυτής μέσα στην πλοήγηση.
- `private int[] BF`
  Ο πίνακας αυτός έχει μέγεθος όσες και οι σελίδες της πλοήγησης και για κάθε μία από τις σελίδες-κόμβους περιέχει τον κόμβο με τον οποίο αποτελεί Back και Forward. Σημειώνεται ότι αν ένας κόμβος αποτελεί Back και Forward με έναν άλλον, αυτός θα είναι και ο μοναδικός.
- `public counter`
  Το πλήθος των Back και Forward που υπάρχουν στην πλοήγηση
- private static Vector valueVector2
  Η λίστα η οποία αποθηκεύει για κάθε σελίδα σε ποια σημεία της πλοήγησης συναντάται.
- `private Vector v`

Το κάθε στοιχείο του είναι και μια σελίδα της πλοήγησης (Η πλοήγηση είναι τώρα σε μορφή χωρίς τα «/» που διαχωρίζουν τις σελίδες).

- `private Vector Possible`

  Περιέχει τα πιθανά Back και Forward

- `private Vector Sorted`

  Είναι ο Vector Possible ταξινομημένος κατά αύξουσα σειρά απόστασης κόμβων, οι οποίοι αποτελούν πιθανά Back και Forward.

- `private Hashtable adjHash`

  Ο Hashtable που περιέχει για κάθε σελίδα της πλοήγησης το σημείο μέσα σε αυτήν στο οποίο συναντάται.

Μέθοδοι:

- `public Undecided(ConnectSQL cone, String pat)`

  Ο κατασκευαστής της κλάσης. Η πλοήγηση που εξετάζεται είναι η pat.

- `public Hashtable Hashes(Vector v)`

  Επιστρέφει τον Hashtable adjHash

- `public Vector FirstSearch(Hashtable hashtable)`

  Βρίσκει τα στοιχειώδη Back και Forward ενημερώνοντας παράλληλα τον πίνακα BF. Επίσης βρίσκει τους κόμβους που αποτελούν πιθανά Back και Forward και τους αποθηκεύει μαζί με τη μεταξύ τους απόσταση στο Vector Possible.

- `public void Algo(int int1,int int2)`

  Βρίσκει τα υπόλοιπα Back και Forward, είτε απευθείας, σε περίπτωση που πρόκειται για τους διπλανούς κόμβους των int1 και int2, είτε με κλήση της μεθόδου Algo2

- `public void Algo2(int n3,int n4,int n1,int n2)`

  Βρίσκει τα περισσότερο «κρυμμένα» Back και Forward.

- `public static Vector Bubblesort(Vector P)`

  Ταξινομεί τα πιθανά Back και Forward κατά αύξοντα αριθμό απόστασης κόμβων.

5.2.2.31 public class Bubblesort6

Ταξινομεί πίνακα από δομές τύπου User κατά αύξοντα αριθμό του πεδίου count. Χρησιμοποιείται από την κλάση UndecidedUI για να ταξινομήσει τους χρήστες με βάση το πλήθος των Back και Forward που έχουν πραγματοποιήσει.

Πεδία:

- `private static int k`

  Το πλήθος των περισσότερο αναποφάσιστων χρηστών που εισάγει ο διαχειριστής.

- `private  static int n`

  Το μέγεθος του πίνακα από δομές User που πρόκειται να ταξινομηθεί.

- `private static Vector firstElements`

  Τα k πρώτα στοιχεία τύπου User που προκύπτουν από την ταξινόμηση

Μέθοδοι:

- `public Bubblesort6(int k1)`

  Ο κατασκευαστής της κλάσης.

- `public static Vector bubblesort (User[] a)`

Ταξινομεί τον πίνακα από στοιχεία τύπου User κατά αύξοντα αριθμό του πεδίου count. Επιστέφει τα k πρώτα στοιχεία.

5.3 Πλατφόρμες και προγραμματιστικά εργαλεία

Στην ενότητα αυτή παρουσιάζονται τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής, καθώς και τις πλατφόρμες στις οποίες αυτή εκτελείται. Επίσης, αναφέρεται με ποιον τρόπο μπορεί κάποιος να εγκαταστήσει την εφαρμογή στο σύστημά του.

5.3.1 Γενικά

Η εφαρμογή που υλοποιήθηκε μπορεί να εκτελεστεί σε περιβάλλον Windows XP . Η γλώσσα προγραμματισμού που χρησιμοποιήθηκε είναι η Java2 και πιο συγκεκριμένα το j2sdk1.4.1_03. Η πλατφόρμα ανάπτυξης και εκτέλεσης της εφαρμογής είναι το περιβάλλον Eclipse. Για την πρόσβαση στη Βάση Δεδομένων χρησιμοποιήθηκαν JDBC οδηγοί. Τέλος, για τη Βάση Δεδομένων του συστήματος χρησιμοποιήθηκε η MySQL 4.1.7.

5.3.2 Εγκατάσταση Συστήματος

Για την πλήρη εγκατάσταση της εφαρμογής χρειάζεται να γίνουν κάποια βήματα. Καταρχήν είναι απαραίτητο να έχει εγκατασταθεί η Java2, κατά προτίμηση η έκδοση που αναφέρθηκε ή μεγαλύτερη. Επίσης πρέπει κάπου να υπάρχει εγκατεστημένη και η MySQL, αν όχι στο ίδιο PC, τότε σε κάποιο που να μπορεί να προσδιοριστεί με ένα IP address. Στη συνέχεια πρέπει να τοποθετηθούν οι απαραίτητες βιβλιοθήκες κλάσεων που χρησιμοποιήθηκαν (τα αρχεία .jar) στους κατάλληλους φακέλους, έτσι ώστε να μπορούν να χρησιμοποιηθούν από τη Virtual Machine της Java. Τα αρχεία αυτά είναι τα υπεύθυνα για τη σύνδεση με τη MySQL, υλοποιούν δηλαδή τον driver για την επικοινωνία μέσω JDBC με τη MySQL, και είναι τα

- mysql –connector-java3.0.16-ga-bin.jar
- jdbc2_0-stdext.jar
- jta-spec1_0_1.jar

Τοποθετούνται στο φάκελο C:\Program Files\Java\j2re1.4.1_03\lib\ext.

Το σύστημα που κατασκευάστηκε βρίσκεται σε δύο φακέλους, οι οποίοι αντιστοιχούν σε Projects του Eclipse: Τον User, ο οποίος αντιστοιχεί στο υποσύστημα του χρήστη και το Manager, ο οποίος αντιστοιχεί στο υποσύστημα του διαχειριστή. Οι φάκελοι αυτοί έχουν τους υπό-φακέλους user και manager αντίστοιχα, μέσα στους οποίους βρίσκονται τα .java και

.class αρχεία του κώδικα υλοποίησης. Για να τρέξει κάποιος την εφαρμογή πρέπει να ακολουθήσει τα εξής βήματα:

1) Αν είναι ο χρήστης:
    - Άνοιγμα μιας γραμμής εντολών (Command prompt)
    - Μεταφορά στη θέση που έχει τοποθετήσει το φάκελο-Project User
    - Εκτέλεση της εντολής java user.UserLogin. Από την κλάση αυτή θα οδηγηθεί κατάλληλα στις επόμενες κλάσεις του υποσυστήματος αυτού.
2) Αν είναι ο διαχειριστής:
    - Άνοιγμα μιας γραμμής εντολών (Command prompt)
    - Μεταφορά στη θέση που έχει τοποθετήσει το φάκελο-Project Manager
    - Εκτέλεση της εντολής java manager.Menu. Από την κλάση αυτή θα οδηγηθεί κατάλληλα στις επόμενες κλάσεις του υποσυστήματος αυτού.

Τέλος, ο τρόπος με τον οποίο χρησιμοποιείται η Βάση δεδομένων είναι ο εξής: Δημιουργούμε μια Βάση Δεδομένων με το όνομα “NaviMoz” (Το όνομα αυτό μπορεί να αλλάξει, στην περίπτωση αυτή όμως πρέπει να αλλαχθεί και στην κλάση όπου υλοποιείται η σύνδεση με τη Βάση Δεδομένων. Στη συνέχεια πρέπει να ξανά-μεταγλωττίσουμε την κλάση). Οι πίνακες της Βάσης αυτής δημιουργούνται μέσα από τον κώδικα και είναι οι :

1) users: Έχει για πεδία του τον αύξοντα αριθμό χρήστη, το όνομα, το επώνυμο, το όνομα χρήστη (username), τον κωδικό χρήστη (password) και την ηλεκτρονική διεύθυνση.
2) session: Έχει για πεδία του τον αύξοντα αριθμό πλοήγησης, τον αριθμό χρήστη που επιτέλεσε την πλοήγηση αυτή, την πλοήγηση και τις χρονικές στιγμές έναρξης και λήξης αυτής.

6

Έλεγχος

Στο κεφάλαιο αυτό παρουσιάζεται ο έλεγχος του συστήματος μέσω ενός λεπτομερούς σεναρίου εκτέλεσης.

6.1 Μεθοδολογία Ελέγχου

Ο έλεγχος του συστήματος πραγματοποιείται με χρήση σεναρίων λειτουργίας τα οποία χρησιμοποιούν όλες τις λειτουργίες του συστήματος. Έχουμε τη δυνατότητα να ανατρέχουμε στη βάση και να ελέγχουμε έτσι αν τα αποτελέσματα που παίρνουμε είναι σωστά. Με σύγκριση των αναμενόμενων με τα λαμβανόμενα αποτελέσματα αξιολογείται η συμπεριφορά και η απόδοση του συστήματος. Μόνη εξαίρεση αποτελούν οι εργασίες συσταδοποίησης, στις οποίες εκ των πραγμάτων τα όρια σωστού και λάθους είναι λίγο ασαφή. Διαισθητικά καταλαβαίνουμε αν εκτελείται σωστή ομαδοποίηση.

6.2 Αναλυτική Παρουσίαση Ελέγχου

6.2.1: Διαπροσωπεία Χρήστη

Στην ενότητα αυτή παρουσιάζεται ένα σενάριο χρήσης του συστήματος από το χρήστη.

6.2.1.1 Εισαγωγή στοιχείων υπάρχοντος χρήστη

Όταν εισάγεται στο σύστημα του ζητείται να συμπληρώσει την παρακάτω φόρμα (σχήμα 6.1)

Σχήμα 6.1: Φόρμα εισαγωγής στοιχείων υπάρχοντος χρήστη

Σε περίπτωση που είναι χρήστης του συστήματος εισάγει στα αντίστοιχα πεδία το Username και το Password του. Το σύστημα πραγματοποιεί έλεγχο αν είναι εγγεγραμμένος χρήστης εξετάζοντας τη Βάση Δεδομένων. Αν τα στοιχεία αυτά δεν αντιστοιχούν σε χρήστη του συστήματος προβάλλεται το παρακάτω μήνυμα:

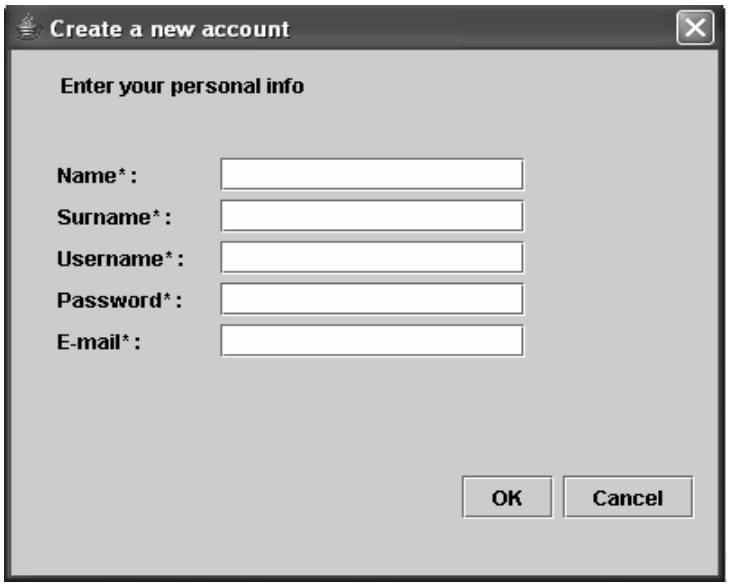
Σχήμα 6.2: Προτρεπτικό μήνυμα

Αν τα στοιχεία είναι σωστά εισάγεται στο σύστημα NaviMoz και εμφανίζεται η κεντρική σελίδα του dmoz, η οποία φαίνεται στο σχήμα 6.3. Από το σημείο αυτό μπορεί να αρχίσει να πραγματοποιεί την πλοήγησή του η οποία όταν κλείσει την εφαρμογή θα καταχωρηθεί στη Βάση Δεδομένων.

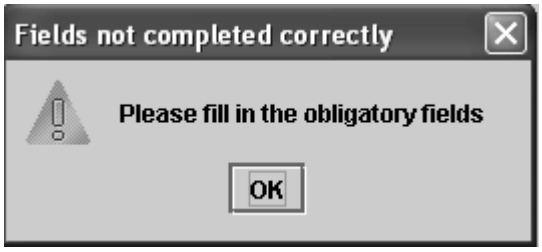
Σχήμα 6.3: Η αρχική σελίδα της ιεραρχίας του dmoz

6.1.2.2 Εγγραφή νέου χρήστη στο σύστημα

Αν δεν είναι χρήστης του συστήματος και πατήσει πάνω στο κουμπί “Register Now”, θα του παρουσιαστεί η φόρμα του σχήματος 6.4. Εισάγει τότε τα στοιχεία που του ζητούνται, τα οποία είναι όλα υποχρεωτικά πεδία. Αν πατήσει το πλήκτρο OK χωρίς να έχει συμπληρώσει κάποιο πεδίο εμφανίζεται το μήνυμα του σχήματος 6.5. Επίσης, πραγματοποιείται έλεγχος αν τα username και password χρησιμοποιούνται ήδη από κάποιον άλλο χρήστη και αν ναι προβάλλονται αντίστοιχα τα μηνύματα των σχημάτων 6.6 και 6.7.



