Προσθήκη Εργαλείων και Πηγών στο Σύστημα Αναζήτησης ezDL

Α.Τ.Ε.Ι. ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή εργασία του
Ιώαννη Παρασκευόπουλου

Υπεύθυνος καθηγητής
Dr. Μιχάλης Σαλαμπάσης

Θεσσαλονίκη, Σεπτέμβιος 2013

Περιεχόμενα

Θα ήθελα να ευχαριστήσω τον καθηγητή μου Dr. Μιχάλη Σαλαμπάση για την καθοδήγηση και την υποστήριξη του καθ' όλη τη διάρκεια διεκπεραίωσης της πτυχιακής μου.

Εισαγωγή

Η αξία της πληροφορίας είναι γνωστή στην ιστορία της ανθρωπότητας. Η πληροφορία είναι το στοιχείο γνώσης που δίνει αξία στα πράγματα και στα συμβάντα γύρω μας. Σήμερα, οι επιχειρήσεις, οι κρατικοί οργανισμοί αλλά και ο κάθε άνθρωπος ατομικά συλλέγουν πληροφορίες και βάσει αυτών λαμβάνουν τις αποφάσεις τους. Θα λέγαμε, λοιπόν, πως η πληροφορία και η ανάκτησή της μπορούν να επηρεάσουν οποιοδήποτε σύστημα, άρα και την ίδια την ζωή.

Πολλές φορές η πληροφορία συγχέεται με την έννοια των δεδομένων. Κάτι τέτοιο είναι αναμενόμενο, καθώς υπάρχει στενή σχέση μεταξύ τους και, κατά συνέπεια, δεν μπορεί να υπάρξει πληροφορία χωρίς δεδομένα. Εύλογα, λοιπόν, μπορούμε να πούμε ότι η ***πληροφορία είναι δεδομένα με σημασία***. Τα δεδομένα σήμερα παράγονται σε τεράστιες ποσότητες, ενώ ογκώδη αρχεία πληροφοριών από το παρελθόν αποθηκεύονται σε νέους χώρους. Η ύπαρξη εταιρειών όπως η Google, το Facebook, το Amazon, το Twitter λαμβάνουν και αποθηκεύουν τεράστιες ποσότητες δεδομένων, ενώ διαφόρων ειδών επιχειρήσεις και οργανισμοί δημιουργούν συλλογές δεδομένων σε διάφορα πεδία (π.χ. Ιατρική, Πατέντες, Βιβλιογραφία) και πολλές φορές τα δεδομένα αυτά χρησιμοποιούνται για την παροχή διαφόρων υπηρεσιών.

Η σημασία, λοιπόν, των ***Συστημάτων Αναζήτησης*** δεδομένων, είτε αυτά βρίσκονται στο Web είτε σε κάποιες ΒΔ, είναι κάτι που απασχολεί και θα συνεχίσει να απασχολεί τον τομέα της Πληροφορικής. Η ταχύτητα ανάπτυξης του Web, αλλά και η συνεχής αλλαγή των ιστότοπων κάνει την ανάκτηση της ζητούμενης πληροφορίας πιο δύσκολη. Η ποσότητα και η εγκυρότητα των πληροφοριών που επιστρέφουν τα διάφορα συστήματα αναζήτησης είναι κάποια από τα κριτήρια που καθορίζουν την επιτυχία της ανάκτησης της πληροφορίας.

Το πρόβλημα γίνεται μεγαλύτερο όταν η πληροφορία που αναζητά κάποιος δεν είναι ορατή από τις γνωστές μηχανές αναζήτησης όπως η Google, η Bing κ.α. Τα δεδομένα αυτά αποτελούν το λεγόμενο ***Deep Web***, το μεγαλύτερο μέρος του οποίου βρίσκεται σε ΒΔ ελεγχόμενης χρήσης. Συνεπώς, η χρήση των παραδοσιακών μηχανών αναζήτησης δεν είναι πάντα επαρκής, γεγονός που επιβάλλει τη δημιουργία επαγγελματικών συστημάτων αναζήτησης στοχευμένου περιεχομένου. Βέβαια, εδώ πρέπει να αναφέρουμε πως η απλότητα, η ευχρηστία αλλά και η επιστροφή καλών αποτελεσμάτων που προσφέρουν οι σύγχρονες μηχανές αναζήτησης όπως η Google, τις καθιστούν κυρίαρχες στο τομέα της ανάκτησης πληροφορίας.

Σύμφωνα με τα παραπάνω, τα γνωστά συστήματα αναζήτησης δεν ήταν πάντα επαρκή. Το κενό αυτό ήρθε να καλύψει μια νέα τεχνολογία αναζήτησης που φέρει τον τίτλο ***Federated Search***. Ο τρόπος λειτουργίας της τεχνολογίας αυτής έγκειται στην ταυτόχρονη ή παράλληλη αναζήτηση σε πολλαπλές πηγές δεδομένων χρησιμοποιώντας έναν ενιαίο χώρο υποβολής ερωτημάτων. Με τον τρόπο αυτό ο χρήστης υποβάλλει ένα ερώτημα σε πολλές πηγές, χωρίς να χρειάζεται να εξοικειωθεί με τα αρχικά περιβάλλοντα αναζήτησης. Προφανώς, η ευχρηστία αυτή βοηθάει στην καλύτερη αξιοποίηση και επεξεργασία των τελικών αποτελεσμάτων, πράγμα που έχει ως αποτέλεσμα την υψηλή παραγωγικότητα. Ένα εγγενές χαρακτηριστικό της ταυτόχρονης αναζήτησης από πολλαπλές πηγές είναι ο μεγάλος όγκος πληροφορίας ***(information overload)*** που συλλέγεται, γεγονός που δημιουργεί την ανάγκη διαχείρισης, οργάνωσης και ταξινόμησης των τελικών αποτελεσμάτων. Οι διαδικασίες αυτές αποτελούν κομβικά στοιχεία της αρχιτεκτονικής των ***μετα-μηχανών***

αναζήτησης, τις οποίες θα αναλύσουμε παρακάτω.

Το ***ezDL*** είναι ένα διαδραστικό εργαλείο αναζήτησης αποτελεσμάτων, μια πλατφόρμα ανάπτυξης για διαδραστικά συστήματα ανάκτησης πληροφοριών και ένα σύστημα αξιολόγησης για τα αποτελέσματα των αναζητήσεων που εκτελεί. Ως μια μετα-μηχανή αναζήτησης ετερογενών πηγών σε ψηφιακές βιβλιοθήκες, το ezDL σχετίζεται με όλα τα πεδία του χώρου της τεχνολογίας της Πληροφορικής που αναφέραμε παραπάνω. Δεδομένου ότι η παρούσα πτυχιακή εργασία αναφέρεται σε ανάπτυξη εργαλείου της πλατφόρμας του ezDL, θα κάνουμε μια σύντομη περιγραφή των πεδίων αυτών παρακάτω. Έπειτα, θα περιγράψουμε την υλοποίηση του ίδιου του εργαλείου και των τεχνολογιών που χρησιμοποιήθηκαν για την εκπόνησή της.

Συστήματα Αναζήτησης

Η διαδικτυακή αναζήτηση πληροφορίας είναι μάλλον από τα πιο συναρπαστικά πράγματα που προσφέρει σήμερα το Internet. Η αμεσότητα και η πληθώρα των αποτελεσμάτων μοιάζουν εξωπραγματικά αφού σε μερικά μόνο δευτερόλεπτα ο χρήστης μπορεί να έχει μπροστά του απαντήσεις στα πιο πολλά του ερωτήματα.

Τα πιο γνωστά εργαλεία αναζήτησης που χρησιμοποιούνται σήμερα για την αναζήτηση πληροφορίας στο ευρύ κοινό είναι δύο: οι θεματικοί κατάλογοι (π.χ. Yahoo) και οι μηχανές αναζήτησης (π.χ. Google). Πολλές φορές, λόγω της υποστήριξης επιπρόσθετων τεχνικών αναζήτησης, κάποια από τα συστήματα χαρακτηρίζονται υβριδικά.