Σχήμα 6.4: Φόρμα για εγγραφή νέου χρήστη στο σύστημα



Σχήμα 6.5: Προειδοποιητικό μήνυμα

Σχήμα 6.6: Προτρεπτικό μήνυμα

Σχήμα 6.7: Προτρεπτικό μήνυμα

Αν οι έλεγχοι διεξαχθούν σωστά εγγράφεται στο σύστημα και εισάγεται για πρώτη φορά στην ιεραρχία του dmoz αφού πρώτα προβληθεί το παρακάτω μήνυμα:

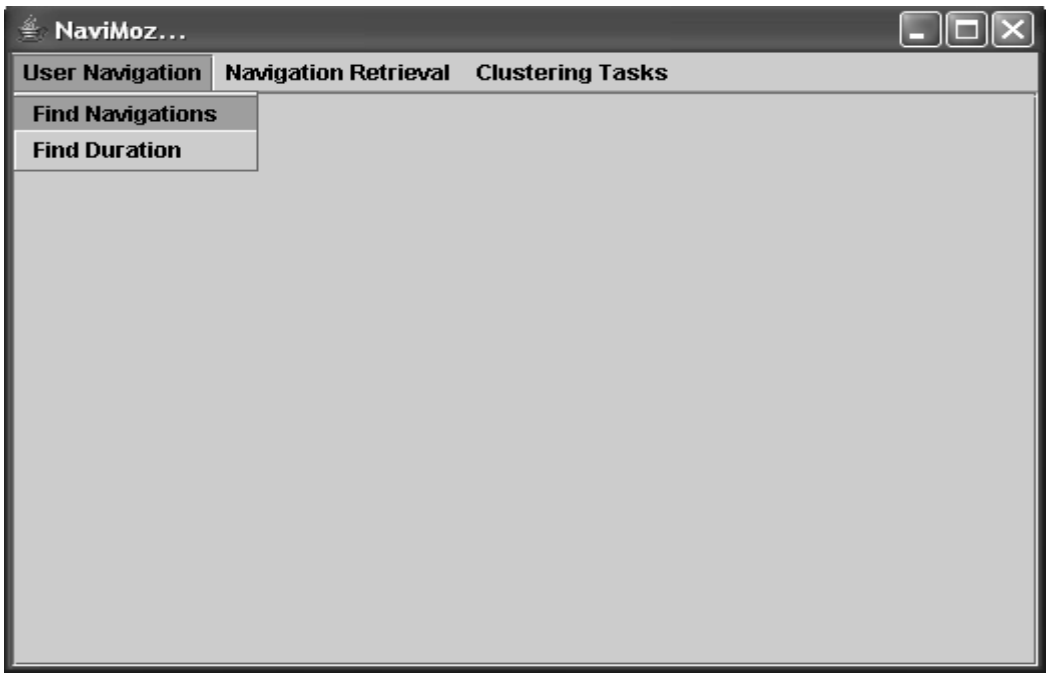
Σχήμα 6.8: Μήνυμα καλωσορίσματος του νέου χρήστη

6.2.2: Διαπροσωπεία διαχειριστή

Στην ενότητα αυτή παρουσιάζεται ένα σενάριο χρήσης του συστήματος από το διαχειριστή. Σημειώνεται ότι με το συγκεκριμένο παράδειγμα δε φαίνονται όλες οι εφαρμογές του συστήματος, αλλά οι πιο σημαντικές.

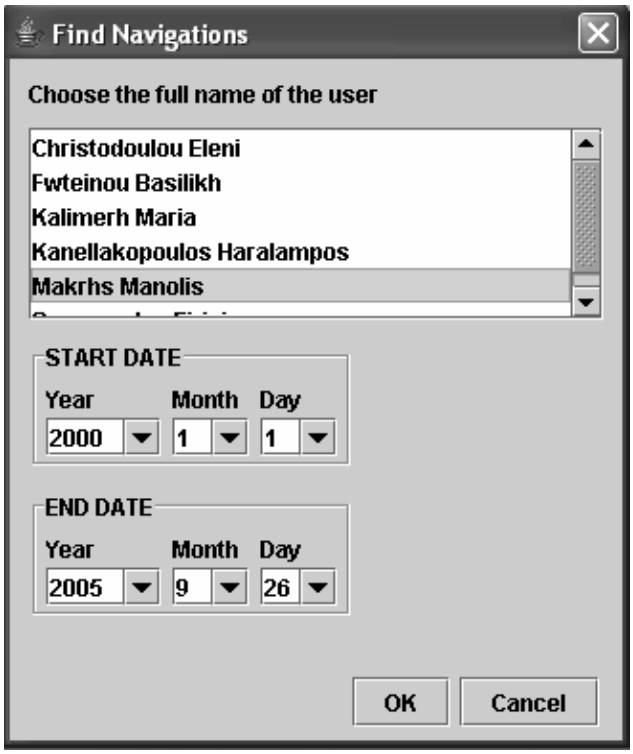
6.2.2.1 Εύρεση των πλοηγήσεων

Με την εκκίνηση του συστήματος η πρώτη φόρμα που παρουσιάζεται στο διαχειριστή είναι η κεντρική του συστήματος και αποτελείται από ένα μενού επιλογών στο οποίο είναι ομαδοποιημένες οι εργασίες του συστήματος με βάση την κατηγοριοποίηση που παρουσιάστηκε στο κεφάλαιο 2. Έστω ότι από αυτή επιλέγει από το μενού “User Navigation” την εργασία “Find Navigations”, επιλέγει δηλαδή να δει τις πλοηγήσεις κάποιου χρήστη μέσα σε συγκεκριμένο χρονικό διάστημα.(σχήμα 6.9)

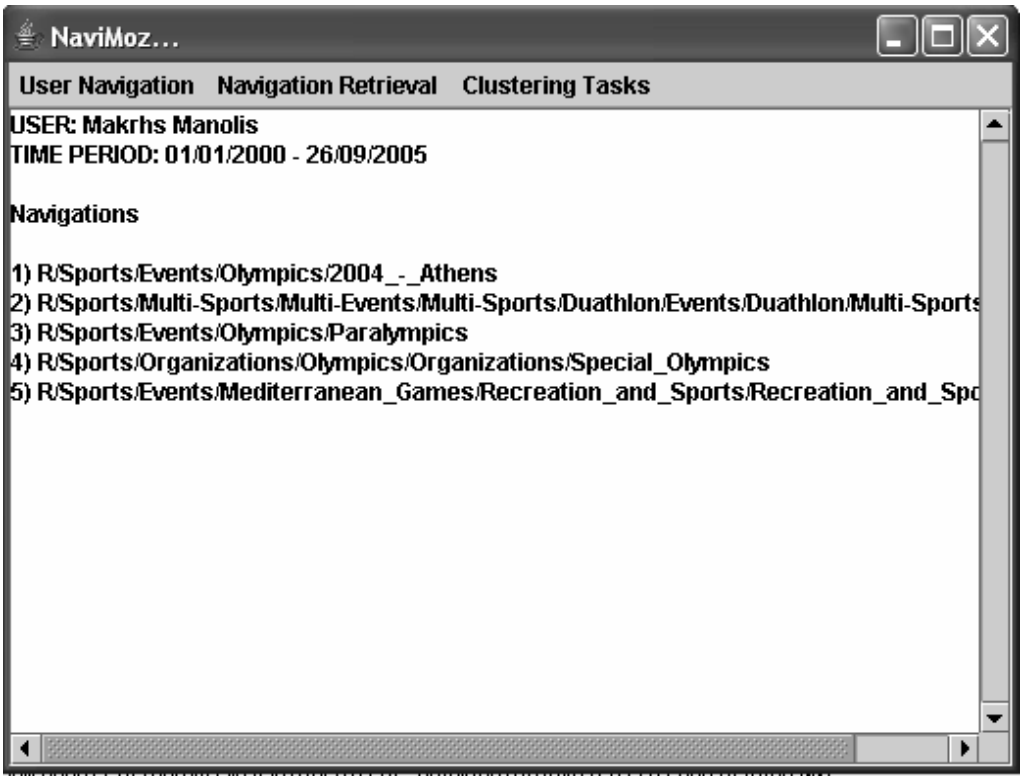


Σχήμα 6.9: Κεντρική φόρμα της διαπροσωπείας του διαχειριστή

Η φόρμα που εμφανίζεται στη συνέχεια είναι αυτή που φαίνεται στο σχήμα 6.10. Από αυτή τη φόρμα ο διαχειριστής επιλέγει έναν από τους χρήστες του συστήματος από τη λίστα, η οποία διαθέτει και γραμμή κύλισης. Έστω ότι ο διαχειριστής επιλέγει το χρήστη «Μακρή Μανόλη». Επίσης επιλέγει και τις ημερομηνίες εντός των οποίων θα ήθελε να εξετάσει τις πλοηγήσεις. Αν δεν επιλέξει ημερομηνίες θα εξεταστούν οι πλοηγήσεις που βρίσκονται εντός των προεπιλεγμένων ημερομηνιών. Σημειώνεται ότι η ημερομηνία έναρξης είναι πάντα η 1/1/2000 και η ημερομηνία λήξης είναι η τρέχουσα ημερομηνία. Το αποτέλεσμα της επιλογής του διαχειριστή, έστω για τις προεπιλεγμένες ημερομηνίες είναι αυτό που φαίνεται στο σχήμα 6.11. Αν ανατρέξουμε στη Βάση Δεδομένων, θα διαπιστώσουμε ότι το αποτέλεσμα αυτό είναι σωστό, και ότι όντως αυτές είναι η πλοηγήσεις του χρήστη Μακρή Μανόλη στο δοσμένο χρονικό διάστημα. Αν ο διαχειριστής πατήσει το πλήκτρο OK χωρίς να έχει επιλέξει χρήστη παρουσιάζεται το προειδοποιητικό μήνυμα του σχήματος 6.12.



Σχήμα 6.10: Επιλογή παραμέτρων για την εργασία "Find Navigations"



Σχήμα 6.11: Αποτελέσματα της εργασίας "Find Navigations"

Σχήμα 6.12: Προειδοποιητικό μήνυμα

Σημειώνεται ότι η φόρμα επιστροφής των αποτελεσμάτων είναι η αρχική φόρμα του συστήματος, μπορεί δηλαδή ο διαχειριστής, μετά την επιστροφή των αποτελεσμάτων, να επιλέξει μια άλλη εφαρμογή του συστήματος. Αυτό ισχύει για όλες τις εργασίες του συστήματος.

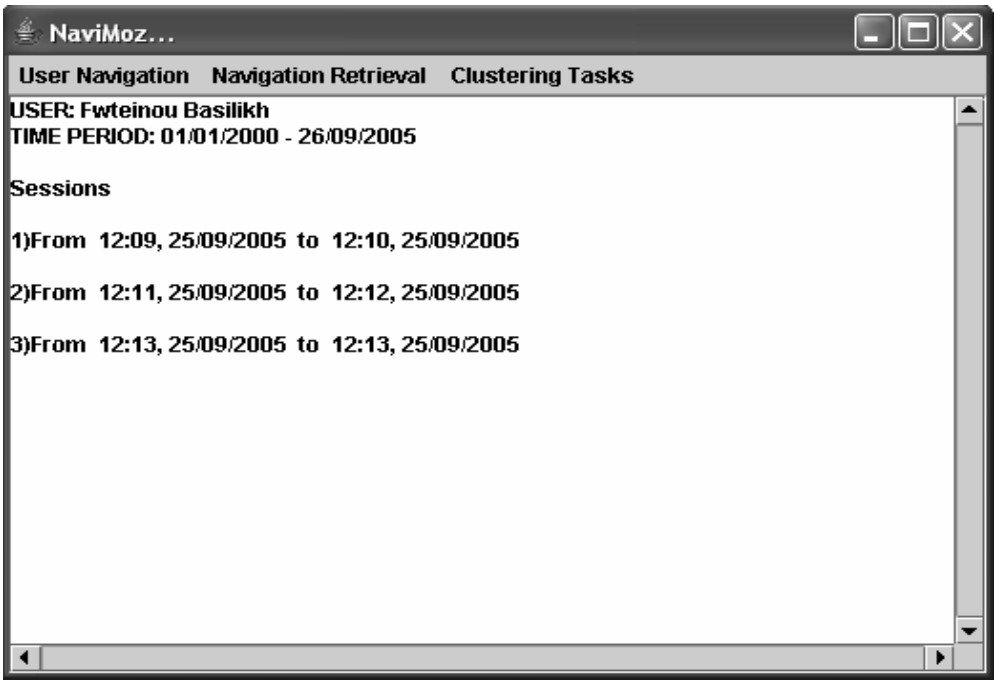
6.2.2.2 Εύρεση των χρονικών στιγμών των πλοηγήσεων

Έστω ότι στη συνέχεια ο διαχειριστής επιλέγει πάλι από την πρώτη κατηγορία εφαρμογών τη "Find Duration", δηλαδή την εύρεση της χρονικής διάρκειας των πλοηγήσεων του επιλεγμένου χρήστη. Η φόρμα επιλογής χρήστη είναι η ίδια με του σχήματος 6.10.

Σχήμα 6.13: Επιλογή της εργασίας "Find Duration"

Το μήνυμα 6.12 προβάλλεται και πάλι στην περίπτωση που δεν έχει επιλεχθεί χρήστης

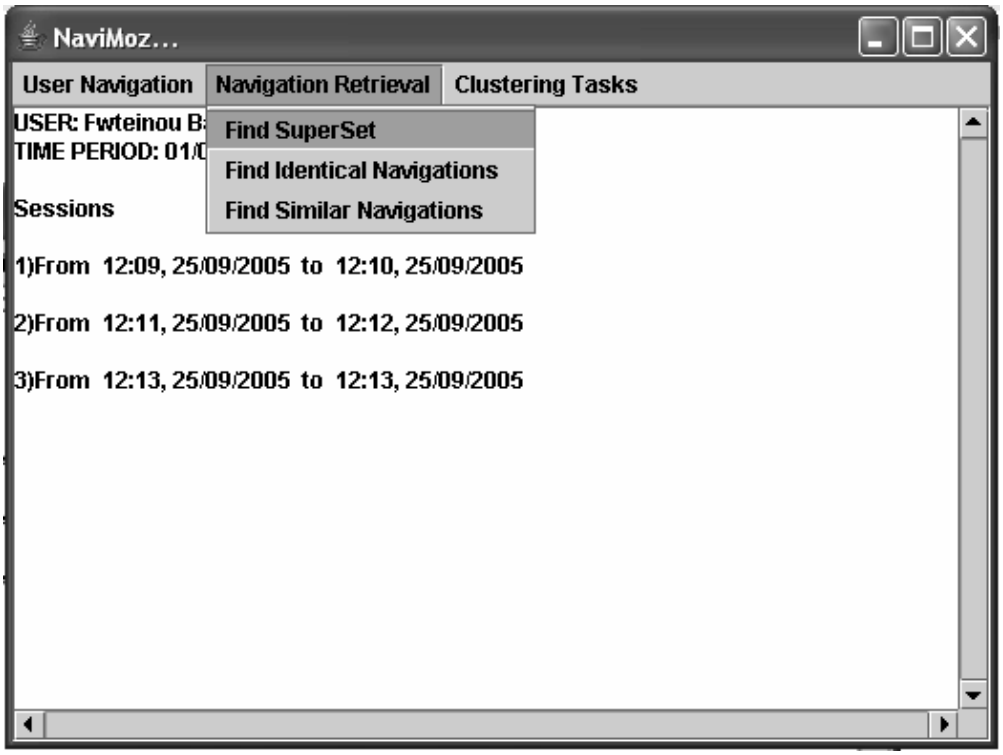
Αν ο διαχειριστής επιλέξει, για παράδειγμα, να δει τις πλοηγήσεις της Φωτεινού Βασιλικής, τα αποτελέσματα που θα λάβει είναι αυτά του σχήματος 6.14. Αν ανατρέξουμε στη Βάση Δεδομένων θα διαπιστώσουμε ότι το αποτέλεσμα είναι σωστό.



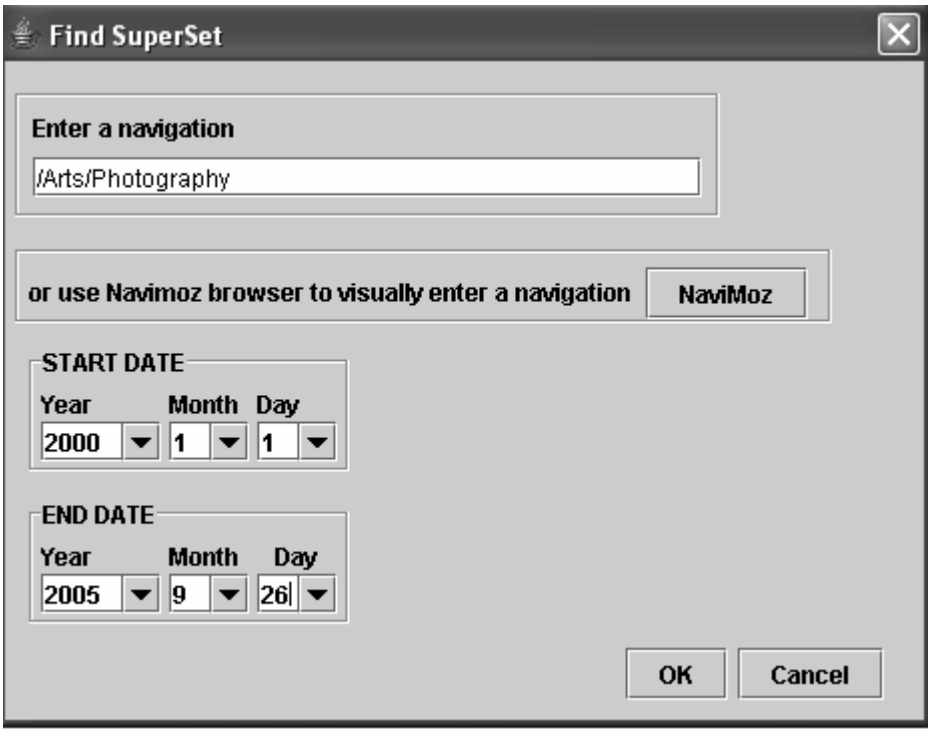
Σχήμα 6.14: Αποτελέσματα της εργασίας "Find Duration"

6.2.2.3 Εύρεση των πλοηγήσεων που είναι υπερσύνολο μιας δοσμένης

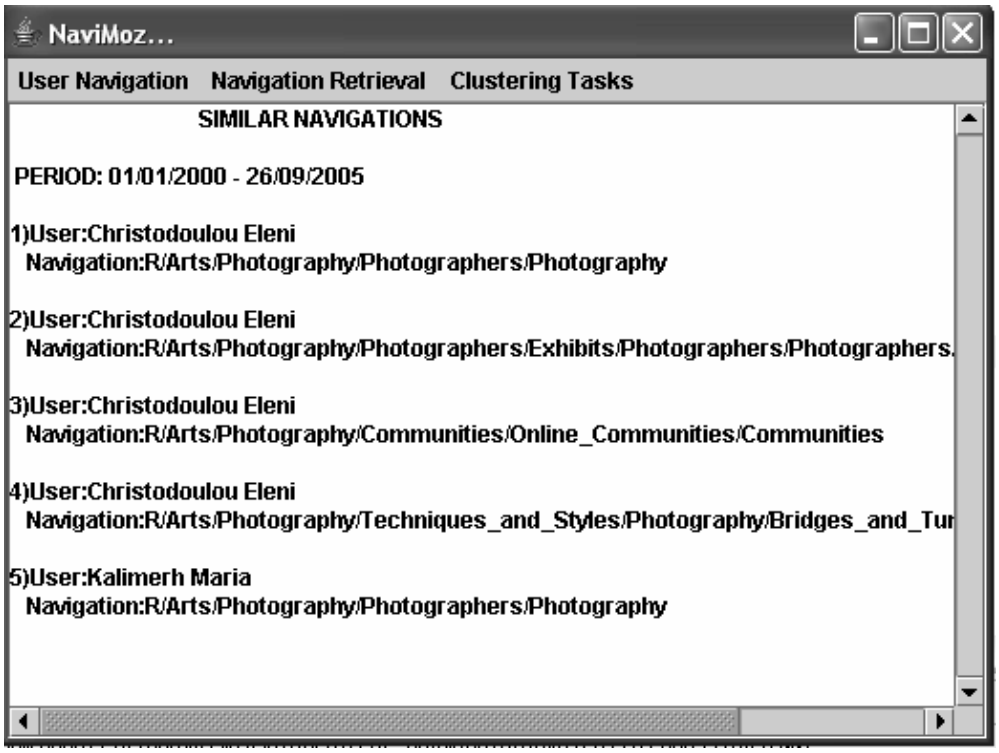
Έστω τώρα ότι ο διαχειριστής επιθυμεί να πραγματοποιηθεί μία από τις εργασίες εξόρυξης δεδομένων δοθείσης μιας πλοήγησης-προτύπου. Οι εργασίες αυτές ανήκουν στη δεύτερη κατηγορία εργασιών και είναι ομαδοποιημένες κάτω από το μενού "Navigation Retrieval". Έστω ότι επιλέγει την εύρεση των πλοηγήσεων εκείνων που αποτελούν υπερσύνολο της δοσμένης, επιλέγει δηλαδή "Find SuperSet". Η φόρμα που εμφανίζεται τότε είναι αυτή του σχήματος 6.16.Ο διαχειριστής μπορεί να εισάγει την πλοήγηση-πρότυπο είτε πληκτρολογώντας την είτε πατώντας το κουμπί NaviMoz. Στη δεύτερη περίπτωση ανοίγει ένας browser με την ιεραρχία του dmoz όπου ο διαχειριστής μπορεί να αρχίσει να πλοηγείται. Η πλοήγησή του αποτελεί την πλοήγηση-πρότυπο. Πρέπει να σημειωθεί ότι σε όλες τις εργασίες της δεύτερης αυτής κατηγορίας, η αρχική σελίδα www.dmoz.org εννοείται και εισάγεται από το πρόγραμμα ως R (ρίζα). Έστω τώρα ότι επιλέγει να δει τις πλοηγήσεις που περιέχουν την υπό-πλοήγηση /Arts/Photography, την οποία πληκτρολογεί. Οι ημερομηνίες είναι οι προεπιλεγμένες. Τα αποτελέσματα δίνονται στο σχήμα 6.17 και αν ανατρέξουμε στη Βάση Δεδομένων παρατηρούμε ότι είναι σωστά. Στα αποτελέσματα δίνονται οι χρήστες που έχουν πραγματοποιήσει την πλοήγηση-υπερσύνολο καθώς και η πλοήγηση αυτή. Αν δεν έχει εισάγει κάποια πλοήγηση στο προβλεπόμενο πεδίο και πατήσει το κουμπί OK, εμφανίζεται το προειδοποιητικό μήνυμα του σχήματος 6.18.



Σχήμα 6.15: Επιλογή της εργασίας "Find SuperSet"



Σχήμα 6.16: Επιλογή των παραμέτρων της εργασίας "Find SuperSet"

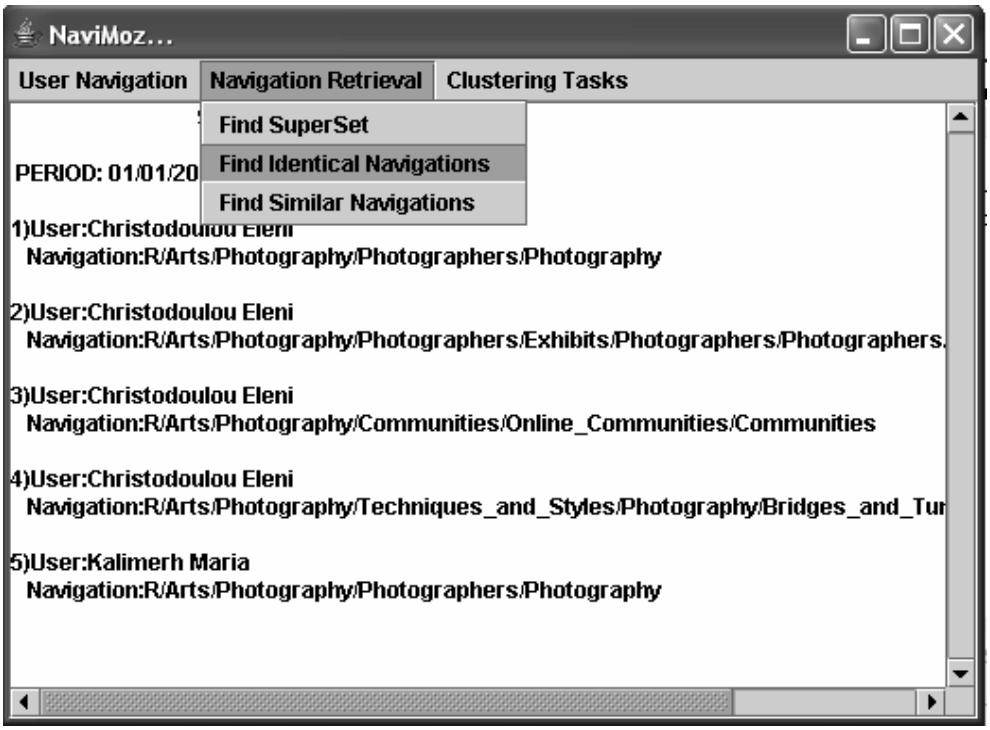


Σχήμα 6.17: Αποτελέσματα της εργασίας "Find SuperSet"

Σχήμα 6.18: Προειδοποιητικό μήνυμα

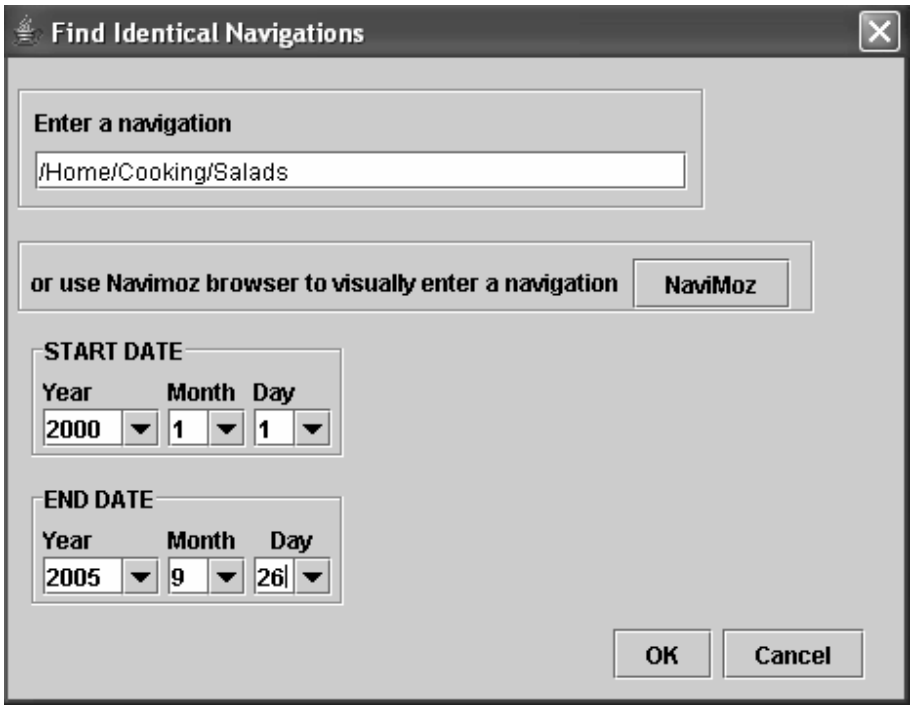
6.2.2.4 Εύρεση των πλοηγήσεων που είναι ίδιες με μια δοσμένη

Έστω τώρα ότι στη συνέχεια ο διαχειριστής επιλέγει την εργασία "Find Identical Navigations" που υπάγεται στη ίδια ομάδα εργασιών. Επιλέγει δηλαδή να δει τις πλοηγήσεις που είναι ακριβώς ίδιες με την πλοήγηση-πρότυπο. Η επιλογή του αυτή φαίνεται στο σχήμα 6.19. Η φόρμα που του παρουσιάζεται είναι η ίδια με της προηγούμενης εργασίας. Μπορεί κι εδώ να εισάγει την πλοήγηση με έναν από τους δύο τρόπους που προαναφέρθηκαν και στην περίπτωση που πατήσει το OK χωρίς να έχει εισάγει κάποια πλοήγηση παρουσιάζεται και εδώ το μήνυμα του σχήματος 6.18. Έστω ότι εισάγει την πλοήγηση Home/Cooking/Salads μέσω του browser, ο οποίος ανοίγει με πάτημα του κουμπιού NaviMoz, για τις προεπιλεγμένες ημερομηνίες(σχήμα 6.20).



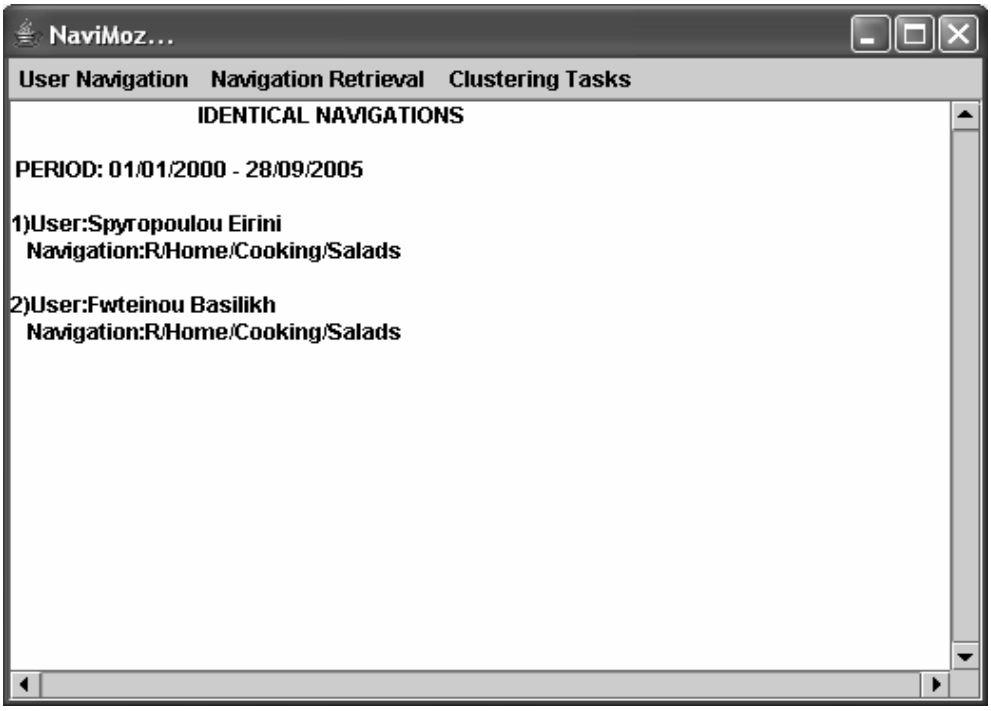
Σχήμα 6.19: Επιλογή της εργασίας "Find Identical Navigations"

.

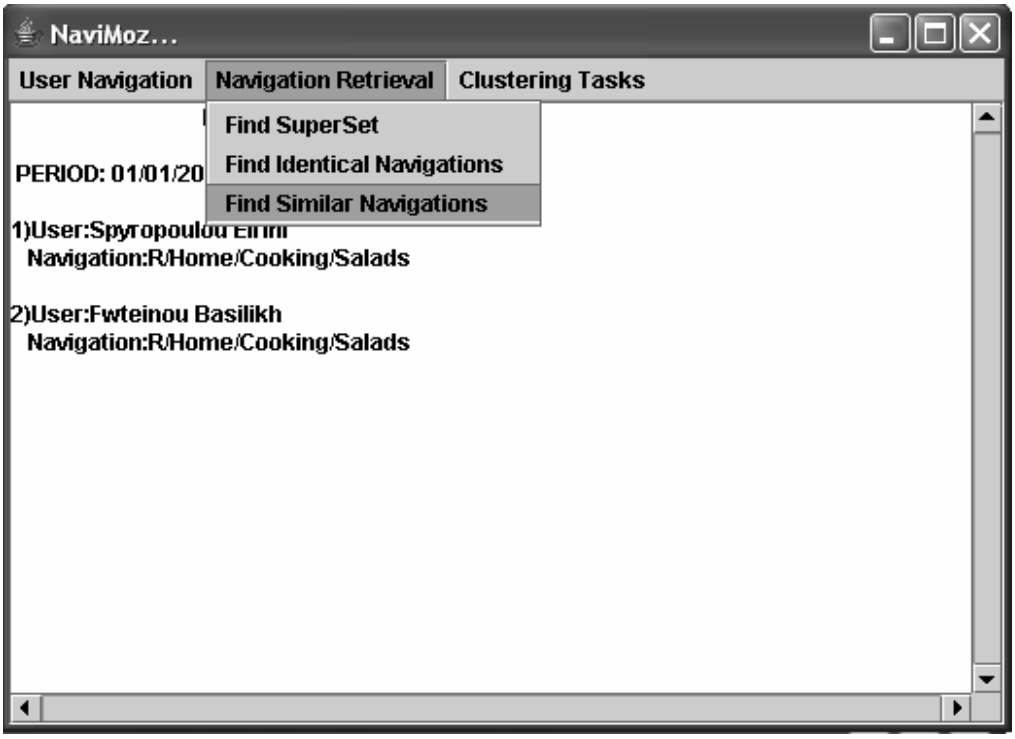


Σχήμα 6.20: Επιλογή παραμέτρων της εργασίας "Find Identical Navigations"

Οι ίδιες πλοηγήσεις επιστρέφονται στη φόρμα των αποτελεσμάτων και φαίνονται στο σχήμα 6.21. Φαίνονται οι χρήστες που έχουν πραγματοποιήσει τις ίδιες πλοηγήσεις καθώς και οι πλοηγήσεις αυτές. Ανατρέχοντας στη Βάση Δεδομένων, διαπιστώνουμε ότι το αποτέλεσμα είναι σωστό.