Οι σύγχρονες τεχνολογικές αλλαγές δημιούργησαν την ανάγκη ύπαρξης πιο επαγγελματικών συστημάτων αναζήτησης. Αυτά επιστρέφουν αποτελέσματα από πηγές που δεν είναι προσβάσιμες από τα κλασικά συστήματα αναζήτησης και χρησιμοποιούν εξειδικευμένες αναλυτικές μεθόδους ανάλογα με τις ανάγκες του εκάστοτε χρήστη.

Παρακάτω περιγράφονται τα δυο κλασικά συστήματα αναζήτησης. Στη συνέχεια, ακολουθεί μια εισαγωγή στις έννοιες *Deep Web* και *Deep Web Harvesting Engines*.

Θεματικοί κατάλογοι

Οι θεματικοί κατάλογοι, αν και έχουν χάσει την αίγλη τους, αποτελούν αξιόπιστες πηγές πληροφορίας. Πρόκειται για τεράστιες συλλογές από υπερσυνδέσμους ιστοσελίδων, ταξινομημένες και κατηγοριοποιημένες σε ιεραρχική δομή ανάλογα με το περιεχόμενό τους. Η κατηγοριοποίηση αυτή γίνεται σε πολλές περιπτώσεις από εξειδικευμένο προσωπικό, τους λεγόμενους *editors,* ή από κάποια εθελοντική κοινότητα. Είναι φανερό πως η ποιότητα των θεματικών καταλόγων ποικίλλει ανάλογα με τους *editors* τους αλλά και με το χαρακτήρα της υπηρεσίας που θέλουν να προσφέρουν. Οι κύριοι τύποι θεματικών καταλόγων είναι οι Ακαδημαϊκοί-Επαγγελματικοί και οι Εμπορικοί θεματικοί κατάλογοι. Οι πρώτοι έχουν σχέση συνήθως με βιβλιοθήκες και ακαδημαϊκά ιδρύματα και εξυπηρετούν τις ερευνητικές ανάγκες τους. Φημίζονται για το σοβαρό χαρακτήρα τους και χρησιμοποιούν αυστηρά κριτήρια για την καταχώρηση και την κατηγοριοποίηση των συνδέσμων. Οι Εμπορικοί θεματικοί κατάλογοι αποσκοπούν στο κέρδος και απευθύνονται στο ευρύ κοινό. Τα θέματα που εμπεριέχουν είναι ποικίλου ενδιαφέροντος και επιδιώκουν να προσελκύσουν όσο το δυνατόν περισσότερους χρήστες.

Όσον αφορά στη λειτουργία αναζήτησης σε θεματικούς καταλόγους, αυτή μπορεί να γίνει είτε ακολουθώντας την ιεραρχική δομή θεματολογίας είτε με τη χρήση ενός μηχανισμού ευρετηρίου. Η χρήση του ευρετηρίου δίνει τη δυνατότητα ταχείας εύρεσης εγγραφών που μπορεί να βρίσκονται βαθύτερα στην ιεραρχία του καταλόγου. Αυτός ο τρόπος αναζήτησης θυμίζει πολύ τις μηχανές αναζήτησης που θα περιγραφούν παρακάτω.

Μηχανές αναζήτησης

Οι μηχανές αναζήτησης είναι συστήματα λογισμικού σχεδιασμένα για την αναζήτηση πληροφορίας στο Web. Τα αποτελέσματα αναζήτησης επιστρέφονται ανάλογα με το ερώτημα που θέτει ο χρήστης και παρουσιάζονται υπο την μορφή των *Search Engine Results Pages*. Οι πληροφορίες μπορεί να είναι σύνδεσμοι σε ιστοσελίδες, εικόνες ή και άλλου τύπου αρχεία, ανάλογα με τις ρυθμίσεις του χρήστη. Αντίθετα με τους θεματικούς καταλόγους, στους οποίους η επεξεργασία γίνεται από ανθρώπους, οι μηχανές αναζήτησης χρησιμοποιούν προγράμματα για την δημιουργία των περιεχομένων τους. Τα προγράμματα αυτά λέγονται *crawlers* και είναι ικανά να σαρώνουν πληθώρα από ιστοσελίδες σε μικρό χρονικό διάστημα, γεγονός που δίνει την ευχέρεια της ενημέρωσης της πληροφορίας των μηχανών αναζήτησης σε πραγματικό χρόνο.

Παρακάτω περιγράφονται τα τρία κύρια μέρη που απαρτίζουν μια μηχανή αναζήτησης.

Crawler

Ο *Web Crawler* είναι το λογισμικό που έχει στόχο την εύρεση των ιστοσελίδων και την παράδοση αυτών στον *indexer*. Η βασική λειτουργία τους αποσκοπεί στην εκτέλεση αιτήματος σε κάποιο διακομιστή και στη λήψη της ζητούμενης σελίδας. Η εκτέλεση αυτών των αιτημάτων γίνεται με τη χρήση διευθύσεων URL που είτε προϋπάρχουν σε κάποια λίστα που τους δίνεται, είτε τοποθετούνται σε μια ουρά κατα την σάρωση μιας ιστοσελίδας που τις εμπεριέχει ως εξωτερικούς συνδέσμους. Είναι φυσικό αυτή η *διαδικασία* να έχει τα πλεονεκτήματα και τα μειονεκτήματά της. Το μόνο σίγουρο είναι οτι, σε πολλές περιπτώσεις, ένας crawler μπορεί να φτάσει σε αρκετά μεγάλο “βάθος” όσον αφορά στην ιεραρχική δομή των υποσελίδων ενός δικτυακού τόπου. Κάποια από τα ζητήματα που αντιμετωπίζουν οι crawlers είναι η εύρεση διπλοεγγραφών κατά την διαδικασία ανίχνευσης, η ανάγκη επανάληψης της σάρωσης κάποιων σελίδων μετά από κάποιο χρονικό διάστημα και η επανακαταχώρηση της ανανεωμένης έκδοσής της, αν υπάρξει τέτοια.

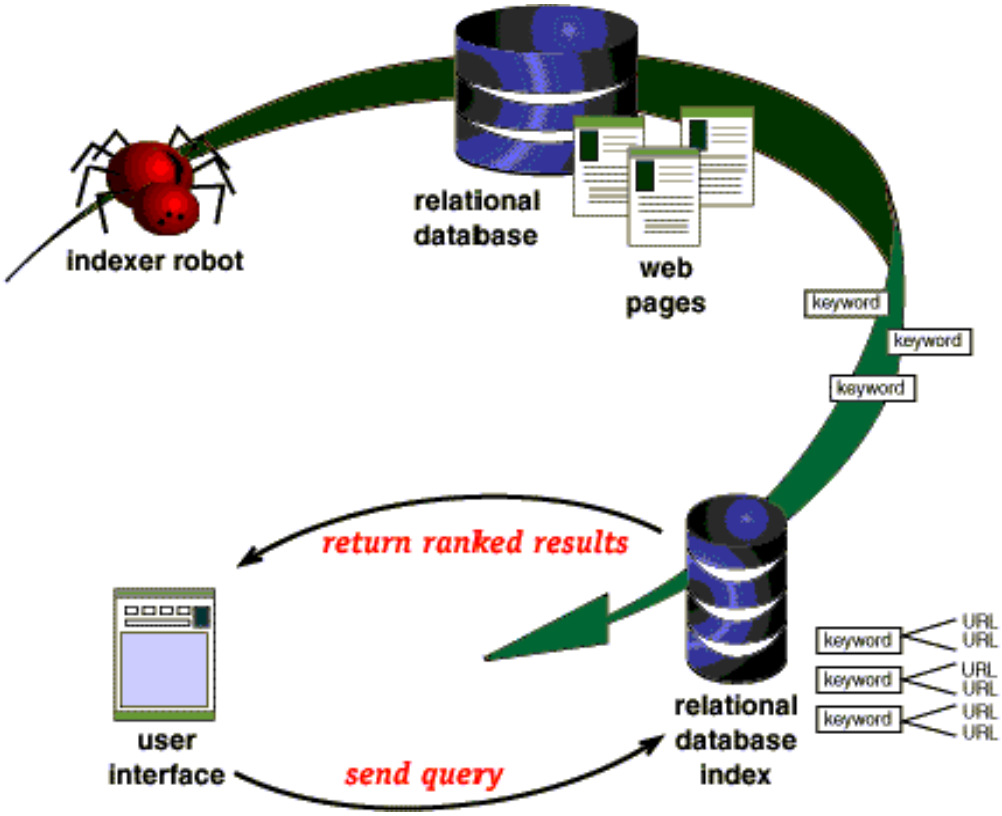
Indexer

Ο *indexer* είναι το λογισμικό που είναι υπεύθυνο για την καταχώρηση των όρων που περιέχονται σε μια ιστοσελίδα και τη δημιουργία του ευρετηρίου για τις αναζητήσεις που πρόκειται να γίνουν. Το ευρετήριο των όρων που έχει η κάθε μηχανή αναζήτησης στη ΒΔ της είναι αποτέλεσμα της διαδικασίας επεξεργασίας του περιεχόμενου των ιστοσελίδων από τον λογισμικό του indexer. Συνήθως, ο *indexer* δεν καταχωρεί όλους τους όρους που εμπεριέχονται, αλλά χρησιμοποιεί τους τίτλους, τις επικεφαλίδες και τα μετα-δεδομένα των ιστοσελίδων, σε μια προσπάθεια καλύτερης απόδοσης και εξοικονόμησης χώρου.

Query Processor

Ο *Query Processor* είναι το κομμάτι λογισμικού μιας μηχανής αναζήτησης που είναι υπεύθυνο για την επιστροφή αποτελεσμάτων στα αντίστοιχα ερωτήματα των χρηστών. Ο *Query Processor* περιλαμβάνει μια διεπαφή γραφικού περιβάλλοντος για την υποβολή των ερωτημάτων του κάθε χρήστη, έναν μηχανισμό αξιολόγησης συνάφειας των όρων του ερωτήματος και των εγγραφών της ΒΔ της μηχανής αναζήτησης και έναν μηχανισμό μορφοποίησης των αποτελεσμάτων που

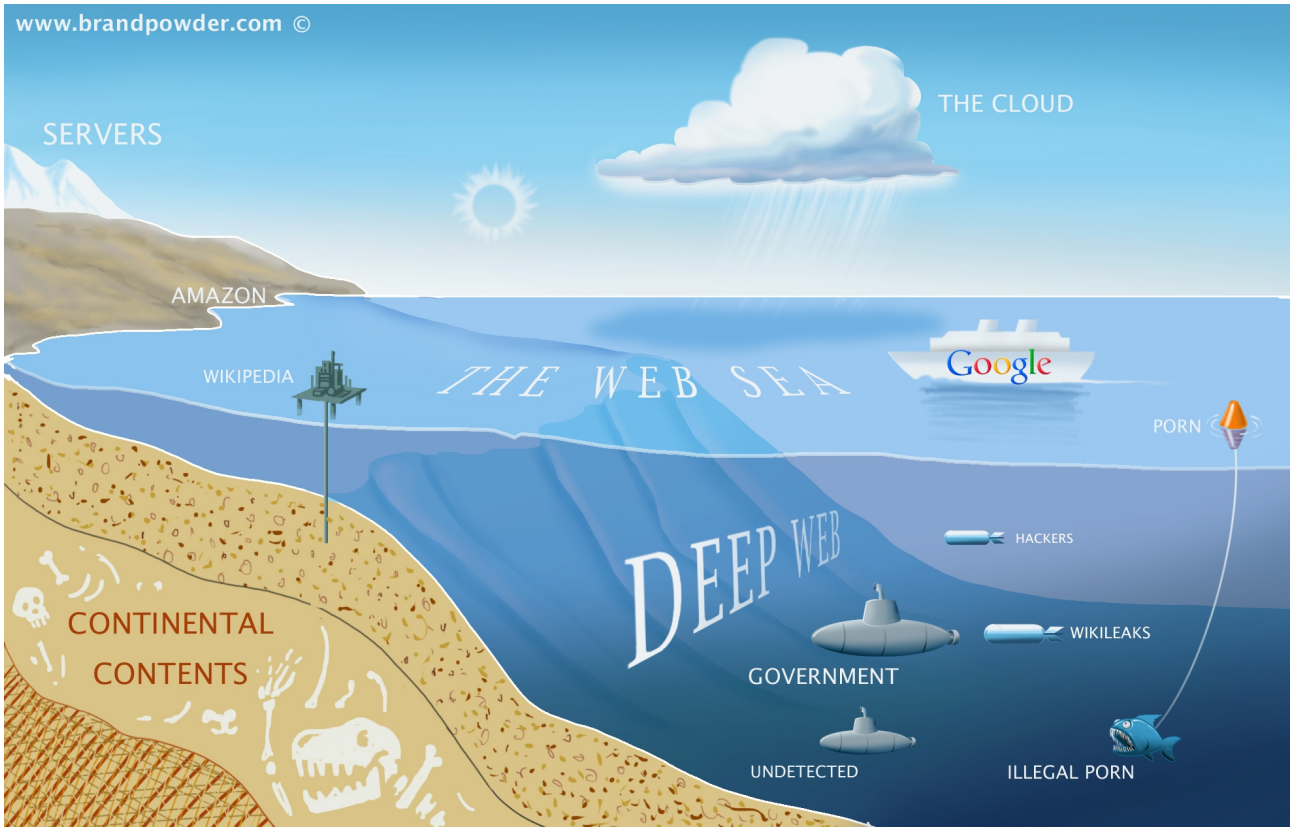
επιστρέφονται τελικά στον χρήστη. Τα αποτελέσματα που επιστρέφονται εξαρτώνται απο πολλές παραμέτρους, μερικές από τις οποίες μπορεί να ορίσει ο χρήστης πριν την υποβολή του ερωτήματός του (π.χ. Γλώσσα ή ημερομηνία αποτελεσμάτων).



Οι τεχνικές που χρησιμοποιούνται από τις μηχανές αναζήτησης δεν είναι γνωστές στο ευρύ κοινό και μόνο εικασίες μπορούν να γίνουν για το πώς δουλεύουν πραγματικά οι μηχανισμοί που περιγράφηκαν παραπάνω. Σε γενικές γραμμές, κάποιες μηχανές αναζήτησης, όπως η Google, χρησιμοποιούν αλγορίθμους ***Page Rank*** δίνοντας διαφορετική βαρύτητα σε κάθε ιστοσελίδα ανάλογα με τους υπάρχοντες όρους σε κάποιο ερώτημα. Από την άλλη, μια διαφορετική προσέγγιση είναι η στατιστική ανάλυση των περιεχομένων των ιστοσελίδων και η επιστροφή αποτελεσμάτων βάσει κάποιων μετρικών.

Deep Web

Στις μέρες μας, η έννοια της αναζήτησης της πληροφορίας του Web έχει κατά πολύ ταυτιστεί με την αναζήτηση πληροφορίας μέσω των μηχανών αναζήτησης ή μέσω των θεματικών καταλόγων. Αν και ο μέσος χρήστης είναι ευχαριστημένος από τις υπηρεσίες που προσφέρουν οι σημερινές μηχανές αναζήτησης και οι θεματικοί κάταλογοι, υπάρχουν πληροφορίες που δεν μπορούν να εντοπιστούν με αυτές τις τεχνικές. Οι πληροφορίες αυτές αποτελούν το λεγόμενο ***Deep Web*** και πολλές φορές το περιεχόμενό τους είναι πιο συναφές και σχετικό συγκριτικά με αυτό του ***Surface Web***. Οι ιστοσελίδες που φιλοξενούν τέτοιες πληροφορίες συνήθως αποτελούν επαγγελματικές λύσεις και τείνουν να έχουν πιο εξειδικεύμενο και σοβαρό χαρακτήρα. Η πρόσβαση σε αυτές τις ιστοσελίδες πολλές φορές μπορεί να χρειάζεται κάποιο είδος πιστοποίησης και συνδρομής του χρήστη. Όσον αφορά στο μέγεθος δεδομένων στα οποία έχει πρόσβαση το *Deep Web*, αυτό είναι 550 φορές μεγαλύτερο από αυτό του ***Surface Web***. Δεν είναι λίγες οι περιπτώσεις κατά τις οποίες βρέθηκαν στο *Deep Web* ιστοσελίδες παράνομου περιεχομένου όπως η κακοποίηση ανηλίκων, η πώληση ναρκωτικών κ.α.