Σχήμα 6.21: Αποτελέσματα της εργασίας "Find Identical Navigations"

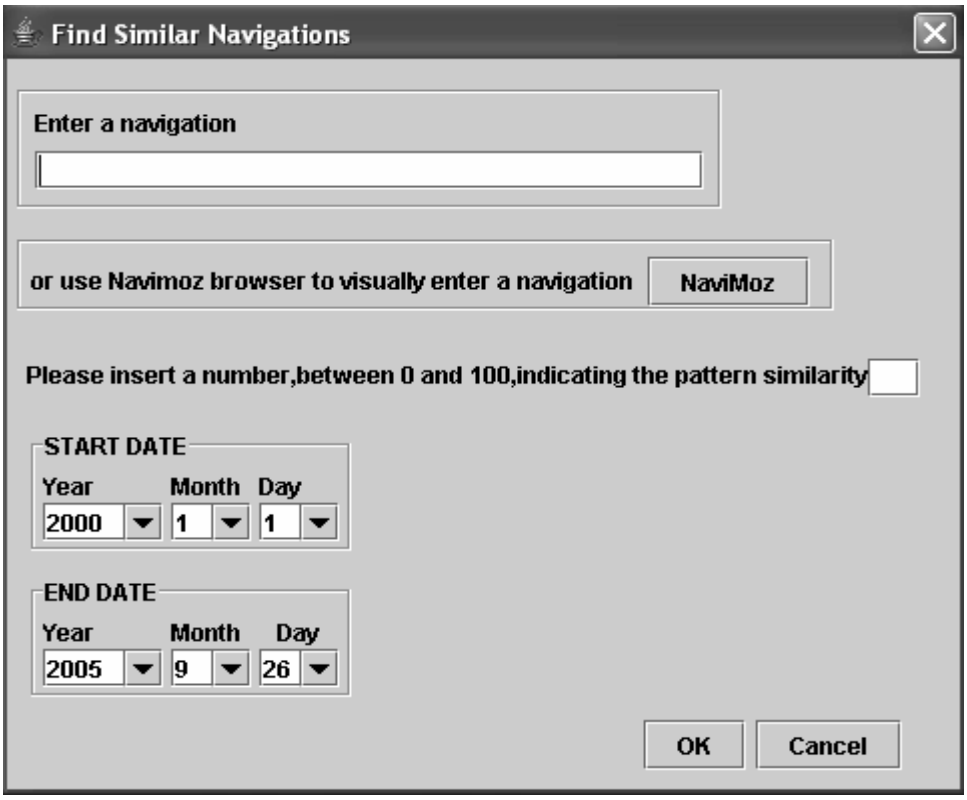


Σχήμα 6.22: Επιλογή της εργασίας "Find Similar Navigations"

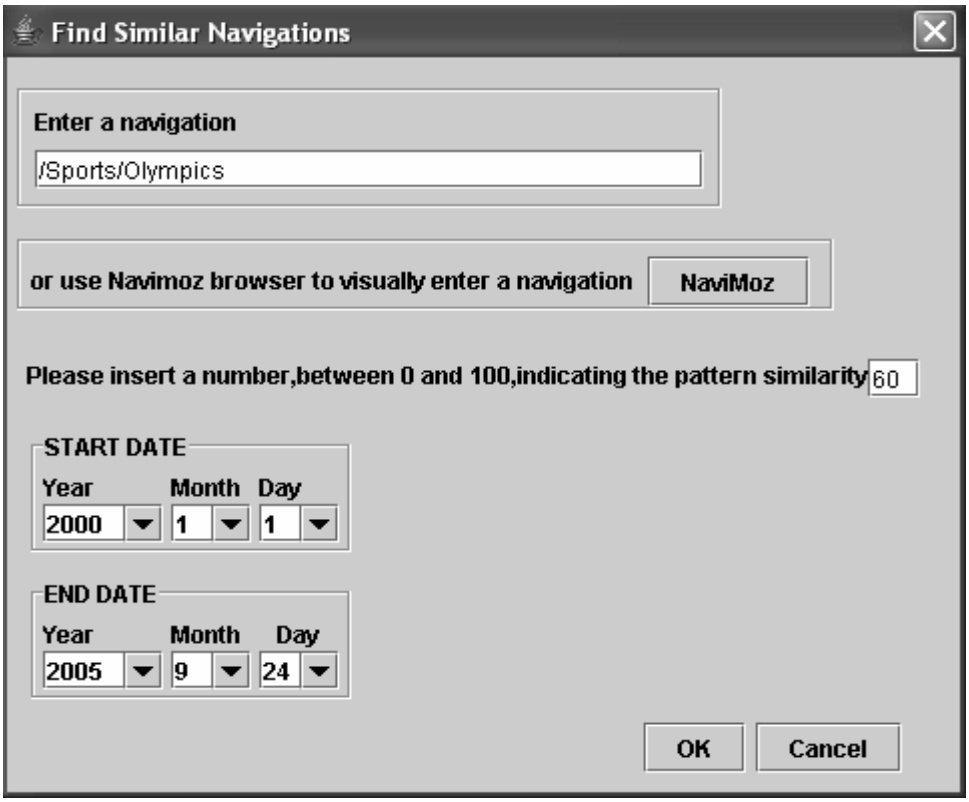
6.2.2.5 Εύρεση των πλοηγήσεων που είναι όμοιες με μια δοσμένη

Έστω ότι στη συνέχεια ο διαχειριστής θέλει να εισάγει μια πλοήγηση και να βρει τις πλοηγήσεις εκείνες που είναι τουλάχιστον κατά ένα ποσοστό όμοιες με τη δοσμένη. Επιλέγει από τη δεύτερη κατηγορία εργασιών την εργασία "Find Similar Navigations" (σχήμα 6.22)

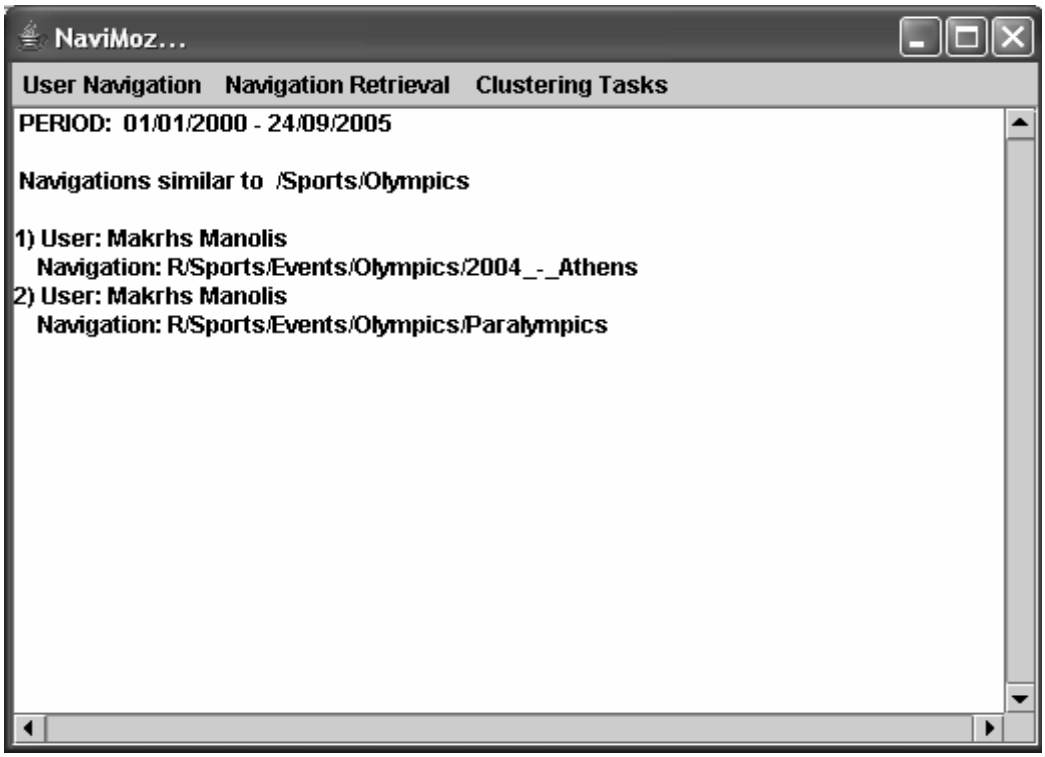
Η φόρμα που εμφανίζεται τότε είναι αυτή του Σχήματος 6.23. Και στην εργασία αυτή παρουσιάζονται οι δύο προαναφερθείσες εναλλακτικές εισαγωγής της πλοήγησης-πρότυπο. Επίσης ο χρήστης εισάγει το ποσοστό % ομοιότητας των πλοηγήσεων με τη δοσμένη. Επιστρέφονται οι πλοηγήσεις που έχουν τουλάχιστον ίσο βαθμό ομοιότητας με το ποσοστό αυτό. Οι ομοιότητες μεταξύ των πλοηγήσεων υπολογίζονται με κατάλληλες τροποποιήσεις της μεταξύ τους δομικής απόστασης. Έστω ότι ο διαχειριστής συμπληρώνει τη φόρμα με τα στοιχεία που φαίνονται στο σχήμα 6.24 (στην επιλογή του αυτή υπάρχουν διαφορετικές ημερομηνίες από τις προεπιλεγμένες). Το σύστημα επιστρέφει τα αποτελέσματα στην κατάλληλη φόρμα, όπως φαίνεται στο σχήμα 6.25. Επιστρέφονται οι πλοηγήσεις που έχουν πραγματοποιηθεί εντός του διαστήματος 1/1/2005-24/9/2005 και είναι κατά τουλάχιστον 60% όμοιες με τη δοσμένη Sports/Olympics, καθώς και οι χρήστες που τις έχουν πραγματοποιήσει. Σημειώνεται ότι όσο μεγαλύτερο είναι το ποσοστό που εισάγει ο διαχειριστής, τόσο περισσότερο όμοιες είναι οι πλοηγήσεις.



Σχήμα 6.23: Η φόρμα της εργασίας "Find Similar Navigations"

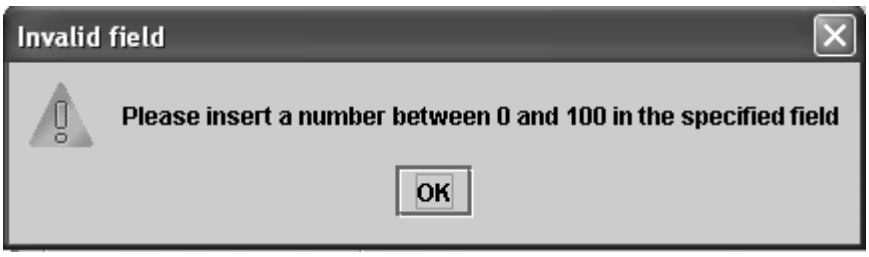


Σχήμα 6.24: Επιλογή παραμέτρων της εργασίας "Find Similar Navigations"



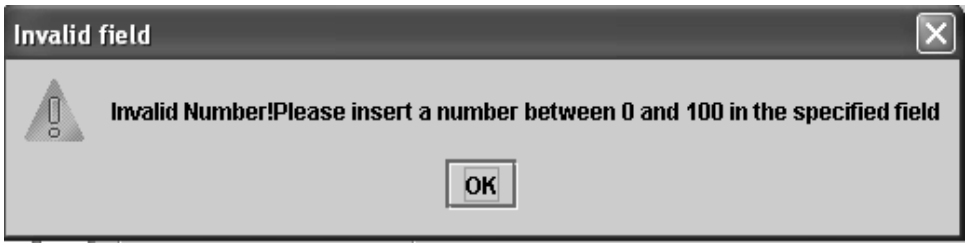
Σχήμα 6.25: Τα αποτελέσματα της εργασίας "Find Similar Navigations"

Αν ο διαχειριστής πατήσει το πλήκτρο OK χωρίς να έχει εισάγει κάποια πλοήγηση εμφανίζεται το προειδοποιητικό μήνυμα του σχήματος 6.18. Επιπλέον, αν δεν έχει εισάγει κάποιο ποσοστό στο προβλεπόμενο πεδίο ή αν έχει εισάγει κάτι άλλο εκτός από αριθμό, παρουσιάζεται το μήνυμα του σχήματος 6.26.



Σχήμα 6.26: Προειδοποιητικό μήνυμα

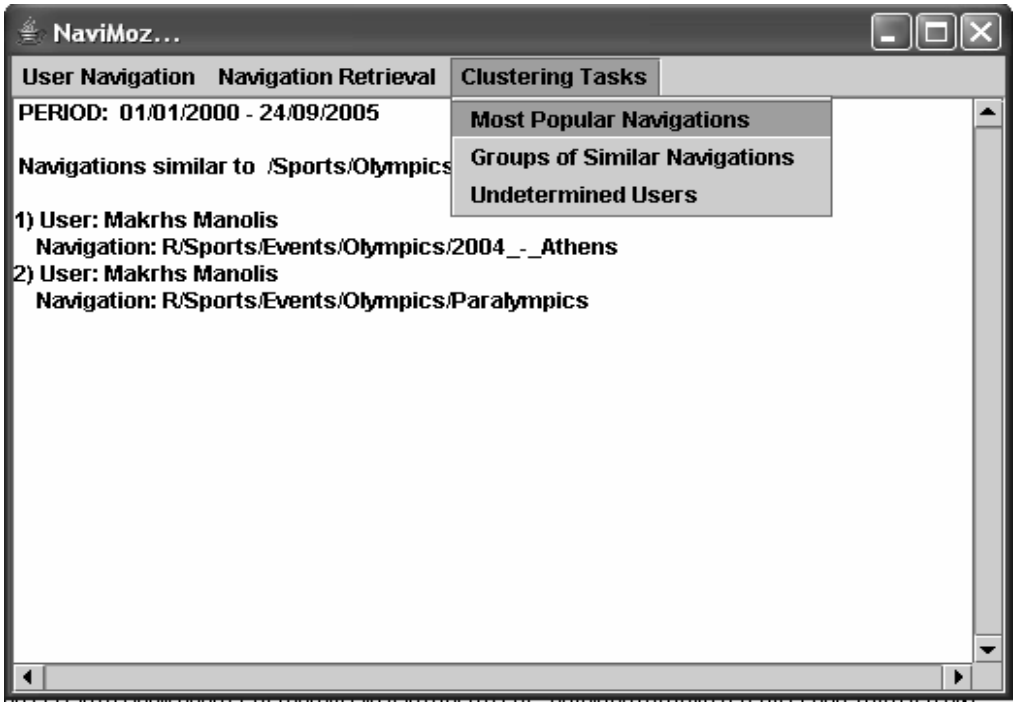
Τέλος, αν έχει εισάγει έναν αριθμό, ο οποίος όμως είναι μεγαλύτερος του 100 ή μικρότερος του 0, παρουσιάζεται το μήνυμα του σχήματος 6.27, το οποίο τον προειδοποιεί για εισαγωγή λάθος αριθμού.