Ορισμός του Deep Web: *Το σύνολο της πληροφορίας που είτε για τεχνικούς λόγους είτε λόγω συνειδητού αποκλεισμού από το κοινό, δεν μπορεί να εντοπιστεί με τις συνήθεις τεχνικές αναζήτησης όπως οι θεματικοί κατάλογοι και οι μηχανές αναζήτησης.*

Με βάση τα παραπάνω, προκύπτει το εύλογο ερώτημα σχετικά με το λόγο για τον οποίο οι πληροφορίες του *Deep Web* δεν είναι προσβάσιμες από τα συνήθη συστήματα αναζήτησης. Αυτό οφείλεται κυρίως στον τύπο των περιεχομένων που αυτό περιλαμβάνει. Παρακάτω αναγράφονται

οι σημαντικότεροι τύποι περιεχομένων του *Deep Web*.

- ***Δυναμικό περιεχόμενο:*** *Πολλές σύγχρονες ιστοσελίδες παράγουν το περιεχόμενό τους δυναμικά, ανάλογα με το ερώτημα που υποβάλλει ο χρήστης. Το ερώτημα αυτό συνήθως δημιουργείται μέσω της υποβολής μιας φόρμας μαζί με τις λέξεις - κλειδιά των διαφόρων πεδίων της φόρμας. Στην περίπτωση αυτή το πραγματικό περιεχόμενο της ιστοσελίδας δεν βρίσκεται κάπου στο Web αλλά σε αναζητήσιμες Βάσεις Δεδομένων. Είναι φανερό πως τα δεδομένα αυτά δεν είναι προσβάσιμα από τους crawler μιας μηχανής αναζήτησης, όπως επίσης είναι αδύνατο τα δεδομένα αυτά να περαστούν στους διάφορους θεματικούς καταλόγους.*
- ***Μη συνδεδεμένο περιεχόμενο:*** *Η ύπαρξη ιστοσελίδων που δεν έχουν υπερσυνδέσμους που να δείχνουν σε αυτές δεν είναι ένα συνηθισμένο φαινόμενο. Ωστόσο σε τέτοιες περιπτώσεις οι crawlers των μηχανών αναζήτησης αδυνατούν να τις επισκεπτούν και να τις καταχωρήσουν στις τοπικές τους ΒΔ. Ο μόνος τρόπος για να γίνουν οι ιστοσελίδες αυτές προσβάσιμες από τα συστήματα αναζήτησης, είναι η χειροκίνητη καταχώρηση στην αντίστοιχη υπηρεσία.*
- ***Περιεχόμενο περιορισμένης πρόσβασης:*** *Κάποιες από τις ιστοσελίδες του Web είτε παρέχουν πρόσβαση σε πιστοποιημένους χρήστες είτε αναγκάζουν το χρήση να συμφωνήσει σε κάποιους όρους πριν από την είσοδό του. Από την άλλη, κάποιες ιστοσελίδες χρεώνουν το περιεχόμενό τους, αναγκάζοντας τον πιθανό χρήστη να πληρώσει κάποιο χρηματικό ποσό. Προφανώς σε καμία περίπτωση τα περιεχόμενα αυτών των ιστοσελίδων δεν είναι προσβάσιμα από τα συνήθη συστήματα αναζήτησης.*
- ***Non-text περιεχόμενο:*** *Πολλές ιστοσελίδες εμπλουτίζουν το περιεχόμενό τους και προσθέτουν, πέρα από απλό κείμενο, περιεχόμενα που παρουσιάζονται με τη μορφή εικόνας, βίντεο ή ήχου. Σε τέτοιες περιπτώσεις οι crawlers αδυνατούν να αντιληφθούν το είδος του περιεχομένου και προφανώς δεν είναι σε θέση να κάνουν σωστή καταγραφή του συγκεκριμένου ιστότοπου. Εδώ θα πρέπει να σημειωθεί ότι πολλές από τις γνωστές μηχανές αναζήτησης (π.χ. Google) έχουν φθάσει σε σημείο να αναγνωρίσουν τα περιεχόμενα των εικόνων και - το πιο εντυπωσιακό - να ψάχνουν στο Web χρησιμοποιώντας μια εικόνα ως ερώτημα.*

Διαφορές Search Engines και Deep Web Harvest Engines

Με τον όρο *harvesting* εννοούμε την πρόσβαση και καταγραφή δεδομένων του *Deep Web* από εξειδικευμένα συστήματα αναζήτησης, τις *μηχανές συγκομιδής Deep Web*. Η διαφορά μεταξύ των παραδοσιακών μηχανών αναζήτησης, όπως η Google, και των μηχανών συγκομιδής Deep Web έγκειται στον τρόπο με τον οποίο δημιουργούν αυτές τα ευρετήριά τους. Οι μηχανές αναζήτησης καταγράφουν τους συνδέσμους των ιστοσελίδων σε ένα ευρετήριο, τις συνδέουν με τις λέξεις-κλειδιά που υπάρχουν στα περιεχόμενα των ιστοσελίδων και, τέλος, επιστρέφουν συναφή αποτελέσματα βάσει του ερωτήματος που τίθεται στο *Query processor.* Από την άλλη, οι *μηχανές συγκομιδής Deep Web* εξάγουν όλο το περιεχόμενο κειμένου από τις ιστοσελίδες για τις οποίες ενδιαφέρεται ο χρήστης, κάνουν ανάλυση των δεδομένων αυτών ανάλογα με τις ανάγκες του κάθε

χρήστη και είναι σε θέση να απαντήσουν σε πολύ εξειδικευμένα ερωτήματα.

Για να γίνουν καλύτερα κατανοητές οι διαφορές μεταξύ των παραδοσιακών *Search Engines* και των *Deep Web Harvest Engines* είναι καλύτερο να επικεντρωθούμε στο πρόβλημα που έρχονται να λύσουν οι μεν και οι δε.

Παλιότερα, στόχος των παραδοσιακών μηχανών αναζήτησης ήταν να κάνουν δυνατή την αναζήτηση των ιστοσελίδων και να παρέχουν όσο το δυνατόν πιο συναφή αποτελέσματα ανάλογα με τα ερωτήματα του χρήστη. Σήμερα, οι ίδιες μηχανές αναζήτησης αντιμετωπίζουν το πρόβλημα της υπερβολικής αύξησης του ρυθμού παραγωγής των δεδομένων και της ανάγκης αυτών να καταγραφούν. Η τακτική που εφαρμόζουν είναι η μερική αποθήκευση δεδομένων κατά την διαδικασία του *indexing*. Έτσι, οι μηχανές αναζήτησης αποθηκεύουν τους όρους που εμφανίζονται πιο συχνά, καθώς επίσης και τη θέση τους στο κείμενο. Επιπλέον, αποθηκεύουν πληροφορίες που υπάρχουν ως *meta-data*, όπως ο τίτλος της ιστοσελίδας, η περιγραφή και η διεύθυνσή της. Οι *μηχανές συγκομιδής Deep Web,* σε αντίθεση με τις κλασικές μηχανές αναζήτησης, αποθηκεύουν ολόκληρο το περιεχόμενο των ιστοσελίδων που καταγράφουν. Αυτός είναι ο ένας από τους λόγους για τον οποίο μπορούν να απαντήσουν σε πιο περίπλοκα ερωτήματα χρηστών. Ο άλλος λόγος είναι η στοχευμένη ανάλυση των δεδομένων αυτών ανάλογα με τις ανάγκες του χρήστη. Η προσέγγιση που ακολουθείται απο τις *μηχανές συγκομιδής Deep Web* είναι η παροχή απαντήσεων, πολλές φορές με οπτικοποίηση των αποτελεσμάτων για καλύτερη κατανόηση, προσανατολισμένη στις ανάγκες του χρήστη και όχι με τη χρήση ενός γενικού αλγορίθμου, όπως συμβαίνει στην περίπτωση των κλασικών μηχανών αναζήτησης. Τέλος, οι *μηχανές συγκομιδής Deep Web* είναι αναγκασμένες να κρατάνε πολλές εκδόσεις των ιστοσελίδων που καταγράφουν. Αν σκεφτούμε ότι το περιεχόμενο των ιστοσελίδων μπορεί να αλλάζει ανα κάποια χρονικά διαστήματα, θα πρέπει αυτές οι αλλαγές να καταγράφονται και να συνυπολογίζονται στην ανάλυση που ακολουθεί. Η καταγραφή των διαφορετικών εκδόσεων των ιστοσελίδων και η ανάγκη παραγωγής γνώσης από μεγάλα *data-sets* ανήκουν στο τομέα ***Big Data*** που είναι άμεσα συνδεδεμένος με τις *μηχανές συγκομιδής Deep Web.*

Παρακάτω δίνεται μια λίστα πηγών όπου υπάρχει ανάγκη χρήσης *Deep Web Harvester.*

- Ηλεκτρονικές εφημερίδες
- Νομικά
- Οικονομικά και διοίκηση επιχειρήσεων
- Υγειονομικά και φαρμακευτικά

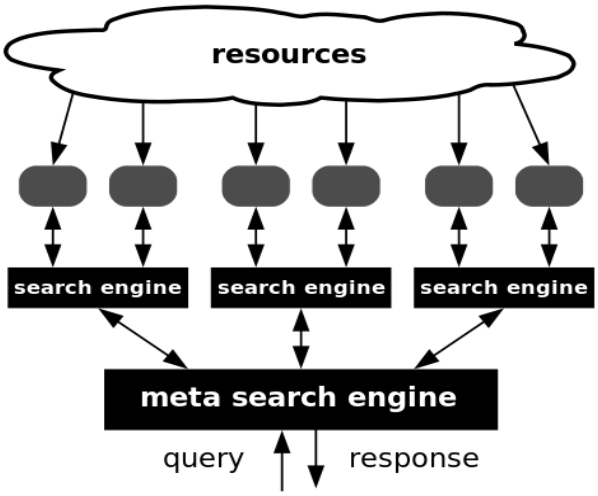
Meta-search

Οι μετα-μηχανές αναζήτησης είναι υπηρεσίες που επιτρέπουν την αναζήτηση σε πολλαπλές μηχανές αναζήτησης. Με την ενέργεια της αναζήτησης αποστέλλονται αιτήματα σε πολλαπλές πηγές και, συναθροίζοντας τα αποτελέσματα, καταλήγουμε τελικά στη συλλογή αποτελεσμάτων όχι μόνο από μια, αλλά από περισσότερες μηχανές αναζήτησης. Συνήθως, τα αποτελέσματα παρουσιάζονται ως μια λίστα των σχετικών εγγραφών και των πηγών από τις οποίες αυτές ελήφθησαν. Σε πολλές μετα-μηχανές δίνεται η δυνατότητα προσαρμογής των πηγών στις οποίες θα εκτελούνται τα ερωτήματα. Αν και υπάρχουν ακόμα κάποιοι που θεωρούν τη χρήση των μετα-μηχανών μη αναγκαία, είναι πλεόν αποδεκτό ότι με τη συνεχή αύξηση του όγκου δεδομένων του Web, τα ερωτήματα δεν θα πρέπει να εκτελούνται από μια μηχανή αναζήτησης αλλά από συνδυασμό αυτών.

Αρχιτεκτονική

Όπως προαναφέρθηκε, μια μετα-μηχανή αναζήτησης επιστρέφει μια συνάθροιση αποτελεσμάτων από πολλές μηχανές αναζήτησης. Πριν, όμως, τα τελικά αποτελέσματα φθάσουν στον χρήστη επιδέχονται κάποια επεξεργασία. Καθώς η συλλογή των αποτελεσμάτων από πολλές μηχανές αναζήτησης μπορεί να περιέχει διπλοαναφορές στην ίδια εγγραφή, τα διπλότυπα αυτά διαγράφονται και επαναταξινομούνται με τη χρήση κάποιου αλγόριθμου. Θα μπορούσε η ίδια εγγραφή-αποτέλεσμα να έχει υψηλή βαθμολογία σε μια μηχανή αναζήτησης και χαμηλή σε μια άλλη. Ως εκ τούτου, στην τελική παρουσίαση θα πρέπει να ληφθούν και οι δυο βαθμολογίες, ορίζοντας, τελικά, μια πιο ομογενοποιημένη βαθμολογία.

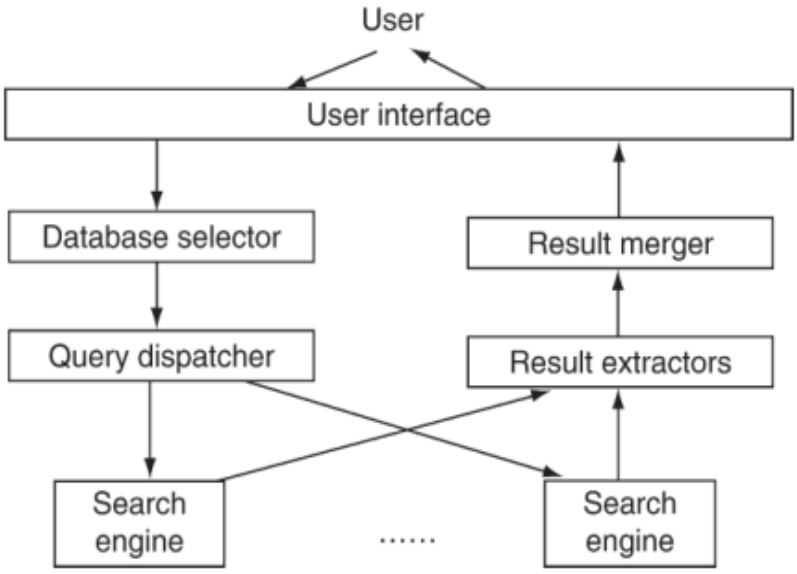
Συνήθως, μια μετα-μηχανή αναζήτησης χρησιμοποιεί τη λεγόμενη εικονική Βάση Δεδομένων (virtual database). Η δυσκολία της εικονικής ΒΔ βρίσκεται στη δημιουργία του καθολικού μοντέλου που θα περιγράφει ετερογενή συστήματα και τις συσχετίσεις των ΒΔ τους. Στα ερωτήματα που τίθενται σε μια μετα-μηχανή, η εικονική ΒΔ αποφασίζει το πώς και το πού θα μεταβιβάσει τα ερωτήματα και με τον τρόπο αυτό επιτυγχάνεται ο έλεγχος του φόρτου κίνησης δεδομένων. Στο παρακάτω σχήμα παρουσιάζεται η αρχιτεκτονική μιας μετα-μηχανής αναζήτησης και κάποια χαρακτηριστικά που συνήθως υλοποιούνται από μια τέτοια.



Λειτουργίες

Οι μετα-μηχανές αναζήτησης παρουσιάζουν κάποια κοινά χαρακτηριστικά, τα οποία σχετίζονται με τη λειτουργικότητά τους. Αν και μπορούμε να αναφερθούμε γενικά σε αυτά τα χαρακτηριστικά, η υλοποίηση αυτών μπορεί να διαφέρει από μετα-μηχανή σε μετα-μηχανή, γεγονός που κάνει βέβαια τη μια μετα-μηχανή καλύτερη από μια άλλη. Τα γενικά χαρακτηριστικά μια τέτοιας μηχανής είναι η *ταυτόχρονη αναζήτηση* σε πολλές μηχανές αναζήτησης, η ικανότητα *κλήσης ερωτημάτων* στις μηχανές αναζήτησης και η *αναγνώριση κατάλληλων αποτελεσμάτων* από τις πολλαπλές πηγές.

Στο παρακάτω σχήμα φαίνονται τα κύρια *components* μιας μετα-μηχανής όπου λαμβάνουν χώρα οι παρακάτω λειτουργίες.