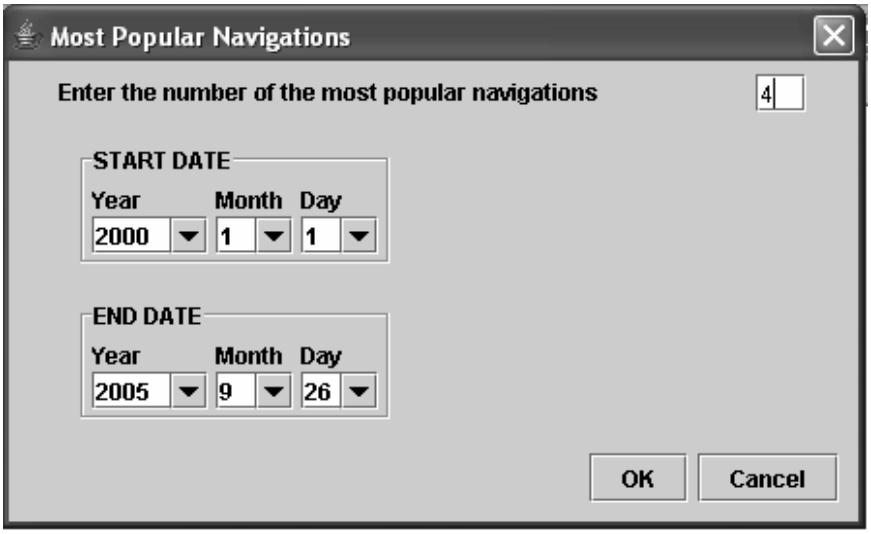
Σχήμα 6.27: Προειδοποιητικό μήνυμα

6.2.2.6 Εύρεση των πιο δημοφιλών πλοηγήσεων

Στη συνέχεια, έστω ότι ο διαχειριστής προχωρά στην τρίτη κατηγορία εργασιών του συστήματος, δηλαδή στις ερωτήσεις εξόρυξης γνώσης και ταυτοποίησης χρηστών. Στις εργασίες αυτές μπορεί να έχει πρόσβαση μέσω του τρίτου μενού της αρχικής φόρμας, το "Clustering Tasks". Για να πραγματοποιήσει την πρώτη εργασία της κατηγορίας αυτής, που είναι η εύρεση των n πιο δημοφιλών πλοηγήσεων εντός του δοσμένου χρονικού διαστήματος, όπου n είναι ένας αριθμός που δίνεται από το διαχειριστή, πρέπει να επιλέξει από το μενού "Clustering Tasks" την εργασία "Most Popular Navigations". Η επιλογή του αυτή δίνεται στο σχήμα 6.28. Η φόρμα που παρουσιάζεται τότε στο διαχειριστή είναι αυτή του σχήματος 6.29. Στο σχήμα 6.29 φαίνεται η φόρμα συμπληρωμένη με τις επιλογές του διαχειριστή. Έστω λοιπόν ότι επιλέγει να δει τις 4 πιο δημοφιλείς πλοηγήσεις που έχουν πραγματοποιηθεί στο διάστημα 1/1/2000-26/9/2005. Με το πάτημα του πλήκτρου OK επιστρέφονται τα αποτελέσματα στη φόρμα των αποτελεσμάτων (σχήμα 6.30)

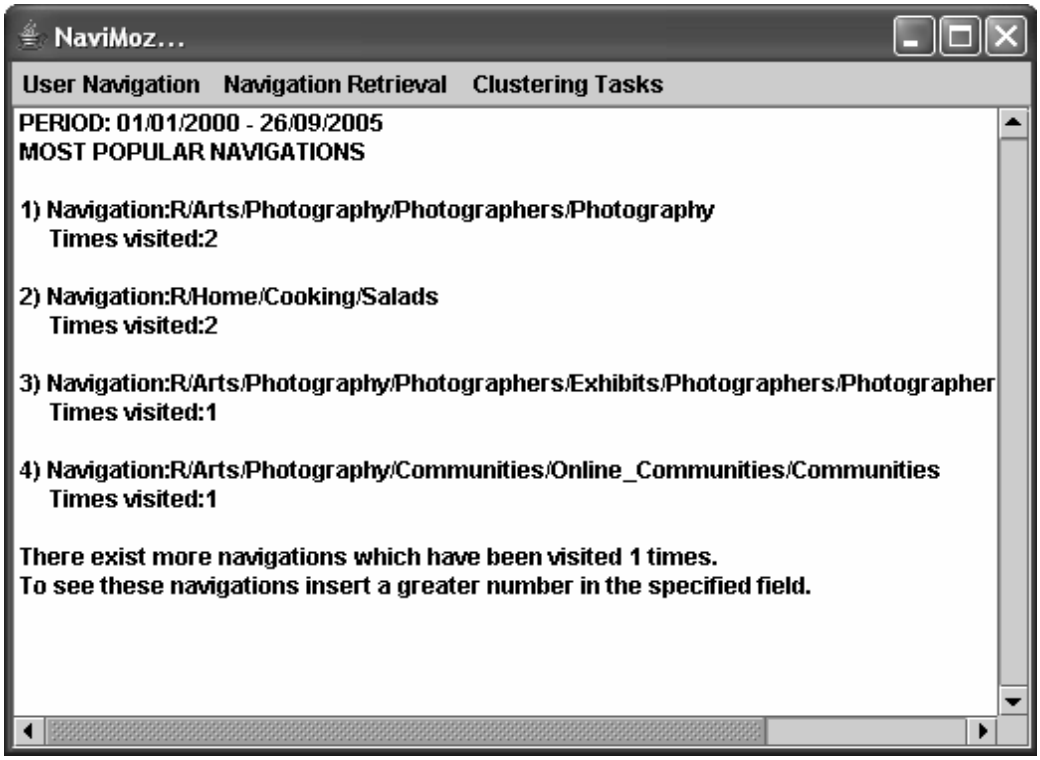


Σχήμα 6.28: Επιλογή της εργασίας "Most Popular Navigations"



Σχήμα 6.29: Επιλογή των παραμέτρων της εργασίας "Most Popular Navigations"

Στα αποτελέσματα παρουσιάζονται οι 4 πιο δημοφιλείς πλοηγήσεις για την επιλεγμένη χρονική περίοδο, καθώς και οι φορές που έχουν πραγματοποιηθεί (Times visited). Στην περίπτωση που υπάρχουν κι άλλες πλοηγήσεις οι οποίες έχουν πραγματοποιηθεί τόσες φορές όσες και η τελευταία που εμφανίζεται στη φόρμα των αποτελεσμάτων, οι οποίες όμως δεν επιστρέφονται από το σύστημα, γιατί στην περίπτωση αυτή θα ήταν πάνω από τον αριθμό που επέλεξε ο διαχειριστής, εμφανίζεται κατάλληλο μήνυμα .



Σχήμα 6.30: Αποτελέσματα της εργασίας "Most Popular Navigations"

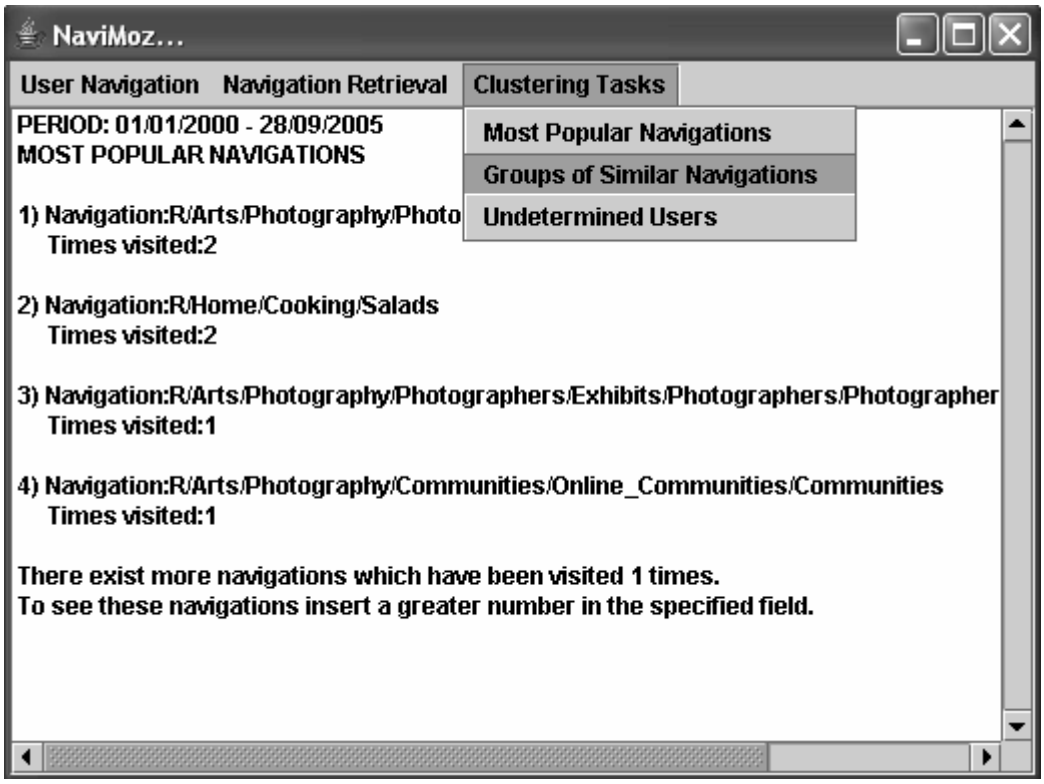
Έτσι λοιπόν, στην παραπάνω περίπτωση εκτέλεσης του προγράμματος, υπάρχουν κι άλλες πλοηγήσεις που έχουν πραγματοποιηθεί μια φορά. Αυτές όμως δεν επιστρέφονται από το σύστημα γιατί ο διαχειριστής έχει επιλέξει να επιστραφούν οι 4 πρώτες εργασίες. Αντί αυτών, προβάλλεται κατάλληλο μήνυμα στο κάτω μέρος της φόρμας. Αν ανατρέξουμε στη Βάση Δεδομένων θα διαπιστώσουμε ότι το αποτέλεσμα του σχήματος 6.28 είναι σωστό. Στην περίπτωση που ο διαχειριστής πατήσει το πλήκτρο OK χωρίς να έχει συμπληρώσει το πλήθος των δημοφιλέστερων πλοηγήσεων στο αντίστοιχο πεδίο, επειδή είναι προεπιλεγμένη η τιμή 1 παρουσιάζεται η πιο δημοφιλής πλοήγηση απ' όλες. Αν όμως εισάγει κάτι άλλο εκτός από αριθμό (για παράδειγμα έναν χαρακτήρα) εμφανίζεται το μήνυμα του σχήματος 6.31. Επίσης, αν εισάγει έναν αριθμό ο οποίος είναι μεγαλύτερος από το πλήθος των πλοηγήσεων, αφού έχουν ομαδοποιηθεί ώστε οι ίδιες να είναι μαζί, εμφανίζεται το μήνυμα του σχήματος 6.32, το οποίο τον προτρέπει να εισάγει έναν αριθμό μικρότερο από έναν συγκεκριμένο, ο οποίος αντιστοιχεί στο πλήθος των ομαδοποιημένων πλοηγήσεων για το επιλεγμένο χρονικό διάστημα. Στο συγκεκριμένο παράδειγμα ο αριθμός αυτός είναι το 29.

Σχήμα 6.31: Προειδοποιητικό μήνυμα

Σχήμα 6.32: Προτρεπτικό μήνυμα

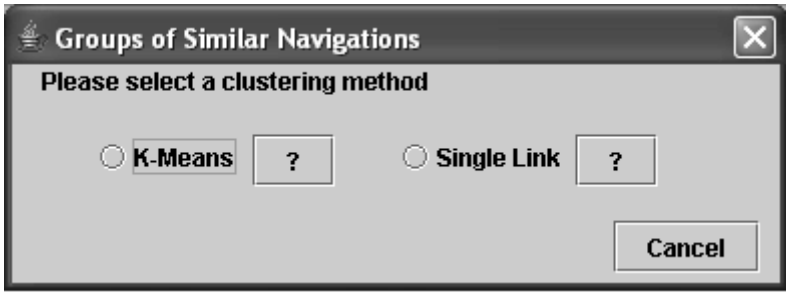
6.2.2.7 Εύρεση των συστάδων των πλοηγήσεων με τη μέθοδο K-Μέσων

Στη συνέχεια, ο διαχειριστής επιλέγει από το μενού "Clustering Tasks" τη δεύτερη εργασία, η οποία είναι η συσταδοποίηση. Επιλέγει λοιπόν την εργασία "Groups of Similar Navigations". Η επιλογή του αυτή φαίνεται στο σχήμα 6.33.



Σχήμα 6.33: Επιλογή της εργασίας "Groups of Similar Navigations"

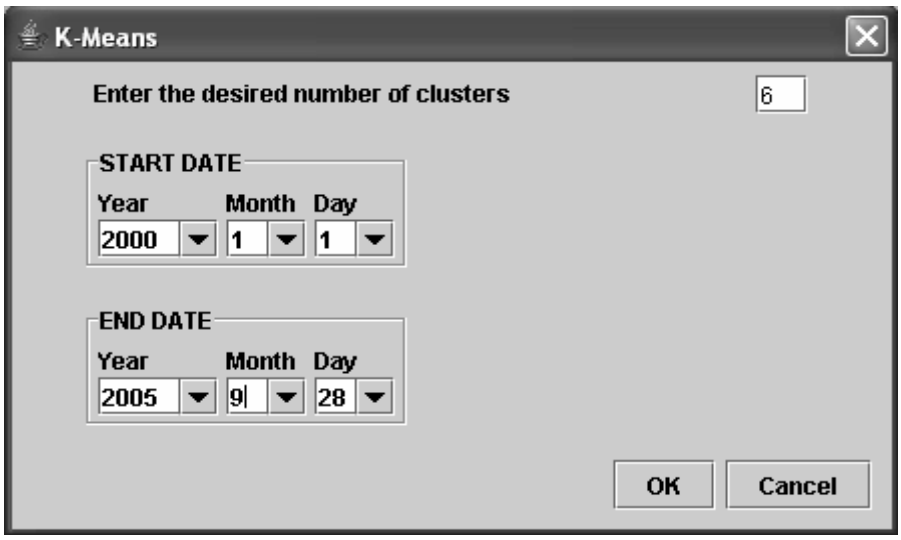
Η φόρμα που παρουσιάζεται τότε στο διαχειριστή είναι αυτή του σχήματος 6.34. Ο διαχειριστής μπορεί να επιλέξει μια από τις μεθόδους συσταδοποίησης μεταξύ του αλγορίθμου των Κ Μέσων (K Means) και της τεχνικής «μονός σύνδεσμος» (Single Link). Οι δύο αυτές επιλογές είναι αλληλοαποκλειόμενες. Πατώντας πάνω στα κουμπιά με το ερωτηματικό «?», ανοίγει ο Internet Explorer σε μία σελίδα που περιέχει πληροφορίες για τις τεχνικές Κ Μέσων και «μονός σύνδεσμος» αντίστοιχα.