Επιλογή μηχανής αναζήτησης

Ένα από τα κύρια χαρακτηριστικά μιας εξελιγμένης μετα-μηχανής αναζήτησης είναι η επιλογή συγκεκριμένων μηχανών αναζήτησης, ανάλογα με το ερώτημα που έχει τεθεί. Για να γίνει δυνατή αυτή η επιλογή, θα πρέπει αρχικά να γίνει μια περισυλλογή κάποιων *αντιπροσωπευτικών εγγραφών* από την κάθε μηχανή που χρησιμοποιείται ως πηγή. Αυτά τα αντιπροσωπευτικα έγγραφα αποθηκεύονται στο σύστημα της μετα-μηχανής και χρησιμοποιούνται για την αξιολόγηση της καταλληλότητας της κάθε μηχανής που αντιπροσωπεύουν, ανάλογα με το ερώτημα που τίθεται εκείνη τη στιγμή. Η επιλογή των αντιπροσωπευτικών εγγράφων, αλλά και η αξιολόγηση της καταλληλότητας ενός ερωτήματος με αυτά, πραγματοποιείται με τη χρήση διαφόρων τεχνικών. Τα αντιπροσωπευτικά έγγραφα μπορεί να οριστούν είτε αυτόματα είτε χειροκίνητα από κάποιον που γνωρίζει το πεδίο αναφοράς της μηχανής αναζήτησης που αντιπροσωπεύουν. Ο τρόπος αυτός, ωστόσο, βοηθάει μόνο σε μια γενική περιγραφή της εκάστοτε μηχανής αναζήτησης και δεν μπορεί

να είναι μια απόλυτη περιγραφή των περιεχομένων της. Μια πιο περίτεχνη μέθοδος είναι η καταγραφή στατιστικών αποτελεσμάτων για τους όρους που υπάρχουν στα περιεχόμενα μιας μηχανής αναζήτησης. Δεδομένου ότι είναι αδύνατον να γίνει μια στατιστική ανάλυση για όλους τους όρους μιας μηχανής αναζήτησης, χρησιμοποιείται ένα κατά προσέγγιση λεξιλόγιο που ανακτάται από τις μηχανές με ειδικά ερωτήματα, τα γνωστά ως ερωτήματα ανίχνευσης.

Από την άλλη, υπάρχουν τεχνικές δημιουργίας *αντιπροσωπευτικών εγγραφών* που χρησιμοποιούν δυναμικά την ίδια τη διαδικασία αναζήτησης. Στην περίπτωση αυτή, το σύστημα μαθαίνει από τα ερώτηματα που προηγήθηκαν και αξιολογεί το τρέχον ερώτημα δίνοντας βάρος σε κάθε όρο του. Μετά την αξιολόγηση του ερωτήματος, το βάρος του κάθε όρου προστίθεται ή αφαιρείται στην γενική βαθμολογία του όρου αυτού στη συγκεκριμένη μηχανή αναζήτησης, ανάλογα με το αν τα αποτελέσματα είναι τα ζητούμενα ή όχι. Με τη πάροδο του χρόνου, στην περίπτωση που ένας όρος έχει θετική τιμή, φαίνεται να έχει ανταποκριθεί καλά σε προγενέστερα ερωτήματα και θεωρείται αντίστοιχα συσχετισμένος με τη μηχανή αναζήτησης στην οποία παρίσταται. Το πλεονέκτημα των τεχνικών αυτού του είδους είναι η διαρκής δυναμική αξιολόγηση των αντιπροσωπευτικών εγγραφών

Συγχώνευση αποτελεσμάτων

Όπως προαναφέρθηκε, η συγχώνευση αποτελεσμάτων είναι μάλλον το απαραίτητο χαρακτηριστικό όλων των μετα-μηχανών αναζήτησης. Όλα τα αποτελέσματα των μηχανών αναζήτησης συγχωνεύονται σε μια ιεραρχημένη λίστα. Αρχικά οι μηχανές αναζήτησης επιστρέφαν μια αριθμητική τιμή για κάθε αποτέλεσμα. Οι τιμές αυτές κανονικοποιούνταν σε ένα διάστημα τιμών και ο αλγόριθμος συγχώνευσης σύγκρινε τα αποτελέσματα βάσει αυτών. Όταν δεν ήταν διαθέσιμες αυτές οι τιμές από τις μηχανές αναζήτησης, οι βαθμολογίες των αποτελεσμάτων μπορούσαν να υπολογιστούν με τεχνικές *voting-based* όπως η τεχνική *Borda Count*. Επίσης, πολλές φορές στις βαθμολογίες αυτές λαμβάνεται και η χρησιμότητα (*usefulness*) της κάθε μηχανής αναζήτησης ξεχωριστά, ένα μέγεθος που μπορεί να υπολογιστεί κατα τη διαδικασία *επιλογής της μηχανής αναζήτησης*. Για παράδειγμα μια βαθμολογία ενός αποτελέσματος μπορεί να πολλαπλασιαστεί με ένα συντελεστή που εκφράζει τη χρησιμότητα της μηχανής αναζήτησης στην οποία ανήκει και να αποκτήσει αντίστοιχη βαρύτητα.

Μια άλλη τεχνική συγχώνευσης αποτελεσμάτων βασίζεται στην ανάκτηση όλων των αποτελεσμάτων του τρέχοντος ερωτήματος στη ΒΔ της μετα-μηχανής αναζήτησης και στον υπολογισμό βαθμολογιών για κάθε αποτέλεσμα χρησιμοποιώντας μια κοινή συνάρτηση αξιολόγησης. Το πλεονέκτημα αυτής της μεθόδου είναι η ομοιόμορφη αξιολόγηση των αποτελεσμάτων, γεγονός που δε γεννάει αμφιβολίες ως προς την ορθότητα υλοποίησης της συνάρτησης αξιολόγησης. Το μεγαλύτερο μειονέκτημα αυτής της μεθόδου είναι η καθυστέρηση του χρόνου λήψης των αποτελεσμάτων και η ανάγκη για επιτόπου ανάλυση αυτών. Για το λόγο αυτό, πολλές από τις μηχανές αναζήτησης επιστρέφουν τα αποτελέσματά τους σε δυο στάδια. Κατα τη διάρκεια του πρώτου, εμφανίζονται οι τίτλοι και μια σύντομη περιγραφή του αντίστοιχου εγγράφου, του λεγόμενου *snippet*. Αν επιλεγεί μια εγγραφή με βάση τον τίτλο και το *snippet*, η μηχανή αναζήτησης επιστρέφει ολόκληρο το έγγραφο. Η διάσπαση της επιστροφής αποτελεσμάτων σε δυο φάσεις βοηθάει στην επιτόπου ανάλυσή τους, δεδομένου ότι μειώνεται κατά πολύ ο όγκος

δεδομένων κατα τη λήψη τους, καθώς και το πλήθος των όρων που θα αναλύθουν.

Τέλος, θα πρέπει να αναφερθούμε στην ιδιαίτερη αξιολόγηση αποτελεσμάτων, τα οποία παρουσιάζονται από περισσότερες μηχανές αναζήτησης ταυτόχρονα. Σε αυτή την περίπτωση, η εγκυρότητα αυτών των αποτελεσμάτων πιθανότατα να είναι μεγαλύτερη σε σχέση με τα υπόλοιπα. Η βαθμολογία των αποτελεσμάτων αυτών υπολογίζεται ως άθροισμα των μερικών βαθμολογιών που έχει λάβει απο την κάθε μηχανή αναζήτησης.

Αυτόματη σύνδεση μηχανής αναζήτησης

Η Γραφική Διασύνδεση Χρήστη (GUI) των περισσότερων μηχανών αναζήτησης είναι υλοποιημένη με τη χρήση HMTL φορμών και κάποιων *text-boxes*, μέσω των οποίων γίνεται η υποβολή των ερωτημάτων. Η ποσότητα και η εννοιολογική σημασία των *text-boxes* εξαρτάται από την ιδιαιτερότητα της αντίστοιχης υπηρεσίας. Για να γίνει η αυτόματη σύνδεση με μια μηχανή αναζήτησης θα πρέπει να είναι γνωστά αυτά τα πεδία, καθώς και ο τόπος και ο τρόπος με τον οποίο αυτά υποβάλλονται. Τέτοια στοιχεία είναι ο διακομιστής, ο οποίος δέχεται τα ερωτήματα, η μέθοδος με την οποία τα δέχεται και ο ακριβής τρόπος με τον οποίο οι όροι των ερωτημάτων περνάνε στο *network connection method*. Για παράδειγμα, ένα *network connection method* θα μπορούσε να είναι ένα HTTP request, που χρησιμοποιεί είτε POST είτε GET για να περάσουν οι οποιεσδήποτε παράμετροι. Όλες αύτες οι πληροφορίες θα πρέπει να είναι γνωστές και να τροποποιηθούν κατάλληλα στον *query dispatcher*, ούτως ώστε όταν γίνει η υποβολή του ερωτήματος από την μετα-μηχανή αναζήτησης, αυτό να υποβληθεί όπως ακριβώς θα υποβαλλόταν από την μηχανή αναζήτησης στην οποία αναφέρεται.