Σχήμα 6.34: Φόρμα επιλογής της εργασίας συσταδοποίησης

Έστω ότι ο διαχειριστής επιλέγει να χρησιμοποιηθεί ο αλγόριθμος Κ Μέσων για τη συσταδοποίηση των πλοηγήσεων, πατώντας πάνω στο κουμπί K Means. Τότε παρουσιάζεται

η φόρμα επιλογών 6.35. Από το διαχειριστή ζητείται να συμπληρώσει τον επιθυμητό αριθμό συστάδων, γιατί ο αλγόριθμος τον απαιτεί σαν είσοδο για να μπορέσει να εκτελεστεί. Η προεπιλεγμένη τιμή είναι 1, οπότε αν ο διαχειριστής δε συμπληρώσει τίποτα και πατήσει το OK, τα αποτελέσματα θα ομαδοποιηθούν σε μια συστάδα. Για το παράδειγμά μας βρέθηκε ότι ένα «καλό» πλήθος συστάδων είναι 6.



Σχήμα 6.35: Επιλογή των παραμέτρων του αλγορίθμου Κ Μέσων

Οι συστάδες επιστρέφονται στην κατάλληλη φόρμα των αποτελεσμάτων. Επίσης τυπώνονται και σε αρχείο. Επειδή τα αποτελέσματα είναι πολλά και πρέπει να κυλιστεί η φόρμα προκειμένου να τα δει ο διαχειριστής, εδώ δίνονται μέσω του αρχείου στο οποίο αποθηκεύονται (6.36), το οποίο φαίνεται παρακάτω:

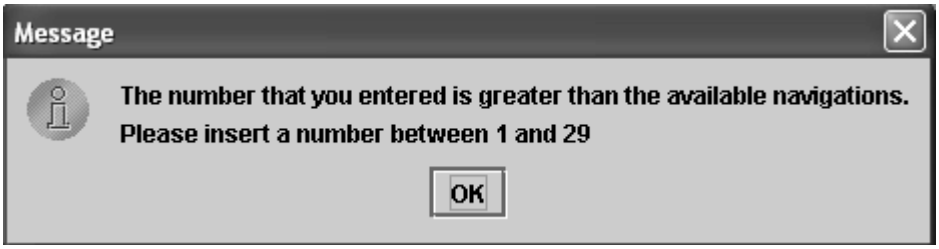
```
CLUSTERS RETRIEVED DURING THE PERIOD: 01/01/2000 - 28/09/2005

CLUSTER 1:
Users:
 Makrhs Manolis
Navigations:

R/Sports/Events/Mediterranean_Games/Recreation_and_Sports/Recreation_
and_Sports   [Makrhs Manolis]
 R/Sports/Events/Olympics/2004_-_Athens   [Makrhs Manolis]
 R/Sports/Events/Olympics/Paralympics   [Makrhs Manolis]
 R/Sports/Multi-Sports/Multi-Events/Multi-
Sports/Duathlon/Events/Duathlon/Multi-Sports/Triathlon   [Makrhs
Manolis]
 R/Sports/Organizations/Olympics/Organizations/Special_Olympics
[Makrhs Manolis]

CLUSTER 2:
Users:
 Spyropoulou Eirini
Navigations:
```

```
 R/Home/Do-It-
Yourself/Basketry/Basket_Artist/Basketry/Baskets/Basketry
[Spyropoulou Eirini]

CLUSTER 3:
Users:
 Kanellakopoulos Haralampos
Navigations:
 R/Regional/Europe/Cyprus/Education/Schools/Education/Cyprus
[Kanellakopoulos Haralampos]

R/Regional/Europe/Greece/Education/Colleges_and_Universities/Panteion
_University   [Kanellakopoulos Haralampos]

R/Regional/Europe/Greece/Education/Education/Distance_Learning/Educat
ion/Distance_Learning/Online_Courses/Online   [Kanellakopoulos
Haralampos]

CLUSTER 4:
Users:
 Zerbas Panagiwths
Navigations:

R/Shopping/Autos/Parts_and_Accessories/Brakes/Parts_and_Accessories/A
utos/Parts_and_Accessories/Electrical/Lighting/Electrical/Parts_and_A
ccessories/Electronics/Alarms/Electronics/Global_Positioning_System/C
ommunications/Global_Positioning_System/Electronics/Parts_and_Accesso
ries/Air_Conditioning/Parts_and_Accessories   [Zerbas Panagiwths]

R/Shopping/Autos/Parts_and_Accessories/Engine/Fuel/Biodiesel/Fuel/Eng
ine/Rebuilt   [Zerbas Panagiwths]

R/Shopping/Music/Equipment/Music/Shopping/Autos/Parts_and_Accessories
/Parking_Accessories   [Zerbas Panagiwths]

R/Shopping/R/Shopping/Autos/Custom_and_Collector_Cars/Antique_and_Cla
ssic/Parts_and_Accessories/Brakes/Parts_and_Accessories/Exterior/Auto
_Body_Parts   [Zerbas Panagiwths]

CLUSTER 5:
Users:
 Kanellakopoulos Haralampos
 Spyropoulou Eirini
 Fwteinou Basilikh
 Kalimerh Maria
Navigations:
 R/Home/Apartment_Living/Roommates/Apartment_Living/Homemaking
[Kanellakopoulos Haralampos]
 R/Home/Cooking/Breakfast/Bed_and_Breakfast/Bed_and_Breakfast
[Spyropoulou Eirini]
 R/Home/Cooking/Gourmet/Cooking/Quick_and_Easy   [Spyropoulou Eirini]
 R/Home/Cooking/Pizza/Cooking/Pasta/Lasagne   [Fwteinou Basilikh]
 R/Home/Cooking/Salads   [Spyropoulou Eirini, Fwteinou Basilikh]
 R/Home/Do-It-Yourself/Soaps   [Spyropoulou Eirini]
 R/Home/Pets/Dogs/Training   [Fwteinou Basilikh]
 R/Recreation/Pets   [Kalimerh Maria]

CLUSTER 6:
Users:
 Kalimerh Maria
 Kanellakopoulos Haralampos
```

```
 Christodoulou Eleni
Navigations:
 R/Arts/Entertainment/Events/Arts_and_Entertainment/Fashion/Modeling
[Kalimerh Maria]
 R/Arts/Music/Instruments/Instruments/Drums   [Kanellakopoulos
Haralampos]
 R/Arts/Photography/Communities/Online_Communities/Communities
[Christodoulou Eleni]

R/Arts/Photography/Photographers/Exhibits/Photographers/Photographers
/Fashion_and_Glamour   [Christodoulou Eleni]
 R/Arts/Photography/Photographers/Photography   [Christodoulou Eleni,
Kalimerh Maria]

R/Arts/Photography/Techniques_and_Styles/Photography/Bridges_and_Tunn
els   [Christodoulou Eleni]
 R/Arts/Radio/Guides/Directories/Directories   [Kalimerh Maria]

R/Arts/Radio/International_Broadcasters/Shortwave_and_DX_Listening/In
ternational_Broadcasters/Radio/Resources   [Kalimerh Maria]

R/Arts/Radio/Personalities/Hendrie,_Phil/Personalities/Radio/Personal
ities/Programs/Voice_Actors   [Kalimerh Maria]
```

6.36: Αρχείο επιστροφής των αποτελεσμάτων του αλγορίθμου Κ Μέσων

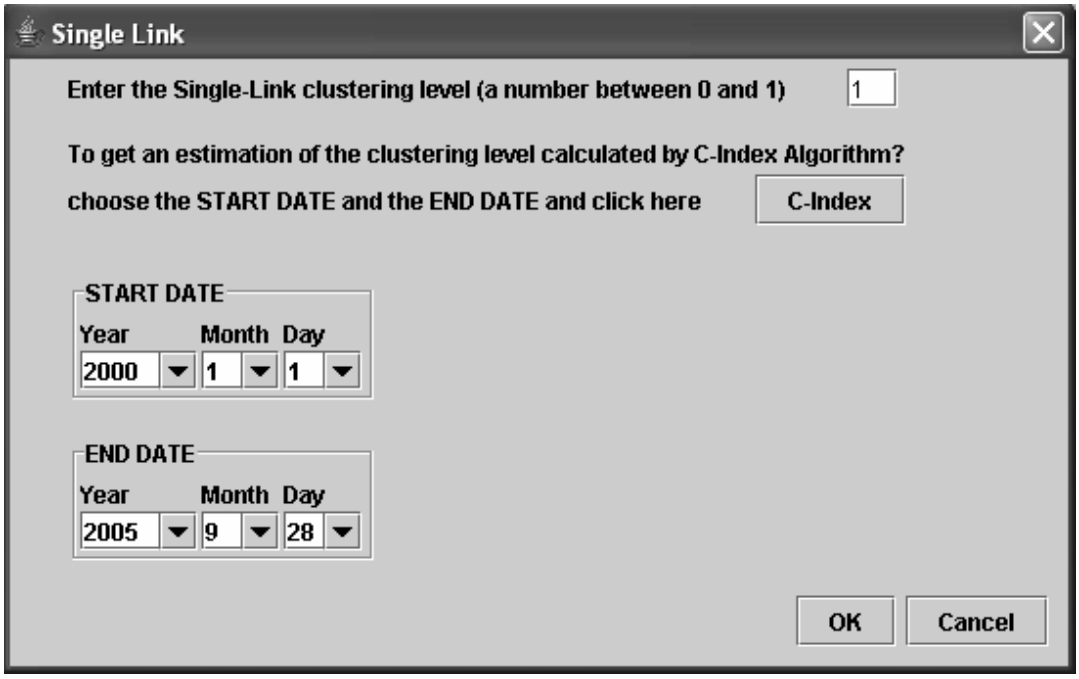
Επιστρέφονται οι συστάδες στις οποίες ομαδοποιούνται τα δεδομένα για τη χρονική περίοδο 1/1/2000 – 28/9/2005. Στην αρχή κάθε συστάδας δίνονται οι χρήστες που έχουν πραγματοποιήσει τις πλοηγήσεις αυτές. Επίσης, για περισσότερες πληροφορίες, δίνεται δίπλα σε κάθε πλοήγηση μέσα σε παρένθεση ο χρήστης που την έχει πραγματοποιήσει. Κατ' αυτόν τον τρόπο παρουσιάζεται και μια ομαδοποίηση των χρηστών του συστήματος με βάση τις πλοηγήσεις που έχουν ακολουθήσει. Για τις μεθόδους συσταδοποίησης δεν υπάρχει κάποιο αναμενόμενο αποτέλεσμα με το οποίο να μπορούμε συγκρίνουμε το λαμβανόμενο ώστε να διαπιστώσουμε αν είναι σωστό ή λάθος. Ουσιαστικά δεν τίθεται θέμα σωστού και λάθους. Διαισθητικά καταλαβαίνουμε ότι η παραπάνω ομαδοποίηση των πλοηγήσεων είναι μια «καλή» ομαδοποίηση. Αν το πλήθος των επιθυμητών συστάδων το οποίο θα εισάγει είναι μεγαλύτερο από το πλήθος των διαφορετικών πλοηγήσεων εμφανίζεται το μήνυμα του σχήματος 6.37, το οποίο τον προτρέπει να εισάγει έναν αριθμό εντός των πλαισίων λειτουργίας του αλγορίθμου.



Σχήμα 6.37: Προτρεπτικό μήνυμα

6.2.2.8 Εύρεση των συστάδων των πλοηγήσεων με τη τεχνική Μονός Σύνδεσμος

Έστω τώρα ότι ο διαχειριστής επιλέγει από τη φόρμα του σχήματος 6.30 τη μέθοδο συσταδοποίησης «μονός σύνδεσμος» (“Single Link”). Η φόρμα που παρουσιάζεται τότε είναι αυτή του σχήματος 6.38. Στη φόρμα αυτή ζητείται από το διαχειριστή να εισάγει το επίπεδο συσταδοποίησης, το οποίο είναι ένας αριθμός μεταξύ 0 και 1, και το οποίο δέχεται σαν είσοδο ο αλγόριθμος. Η προεπιλεγμένη τιμή του επιπέδου αυτού είναι 1.



Σχήμα 6.38: Επιλογή των παραμέτρων της τεχνικής «μονός σύνδεσμος»

Το χαρακτηριστικό με τη φόρμα αυτή είναι ότι αντί να αφήσει το χρήστη να μαντέψει ένα καλό επίπεδο συσταδοποίησης, πράγμα το οποίο μπορεί να τον οδηγήσει σε πολλές δοκιμές, του δίνει και την εναλλακτική να εισάγει το επίπεδο συσταδοποίησης που υπολογίζεται από τον αλγόριθμο CIndex. Όπως έχει αναφερθεί και σε προηγούμενο κεφάλαιο, ο αλγόριθμος αυτός υπολογίζει το επίπεδο συσταδοποίησης εκείνο για το οποίο η τεχνική «μονός σύνδεσμος» δημιουργεί τη βέλτιστη συσταδοποίηση. Ο διαχειριστής έχει πρόσβαση στο προτεινόμενο αυτό νούμερο με το πάτημα του κουμπιού CIndex. Σημειώνεται ότι πρέπει πρώτα να έχουν επιλεχθεί οι ημερομηνίες, έτσι ώστε ο αλγόριθμος CIndex να υπολογίσει το επίπεδο βέλτιστης συσταδοποίησης για τις πλοηγήσεις εντός των επιλεγμένων ημερομηνιών. Με το πάτημα του κουμπιού CIndex γίνονται οι απαραίτητοι υπολογισμοί και εμφανίζεται η πρόταση με τη μορφή μηνύματος. Για το συγκεκριμένο παράδειγμα λαμβάνεται το μήνυμα του σχήματος 6.39:

Σχήμα 6.39: Επίπεδο συσταδοποίησης που προτείνει ο αλγόριθμος CIndex

Υπενθυμίζεται ότι ο διαχειριστής αφήνεται ελεύθερος να επιλέξει ανάμεσα στο επίπεδο συσταδοποίησης που του προτείνεται και σε κάποιο άλλο. Αν στο παράδειγμά μας επιλέξει το επίπεδο 0.6, θα έχουμε μια «καλή» συσταδοποίηση. Επειδή η περιοχή εμφάνισης των αποτελεσμάτων πρέπει να κυλιστεί για να τα εμφανίσει όλα, παρακάτω παρατίθεται το αρχείο 6.40 στο οποίο αποθηκεύονται, όπως και στην περίπτωση εφαρμογής του αλγορίθμου των Κ Μέσων.

```
CLUSTERS RETRIEVED DURING THE PERIOD: 01/01/2000 - 28/09/2005

CLUSTER 1:
Users:
 Kalimerh Maria
Navigations:
 R/Arts/Entertainment/Events/Arts_and_Entertainment/Fashion/Modeling
[Kalimerh Maria]

CLUSTER 2:
Users:
 Kanellakopoulos Haralampos
Navigations:
 R/Arts/Music/Instruments/Instruments/Drums  [Kanellakopoulos
Haralampos]

CLUSTER 3:
Users:
 Christodoulou Eleni
 Kalimerh Maria
Navigations:

R/Arts/Photography/Photographers/Exhibits/Photographers/Photographers
/Fashion_and_Glamour  [Christodoulou Eleni]
 R/Arts/Photography/Photographers/Photography  [Christodoulou Eleni,
Kalimerh Maria]

R/Arts/Photography/Techniques_and_Styles/Photography/Bridges_and_Tunn
els  [Christodoulou Eleni]
 R/Arts/Photography/Communities/Online_Communities/Communities
[Christodoulou Eleni]

CLUSTER 4:
Users:
 Kalimerh Maria
Navigations:
```

```
R/Arts/Radio/International_Broadcasters/Shortwave_and_DX_Listening/In
ternational_Broadcasters/Radio/Resources  [Kalimerh Maria]

R/Arts/Radio/Personalities/Hendrie,_Phil/Personalities/Radio/Personal
ities/Programs/Voice_Actors  [Kalimerh Maria]
 R/Arts/Radio/Guides/Directories/Directories  [Kalimerh Maria]

CLUSTER 5:
Users:
 Kanellakopoulos Haralampos
Navigations:
 R/Home/Apartment_Living/Roommates/Apartment_Living/Homemaking
[Kanellakopoulos Haralampos]

CLUSTER 6:
Users:
 Spyropoulou Eirini
 Fwteinou Basilikh
 Kalimerh Maria
Navigations:
 R/Home/Cooking/Gourmet/Cooking/Quick_and_Easy  [Spyropoulou Eirini]
 R/Home/Cooking/Pizza/Cooking/Pasta/Lasagne  [Fwteinou Basilikh]
 R/Home/Cooking/Salads  [Spyropoulou Eirini, Fwteinou Basilikh]
 R/Home/Do-It-
Yourself/Basketry/Basket_Artist/Basketry/Baskets/Basketry
[Spyropoulou Eirini]
 R/Home/Do-It-Yourself/Soaps  [Spyropoulou Eirini]
 R/Home/Pets/Dogs/Training  [Fwteinou Basilikh]
 R/Recreation/Pets  [Kalimerh Maria]
 R/Home/Cooking/Breakfast/Bed_and_Breakfast/Bed_and_Breakfast
[Spyropoulou Eirini]

CLUSTER 7:
Users:
 Kanellakopoulos Haralampos
Navigations:

R/Regional/Europe/Greece/Education/Colleges_and_Universities/Panteion
_University  [Kanellakopoulos Haralampos]

R/Regional/Europe/Greece/Education/Education/Distance_Learning/Educat
ion/Distance_Learning/Online_Courses/Online  [Kanellakopoulos
Haralampos]
 R/Regional/Europe/Cyprus/Education/Schools/Education/Cyprus
[Kanellakopoulos Haralampos]

CLUSTER 8:
Users:
 Zerbas Panagiwths
Navigations:

R/Shopping/Autos/Parts_and_Accessories/Engine/Fuel/Biodiesel/Fuel/Eng
ine/Rebuilt  [Zerbas Panagiwths]

R/Shopping/Music/Equipment/Music/Shopping/Autos/Parts_and_Accessories
/Parking_Accessories  [Zerbas Panagiwths]

R/Shopping/R/Shopping/Autos/Custom_and_Collector_Cars/Antique_and_Cla
ssic/Parts_and_Accessories/Brakes/Parts_and_Accessories/Exterior/Auto
_Body_Parts  [Zerbas Panagiwths]
```

```
R/Shopping/Autos/Parts_and_Accessories/Brakes/Parts_and_Accessories/A
utos/Parts_and_Accessories/Electrical/Lighting/Electrical/Parts_and_A
ccessories/Electronics/Alarms/Electronics/Global_Positioning_System/C
ommunications/Global_Positioning_System/Electronics/Parts_and_Accesso
ries/Air_Conditioning/Parts_and_Accessories  [Zerbas Panagiwths]

CLUSTER 9:
Users:
 Makrhs Manolis
Navigations:
 R/Sports/Events/Olympics/2004_-_Athens  [Makrhs Manolis]
 R/Sports/Events/Olympics/Paralympics  [Makrhs Manolis]
 R/Sports/Organizations/Olympics/Organizations/Special_Olympics
[Makrhs Manolis]

R/Sports/Events/Mediterranean_Games/Recreation_and_Sports/Recreation_
and_Sports  [Makrhs Manolis]

CLUSTER 10:
Users:
 Makrhs Manolis
Navigations:
 R/Sports/Multi-Sports/Multi-Events/Multi-
Sports/Duathlon/Events/Duathlon/Multi-Sports/Triathlon  [Makrhs
Manolis]
```

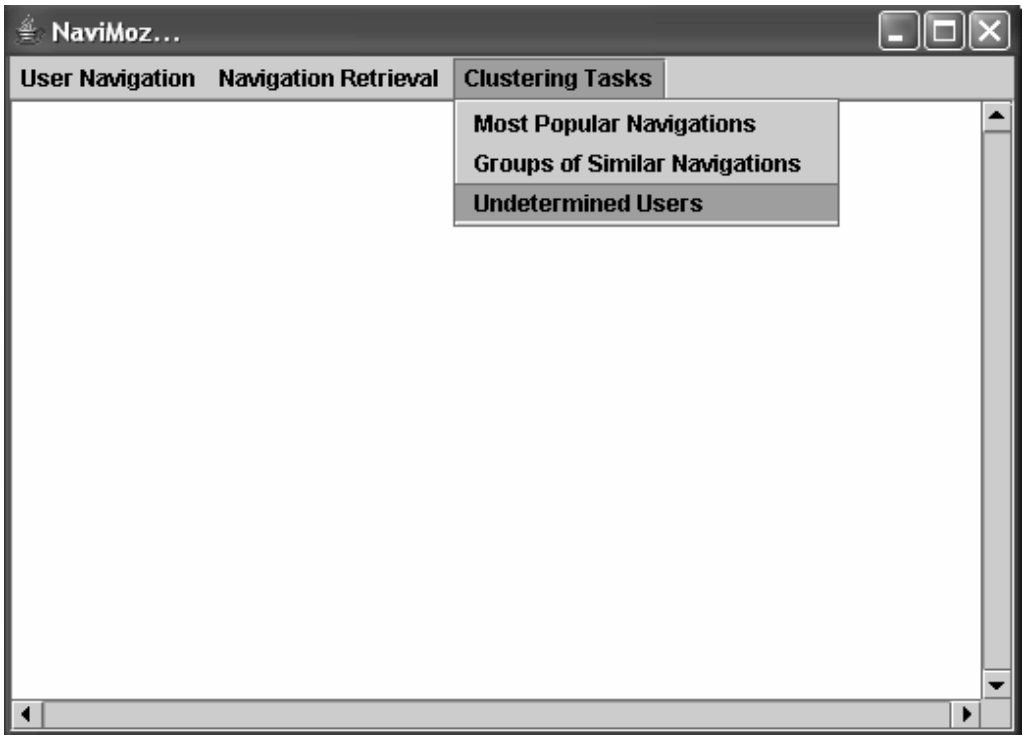
6.40: Αρχείο επιστροφής των αποτελεσμάτων της τεχνικής «μονός σύνδεσμος»

Επιστρέφονται οι δημιουργούμενες συστάδες των πλοηγήσεων καθώς και οι χρήστες που τις έχουν πραγματοποιήσει, όπως ακριβώς και στην περίπτωση των Κ Μέσων. Έτσι έχουμε και στην περίπτωση αυτή μια ομαδοποίηση των χρηστών του συστήματος. Και πάλι, διαισθητικά καταλαβαίνουμε ότι πρόκειται για μια «καλή» συσταδοποίηση. Αν ο διαχειριστής εισάγει στο ζητούμενο επίπεδο συσταδοποίησης κάτι άλλο εκτός από αριθμό (για παράδειγμα ένα σύμβολο), ή εισάγει αριθμό μικρότερο του 0 ή μεγαλύτερο του 1, εμφανίζεται το προειδοποιητικό μήνυμα του σχήματος 6.41, το οποίο τον ενημερώνει για το λάθος του και του υποδεικνύει να το διορθώσει.

Σχήμα 6.41: Προειδοποιητικό μήνυμα

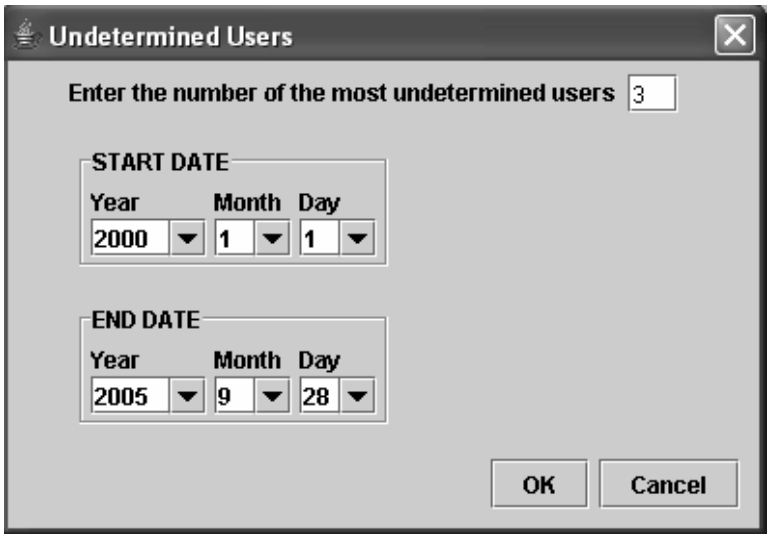
6.2.2.9 Εύρεση των πιο αναποφάσιστων χρηστών του συστήματος

Έστω τώρα ότι ο διαχειριστής επιθυμεί να δει ποιοι είναι οι περισσότερο αναποφάσιστοι χρήστες του συστήματος. Η εφαρμογή αυτή είναι η τελευταία από την τρίτη κατηγορία εργασιών. Μπορεί να επιλέξει την εκτέλεσή της αν επιλέξει “Undetermined Users” από το μενού “Clustering Tasks”, όπως φαίνεται και στο σχήμα 6.42.

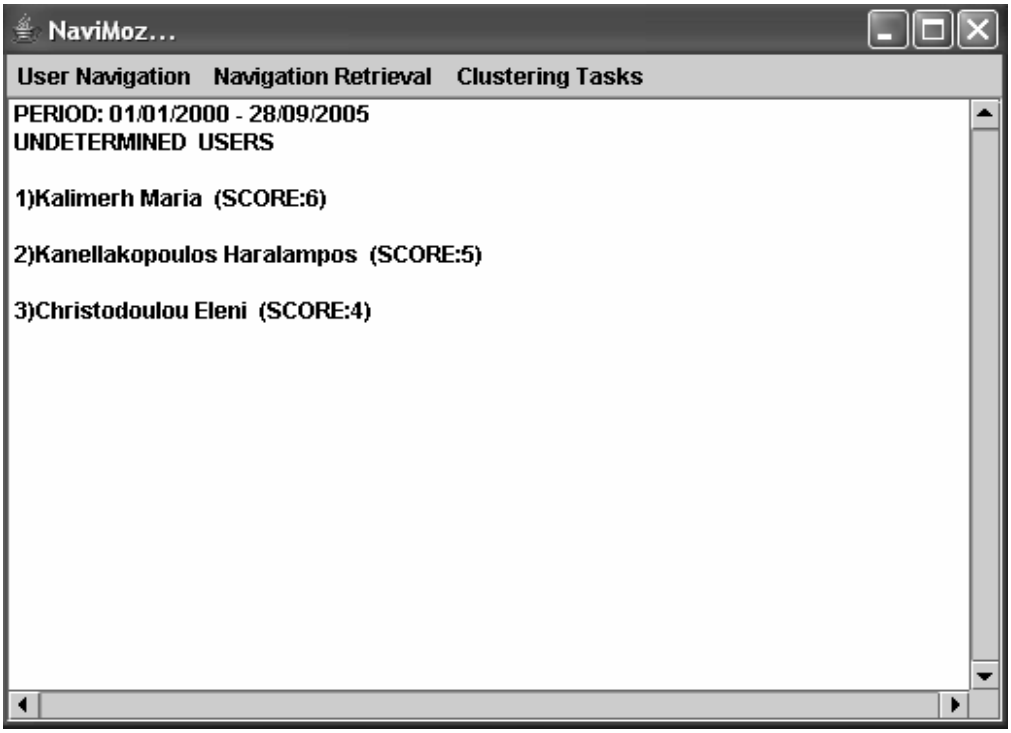


Σχήμα 6.42: Επιλογή της εργασίας “Undetermined Users”

Η φόρμα που εμφανίζεται τότε είναι αυτή του σχήματος 6.43. Η φόρμα αυτή ζητά από το διαχειριστή να εισάγει το πλήθος των περισσότερο αναποφάσιστων χρηστών του συστήματος και το χρονικό διάστημα ενδιαφέροντος. Η αναποφασιστικότητα ενός χρήστη, όπως έχει αναλυτικά εξηγηθεί και σε προηγούμενο κεφάλαιο, μετράται με βάση το πόσες φορές έχει πατήσει τα κουμπιά Back και Forward στις πλοηγήσεις του. Η προεπιλεγμένη τιμή είναι 1, έτσι σε περίπτωση που ο διαχειριστής δεν εισάγει αριθμό θα επιστραφεί ο πιο αναποφάσιστος χρήστης του συστήματος. Αν ο διαχειριστής επιλέξει να δει τους 3 πιο αναποφάσιστους χρήστες, το αποτέλεσμα που θα λάβει θα είναι αυτό του σχήματος 6.44. Στα αποτελέσματα φαίνονται οι 3 πιο αναποφάσιστοι χρήστες κατά φθίνοντα βαθμό αναποφασιστικότητας. Μέσα σε παρένθεση δίνεται με την ένδειξη “SCORE” ο βαθμός αυτός, ο οποίος αντιστοιχεί στο πλήθος των Back και forward που έχουν πραγματοποιήσει. Αν ανατρέξουμε στη Βάση Δεδομένων θα δούμε ότι τα αποτελέσματα είναι σωστά.



Σχήμα 6.43: Επιλογή των παραμέτρων της εργασίας "Undetermined Users"

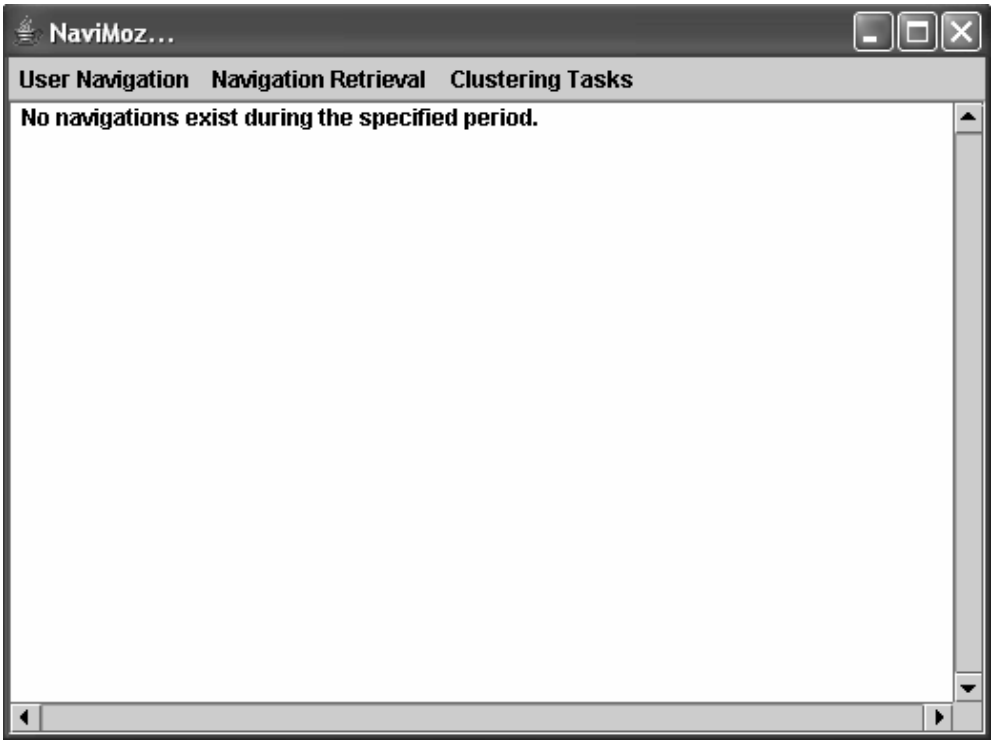


Σχήμα 6.44: Αποτελέσματα της εργασίας "Undetermined Users"

Αν ο αριθμός που θα επιλέξει ο διαχειριστής είναι μεγαλύτερος από το πλήθος των χρηστών του συστήματος, προβάλλεται ενημερωτικό μήνυμα το οποίο τον προτρέπει να επιλέξει έναν αριθμό μικρότερο από το πλήθος των χρηστών. Στο παράδειγμά μας, όπου οι εγγεγραμμένοι στο σύστημα χρήστες είναι 7, αν ο διαχειριστής εισάγει το νούμερο 10, εμφανίζεται το μήνυμα του σχήματος 6.45:

Σχήμα 6.45: Προτρεπτικό μήνυμα

Αν μεταξύ των ημερομηνιών που θα επιλέξει ο διαχειριστής δεν έχουν πραγματοποιηθεί πλοηγήσεις, η φόρμα των αποτελεσμάτων θα επιστρέψει με κατάλληλο μήνυμα, όπως φαίνεται στο σχήμα 6.46.