Αυτόματη εξαγωγή αποτελεσμάτων αναζήτησης

Τα αποτελέσματα που επιστρέφει μια μηχανή αναζήτησης μπορεί να επιστραφούν υπο τη μορφή μιας HTML σελίδας ή ενός XML εγγράφου ή ακόμα και ως μια δομή BLOB. Αυτό εξαρτάται από την υπηρεσία που προσφέρει η κάθε μηχανή αναζήτησης και από την τεχνολογία που χρησιμοποιείται για την υλοποίησή της. Σε κάθε περίπτωση, τα δεδομένα αυτά θα πρέπει να τροποποιηθούν και, με την κατάλληλη εξαγωγή της ζητούμενης πληροφορίας, να παρουσιαστούν ομοιόμορφα στο UI της μετα-μηχανής αναζήτησης. Είναι πιθανό σε πολλές από τις μηχανές αναζήτησης και ιδίως σε αυτές που επιστρέφουν μια HTML σελίδα, να επιστρέφονται και άχρηστες πληροφορίες που θα πρέπει να αγνοηθούν. Τέτοιες πληροφορίες μπορεί να είναι κάποιου είδους διαφημίσεις ή αλλου είδους σύνδεσμοι. Κάθε μηχανή αναζήτησης χρειάζεται ένα ξεχωριστό *component* για να αναλάβει αυτή την εργασία εξαιτίας της ιδιαιτερότητας με την οποία μπορεί να παρουσιάζονται τα αποτελέσματά της. Το *component* αυτό ονομάζεται *extraction wrapper* και η διαδικασία της εξαγωγής μπορεί να γίνει με διάφορες τεχνικές. Κάποιες από τις πιο γνωστές είναι :

- Fiva Tech *(A page-level web data extraction technique)*
- EXALG *(A template extraction in two stages)*
- ViPER *(Visual perception based Extraction of Records)*
- DeLa *(Data Extraction and Label Assignment for Web Databases)*

- DEPTA *(Data Extraction based on Partial Tree Alignment).*
- NET *(Nested data extraction using Tree matching and Visual cues)*
- IEPAD *(An information extraction system which applies pattern discovery techniques)*

Μειονεκτήματα

Θα ήταν συνετό να αναφέρουμε τις πιθανές αδυναμίες που μπορεί να παρουσιάσει μια μετα-μηχανή αναζήτησης. Κάποιες από αυτές παρουσιάζονται παρακάτω:

- Οι μετα-μηχανές αναζήτησης αδυνατούν να ανακτήσουν το σύνολο των αποτελεσμάτων των μηχανών αναζήτησης στις οποίες βασίζονται. Αν και αυτό μπορεί να λυθεί συνήθως με διάφορες τεχνικές, αποτελεί ένα φαινόμενο που αναμένεται να αντιμετωπίσει κανείς στις μετα-μηχανές αναζήτησης. Για παράδειγμα, μπορεί να λαμβάνονται μόνο τα δέκα πρώτα αποτελέσματα μιας μηχανής αναζήτησης, γεγονός που αμέσως μειώνει την εγκυρότητα των αποτελεσμάτων και τη χρήση της κάθε μηχανής.
- Κάθε μηχανή αναζήτησης μπορεί να έχει συγκεκριμένη σύνταξη για τη διατύπωση ερωτημάτων και να κάνει χρήση διαφορετικού τύπου τελεστών για τη δημιουργία κριτηρίων αναζήτησης. Συχνά, μια μετα-μηχανή αναζήτησης είτε δε λαμβάνει όλες τις δυνατές επιλογές που της επιτρέπει η κάθε μηχανή αναζήτησης είτε, για λόγους συνοχής, υποστηρίζει ένα μικρότερο σύνολο των επιλογών αυτών.
- Ένα άλλο σύνηθες φαινόμενο είναι η παρουσία ασυμβατότητας στα πεδία αναζήτησης που προσφέρουν οι διάφορες μηχανές αναζήτησης, καθιστώντας περίπλοκη τη διαδικασία αίτησης ερωτημάτων. Για παράδειγμα, μια μηχανή αναζήτησης μπορεί να δέχεται ερωτήματα με αναφορά στο πεδίο URL, ενώ μια άλλη να μην το υποστηρίζει. Ο τρόπος με τον οποίο θα υλοποιηθεί μια τέτοια ασυμβατότητα εξαρτάται από την τακτική προσέγγισης που ακολουθεί η εκάστοτε μετα-μηχανή αναζήτησης.
- Πολλές από τις μηχανές αναζήτησης προσφέρουν την υπηρεσία τους επί πληρωμή. Η πληρωμή μπορεί να εξαρτάται από χρονικούς περιορισμούς ή από το πλήθος των αποτελεσμάτων που θα λαμβάνει η μετα-μηχανή. Εναλλακτικά, ενδέχεται να χορηγείται απλώς μια άδεια χρήσεως της ζητούμενης υπηρεσίας.
- Η υπηρεσία της μετα-μηχανής αναζήτησης μπορεί να έχει μη προβλέψιμη συμπεριφορά, εφόσον τα αποτελέσματά της βασίζονται σε προηγούμενα αποτελέσματα των διαφόρων μηχανών αναζήτησης και εξαρτώνται από την παροχή των υπηρεσιών τους. Αν για κάποιο λόγο αλλάξει ο τρόπος σύνταξης των ερωτημάτων σε κάποια από αυτές τις μηχανές αναζήτησης, είναι φανερό ότι η αλλαγή αυτή θα επηρεάσει τα τελικά αποτελέσματα. Για το λόγο αυτό, μια μετα-μηχανή αναζήτησης δεν μπορεί να θεωρηθεί μια σταθερής αξίας υπηρεσία.

Μετα-μηχανές του μέλλοντος

Οι μελλοντικές μετα-μηχανές αναζήτησης αναμένεται να υποστηρίζουν δυο επιπλέον χαρακτηριστικά πέρα των προαναφερθέντων. Καταρχάς, καθίσταται αναγκαίο να έχουν τη δυνατότητα να προσαρμοστούν στη ραγδαία αύξηση κίνησης δεδομένων μέσω του Web και να μη βασίζονται σε ένα μικρό πλήθος από μηχανές αναζήτησης. Ωστόσο, κάτι τέτοιο δεν είναι πάντα εφικτό, λόγω που πεδίου δράσης της υπηρεσίας που επιθυμεί να προσφέρει μια μετα-μηχανή αναζήτησης. Το δεύτερο χαρακτηριστικό έχει να κάνει με την αυτοματοποίηση της προσαρμογής σε αλλαγές που συμβαίνουν στις μηχανές αναζήτησης από τις οποίες εξαρτάται η μετα-μηχανή αναζήτησης. Όπως προαναφέρθηκε, οι μηχανές αναζήτησης αλλάζουν, συχνά, τις παραμέτρους σύνδεσης ή τα ονόματα των πεδίων υποβολής των ερωτήματων. Τέτοιου είδους αλλαγές μπορούν να αχρηστεύσουν τις πηγές μιας μετα-μηχανής αναζήτησης και ουσιαστικά να ρίξουν την ποιότητα της υπηρεσίας που αυτή προσφέρει. Ο τρόπος με τον τον οποίο μπορεί κάποιος να παρακολουθεί αυτές τις αλλαγές, αλλά και να προσαρμόζει καταλλήλως το λογισμικό του σε αυτές σε σύντομο χρονικό διάστημα, είναι κάποια από τα ζητήματα που παρουσιάζουν μεγάλο ενδιαφέρον στην επιστημονική κοινότητα του συγκεκριμένου χώρου. Θα είχε, τέλος, ενδιαφέρον εάν αυτές οι τροποποιήσεις εκτελούνταν αυτόματα.