Σχήμα 6.46: Μήνυμα σε περίπτωση μη ύπαρξης πλοηγήσεων

Αντίστοιχα μηνύματα παρουσιάζονται για όλες τις εργασίες σε περίπτωση που δεν έχουν πραγματοποιηθεί πλοηγήσεις στο επιλεγμένο χρονικό διάστημα.

7

Επίλογος

Το κεφάλαιο αυτό ολοκληρώνει την παρουσίαση της διπλωματικής εργασίας με μια σύντομη ανασκόπηση αυτής και αναφορά σε πιθανές μελλοντικές επεκτάσεις της.

7.1 Σύνοψη και Συμπεράσματα

Οι πύλες καταλόγων επιλέγονται συχνά από τους χρήστες για τις αναζητήσεις τους στο διαδίκτυο, εξαιτίας της δομημένης πληροφορίας που παρέχουν. Εξαιτίας της μεγάλης αυτής χρήσης των πυλών καταλόγων, θα είναι εξαιρετικά χρήσιμη η επεξεργασία των πλοηγήσεων των χρηστών με στόχο την τροποποίηση της πύλης κατά τρόπο που να εξυπηρετεί την αποτελεσματικότερη αναζήτηση της πληροφορίας. Σκοπός της συγκεκριμένης διπλωματικής εργασίας ήταν η ανάπτυξη ενός νέου συστήματος επεξεργασίας των πλοηγήσεων των χρηστών μιας πύλης από το διαχειριστή της πύλης αυτής και εξόρυξης γνώσης από αυτές. Ο διαχειριστής έχει τη δυνατότητα να θέτει κάποιες ερωτήσεις μέσω μιας κατάλληλα διαμορφωμένης διασύνδεσης χρήστη και να λαμβάνει μέσω αυτής τα αποτελέσματα. Οι ερωτήσεις που μπορεί να εκτελεί ποικίλλουν από απλές ερωτήσεις ταυτοποίησης χρηστών, μέχρι εξόρυξη δεδομένων με βάση κάποια πλοήγηση-πρότυπο, ή ακόμα και εργασίες συσταδοποίησης των πλοηγήσεων και των χρηστών.

Η επεξεργασία των πλοηγήσεων χρηστών υπήρξε αντικείμενο πολλών ερευνών στο παρελθόν. Η βασική διαφορά του συστήματος NaviMoz που αναπτύσσεται εδώ από τα ήδη υπάρχοντα συστήματα είναι ότι οι εργασίες του βασίζονται στην επεξεργασία της πορείας του κάθε χρήστη ως μια ολότητα, πράγμα που σημαίνει ότι η επεξεργασία αλλά και η εξόρυξη γνώσης πραγματοποιείται με εξέταση της συνολικής πλοήγησης του κάθε χρήστη. Με τον όρο πλοήγηση δεν εννοούμε ακολουθία σελίδων, όπως συνηθίζεται, αλλά ακολουθία θεματικών κατηγοριών που επισκέπτεται ο χρήστης μέχρι να βρει τις ιστοσελίδες που θέλει. Έτσι, για παράδειγμα, εντοπίζονται ομοιότητες και διαφορές μεταξύ της ολικής πορείας των χρηστών με την ομοιότητα του περιεχομένου των πλοηγήσεων, τη δομική τους απόσταση αλλά και το μέγεθός τους να διαδραματίζουν πρωτεύοντα ρόλο. Αντιθέτως, τα προϋπάρχοντα

συστήματα επεξεργάζονται τις σελίδες που επισκέπτεται ο χρήστης, δηλαδή ουσιαστικά τα σημεία κατάληξης των πλοηγήσεων. Στηρίζονται δηλαδή στο Web log και πραγματοποιούν την εξόρυξη γνώσης και τις όποιες ομαδοποιήσεις με βάση το περιεχόμενο των σελίδων που επισκέφτηκε ο χρήστης. Μία ακόμη διαφορά του συστήματος NaviMoz από τα προϋπάρχοντα, είναι ότι εξετάζει πλοηγήσεις που πραγματοποιούνται σε ιεραρχίες πυλών καταλόγων, ενώ τα ήδη προϋπάρχοντα δεν επικεντρώνονται σε εξέταση ιεραρχιών, αλλά αντίθετα εξετάζουν τις σελίδες μιας πλοήγησης γενικότερα.

Στη συνέχεια, παρουσιάστηκε το θεωρητικό υπόβαθρο της διπλωματικής εργασίας. Αρχικά δόθηκε ο βασικός ορισμός του γράφου πύλης καταλόγου και στη συνέχεια παρατέθηκαν ουσιαστικές πληροφορίες πάνω στην εξόρυξη δεδομένων (Data Mining), στις εργασίες συσταδοποίησης καθώς και στην επεξήγηση της εύρεσης της δομικής απόστασης μεταξύ δύο συμβολοακολουθιών, εργασία στην οποία στηρίζονται πολλές εφαρμογές του συστήματος. Στην περίπτωση των εργασιών συσταδοποίησης παρουσιάστηκαν, εκτός από γενικές πληροφορίες, και οι δύο αλγόριθμοι που χρησιμοποιούνται από το σύστημα για συσταδοποίηση, η τεχνική των K Μέσων και η τεχνική «μονός σύνδεσμος», όπως επίσης και ένας βοηθητικός αλγόριθμος συσταδοποίησης στην περίπτωση χρήσης της τεχνικής «μονός σύνδεσμος», ο C_Index.

Παρουσιάστηκε ακόμα η ανάλυση των απαιτήσεων από το σύστημα και η σχεδίαση αυτού, όπου έγινε και χρήση επεξηγηματικών διαγραμμάτων. Το σύστημα χωρίστηκε σε υποσυστήματα για την ανάλυση των απαιτήσεων, ενώ στην περίπτωση της σχεδίασης τα υποσυστήματα υποδιαιρέθηκαν στις επιμέρους εφαρμογές που αυτά επιτελούν.

Στη συνέχεια, παρουσιάστηκε η υλοποίηση του συστήματος. Αρχικά επεξηγήθηκαν οι σημαντικότεροι αλγόριθμοι που χρησιμοποιήθηκαν στην ανάπτυξη της εφαρμογής. Επίσης, δόθηκαν στοιχεία για την ανάπτυξη της εφαρμογής σε γλώσσα προγραμματισμού Java με συνοπτική περιγραφή των κλάσεων που δημιουργήθηκαν. Τέλος, επεξηγήθηκε η διαδικασία εγκατάστασης και αρχικοποίησης του συστήματος.

Τέλος, παρουσιάστηκε ένα σενάριο χρήσης του συστήματος και περιγράφηκαν οι φόρμες που το σύστημα χρησιμοποιεί για την υποβολή ερωτήσεων, την εμφάνιση μηνυμάτων και την επιστροφή των αποτελεσμάτων.

7.2 Μελλοντικές Επεκτάσεις

Στο σημείο αυτό, έχοντας πια μια συνολική εικόνα της λειτουργίας του συστήματος NaviMoz, μπορούν να παρουσιαστούν κάποιες πιθανές μελλοντικές επεκτάσεις αυτού, οι οποίες αναφέρονται παρακάτω:

- Μια άλλη πιθανή εργασία του συστήματος είναι η εύρεση των πλοηγήσεων εκείνων που είναι ίδιες μέχρι ένα σημείο, το οποίο ορίζεται από το διαχειριστή, και στη συνέχεια διαφοροποιούνται. Για παράδειγμα, μπορεί ο διαχειριστής να επιθυμεί να δει τις πλοηγήσεις αυτές των οποίων το πρώτο 50% είναι ίδιο και στη συνέχεια διαφοροποιούνται. Εκτός από απλή εύρεση των πλοηγήσεων αυτών, το σύστημα θα μπορεί να πραγματοποιεί άλλη μια ομαδοποίηση των χρηστών στηριζόμενο στο νέο αυτό κριτήριο. Κατ' αυτόν τον τρόπο μπορεί να βρεθεί το σημείο μεταβολής του ενδιαφέροντος των χρηστών, γνώση η οποία μπορεί να χρησιμοποιηθεί για την περαιτέρω βελτίωση της χρηστικότητας της πύλης.
- Το σύστημα NaviMoz πραγματοποιεί μία εξόρυξη δεδομένων η οποία μπορεί να χρησιμοποιηθεί κατά πολλούς τρόπους στη συνέχεια. Απώτερος στόχος είναι να ενταχθεί το σύστημα αυτό σε ένα γενικότερο πλαίσιο εργασιών αύξησης της χρηστικότητας της πύλης. Έτσι λοιπόν, τα αποτελέσματα του συστήματος μπορούν να χρησιμοποιηθούν για την προσωποποίηση της πύλης έτσι ώστε να ταιριάζει στην ιδιοσυγκρασία και στις προτιμήσεις του κάθε χρήστη ή ομάδων χρηστών. Για παράδειγμα, όσον αφορά στους αναποφάσιστους χρήστες, θα μπορούν να τους παρέχονται περισσότερες πληροφορίες σχετικές με τους συνδέσμους που υπάρχουν σε κάθε κατηγορία της πύλης, με βάση τις οποίες θα μπορούν να επιλέγουν την επόμενη τοποθεσία που θα επισκεφτούν, αποφεύγοντας τις συνεχείς παλινδρομήσεις που είναι πιθανό να τους κουράσουν. Επίσης, με βάση τα αποτελέσματα της εύρεσης των πλοηγήσεων των χρηστών, θα μπορούν να τους παρέχονται κυρίως οι σύνδεσμοι προς τις περιοχές όπου συγκεντρώνεται το ενδιαφέρον τους.

8

Βιβλιογραφία

[ACK+01] Alexaki S., Christophides V., Karvounarakis G., Plexousakis D., ICS-FORTH (Institute of Computer Science-Foundation for Research and Technology Hellas),2001

[AH02] Anderson C. R., Horvitz E.: A Dynamic Personalized Start Page. In Proceedings of the 11th WWW Conference 2002.

[AR03] A. Ajith, V. Ramos. Web Usage Mining Using Artificial Ant Colony Clustering and Genetic Programming. Congress on Evolutionary Computation, IEEE Press ISBN 078-0378-04-0 pp 1384-1391, Canberra, Australia, Dec 2003.

[Bor94] Stephen P. Borgatti, How to explain Hierarchical Clustering, http://www.analytictech.com/networks/hiclus.htm, 1994

[CLR+01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, MIT Press, Second Edition, 2001

[DCS+] Dalamagas T., Cheng T., Sellis T., Winkel K., Συσταδοποίηση Αρχείων XML με χρήση Δομικών Περιλήψεων και Μετρικής Δομικής Ομοιότητας.

[DMS03] Dalamagas T., Meliou A., Sellis T.: Modeling and Manipulating the Structure of Portal Catalogs, 2003

[FPS96] Fayad U., Piatetsky-Shapiro G., Smyth P., The KDD Process for Extracting Useful Knowledge from Volumes of Data, Vol. 39, No. 11, November 1996

[FS05] X. Fang, O. R. Liu Sheng. Designing a Better Web Portal for Digital Government: A Web-mining Based Approach, http://diggov.org/library/library/dgo2005/demosb/fang_designing.pdf

[GR69]J. C. Gower and G. J. S. Ross, Minimum spanning trees and single linkage cluster analysis, *Applied Statistics*, 18:54-64, 1969.

[KAO04] Kaneta Y., Md. Ahaduzzaman Munna, T. Ohkawa. A Method of Extracting Sentences Related to Protein Interaction from Literature using a Structure Database. Second European Workshop on Data Mining and Text Mining for Bioinformatics, ECML PKDD, Italy, September 2004.

[KJ00] Kamdar T. Joshi A., On Creating Adaptive Web Sites using Web Log Mining, TR-CS-00-05. Department of Computer Science and Electrical Engineering University of Maryland, Baltimore Country, (2000).

[KNO+03] L. Krishnamurthy , J. Nadeau , G. Ozsoyoglu , M. Ozsoyoglu, G. Schaeffer, M. Tasan, W. Xu . Pathways Database System: An integrated set of tools for biological pathways. ACM 1-58113-624-2/03/2003

[MDL+00] Mobasher B., H. Dai, T. Luo, Y. Sung, J. Zhu.: Integrating Web Usage and Content Mining for More Effective Personalization. In Proceedings of the International Conference on E-Commerce and Web Technologies, Greenwich, UK, 165-176, 2000

[PLB+04] R. G. Pensa1, C. Leschi1, J. Besson and J. Boulicaut. Assessment of discretization techniques for relevant pattern discovery from gene expression data.2nd Workshop on Data Mining in Bioinformatics, Seattle, USA, August 2004

[PPP+04]D. Pierrakos, G. Paliouras, C. Papatheodorou, V. Karakaletsis and M. Dikaiakos. Web community directories: A new approach to web personalization. In B.B et al., editor, *Web Mining: From Web to Semantic Web, EMWF 2003,* volume 3209 of LNCS, pages 113-129. Springer, 2004.

[SF98] Spiliopoulou N., Faulstich L. C.: WUM: A Web Utilization Miner. In International Workshop on the Web and Databases, Valencia, Spain, (1998)

[Tek04] Teknomo K., Ph. D., KMeans Clustering Tutorial, December 2004

[TK02] F. Toolan, N. Kusmerick. Mining web logs for personalized site maps. In Proc. Workshop Mining for Enhanced Web Search, 2002. Int. Conf. Web Information Systems Engineering.

[Van04] Vanessa (Cheng-Hsien Chih), Critical Review of Technology #2- DMOZ Open Directory Project, EIL 590, E2 Fall 2004

[Voo85] Ellen M. Voorhees, The Effectiveness and Efficiency of Agglomerative Hierarchic Clustering in Document Retrieval, Ph. D. Thesis, TR 85-705, October 1985.

[WF74] Robert A. Wagner, Michael J. Fischer, The String to String Correction Problem, Journal of the Association for the Computer Machinery, Vol 21, No.1, pp.168-173, January 1974.

[Yeu05] Ka Yee Yeung, Center for Expression Arrays, University of Washington, http://faculty.washington.edu/kayee/talks/cluster101.ppt, 2005

[Τζα02] Σπύρος Γ. Τζαφέστας, Υπολογιστική Νοημοσύνη-Τόμος Α:Μεθοδολογίες, Εθνικό Μετσόβιο Πολυτεχνείο, 2002