Federated Search & ezDL

Ορισμός: *Federated search ονομάζεται η τεχνολογία ανάκτησης πληροφορίας που αποσκοπεί στην ανάπτυξη μεθόδων αναζήτησης σε κατανεμημένα και πιθανότατα ετερογενή σύνολα δεδομένων, καθώς και στην επιστροφή των επιμέρους αποτελεσμάτων ως μια ενιαία συλλογή.*

Βασικά χαρακτηριστικά

Στα συστήματα αναζήτησης τύπου Federated Search, ο στόχος είναι η αναζήτηση σε ανεξάρτητες ετερογενείς συλλογές δεδομένων και η επιστροφή των αποτελεσμάτων ως μια μεμονομένη λίστα με έναν αποδοτικό τρόπο.

Το κεντρικό μέρος του συστήματος αναζήτησης τύπου Federated Search, ο *Query-federator* ή αλλιώς ο *Broker,* έχει ως κύρια λειτουργία του την λήψη των ερωτημάτων του χρήστη και την υποβολή τους στις κατάλληλες υπηρεσίες αναζήτησης. Οι υπηρεσίες που θεωρούνται κατάλληλες είναι αυτές που θα έχουν τις πιο σχετικές συλλογές δεδομένων με τα υποβληθέντα ερωτήματα. Για να γίνει δυνατή η επιλογή, ο *Query-federator* έχει αποθηκεύσει κάποιες πληροφορίες για τα δεδομένα της κάθε υπηρεσίας που έχει στη διάθεσή του. Βέβαια, κάτι τέτοιο είναι δυνατό με τις υπηρεσίες που έχουν συνεργάσιμο περιβάλλον *(cooperative environment)* και παρέχουν στατιστικά στοιχεία για τις συλλογές τους. Για τα μη συνεργάσιμα περιβάλλοντα *(uncooperative environment),* χρησιμοποιείται η τεχνική της συλλογής πληροφορίας μέσω υποβολής δειγμάτων-ερωτημάτων (probe queries). Οι συλλογές αυτές ονομάζονται *representation set* και βοηθούν στον υπολογισμό συνάφειας ενός ερωτήματος και των υπηρεσιών που είναι διαθέσιμες. Τα τελικά αποτελέσματα συχνά αποτελούνται από απαντήσεις που επιστρέφονται από πολλαπλές συλλογές.

Τα κύρια θέματα που απασχολούν τον τομέα του *Federation Search* είναι τα: ***Collection selection problem***, ***Collection representation problem*** και ***Result merging***.

Πλεονεκτήματα και μειονεκτήματα

Η αναζήτηση τύπου *Federated Search* δεν είναι πανάκεια. Πολλές φορές, οι κλασικές μηχανές αναζήτησης αποτελούν μια επαρκή λύση, ενώ άλλες φορές υπάρχει η ανάγκη μιας πιο εξειδικευμένης λύσης. Η επιλογή χρήσης του κάθε εργαλείου εξαρτάται από τις ανάγκες του χρήστη. Παρακάτω αναφέρονται κάποια πλεονεκτήματα και μειονεκτήματα του *Federated Search*.

- Ο χρήστης επωφελείται από την χρήση ενός εργαλείου *Federated Search* όσον αφορά στο χρόνο που χρειάζεται για τις αναζητήσεις.
- Η ανάκτηση δεδομένων από κάποιες πηγές είναι αδύνατη ακόμα και με την χρήση εργαλείου τύπου *Federated Search.*
- Η ύπαρξη ενός ενιαίου χώρου οργάνωσης των ερωτημάτων και των αποτελεσμάτων καθιστά εύκολη την υποβολή των ερωτημάτων και την επεξεργασία των αποτελεσμάτων. Συνήθως, τα εργαλεία τύπου *Federated Search* συνοδεύονται με μια σειρά από εργαλεία διαχείρισης των αποτελεσμάτων. Τέτοια μπορεί να είναι εργαλεία ομαδοποίησης, περίπλοκης αναζήτησης και οπτικοποίησης των τελικών αποτελεσμάτων.

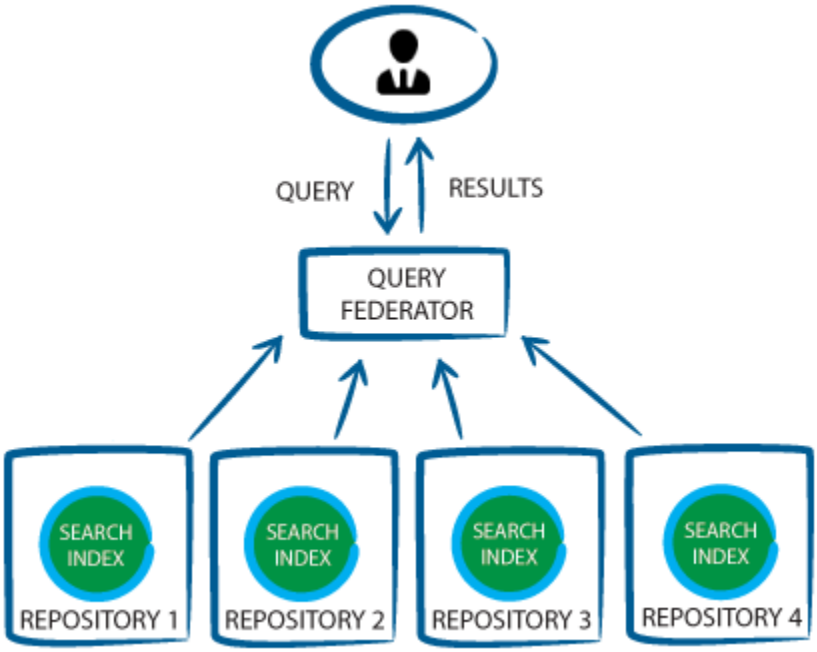
- Ωστόσο, κατά την διαδικασία υποβολής των ερωτημάτων υπάρχει περίπτωση να μην γίνονται επιτρεπτές όλες οι συντακτικές δυνατότες. Αυτό συμβαίνει λόγω της διαφορετικότητας των υπηρεσιών στις οποίες απευθύνεται το κάθε ερώτημα.
- Τα εργαλεία τύπου *Federated Search* χαρακτηρίζονται περισσότερο ως υπηρεσία και όχι ως λογισμικό. Κοινώς, η εξάρτηση των εργαλείων τύπου *Federated Search* από άλλες μηχανές αναζήτησης και από άλλες υπηρεσίες μειώνει, συχνά, την εγγύτητα των αποτελεσμάτων και την ευρωστία του εργαλείου.

Αρχιτεκτονική

Υπάρχουν δύο προσεγγίσεις όσον αφορά στην αρχιτεκτονική του *Federated search,* η αρχιτεκτονική με *query-time merging* και η αρχιτεκτονική με *index-time merging*.

QUERY-TIME MERGING

Όπως φαίνεται στο σχήμα παρακάτω, ο *Query-federator* λαμβάνει τα ερωτήματα και τα αποστέλλει σε όλες τις μηχανές αναζήτησης με τις οποίες είναι συνδεδεμένος. Έπειτα λαμβάνει τις απαντήσεις, τις ενοποιεί και τις εμφανίζει στον χρήστη.



Πλεονεκτήματα

Το βασικό πλεονέκτημα αυτής της αρχιτεκτονικής είναι η εύκολη υλοποίησή της, αφού, όπως θα δούμε παρακάτω, το επιπλέον indexing των δεδομένων δεν είναι απαραίτητο. Το query federation system απλώς αποστέλλει τα ερωτήματα και συλλέγει τα δεδομένα.

Σε κάποιες περιπτώσεις, η αρχιτεκτονική αυτή είναι η μόνη προσέγγιση. Τέτοιες είναι:

- όταν υπάρχει η ανάγκη πρόσβασης σε μεγάλης κλίμακας περιεχόμενο του Web, πράγμα που γίνεται μέσω κάποιας μηχανής αναζήτησης όπως η Google.
- όταν υπάρχει η ανάγκη πρόσβασης σε ιδιόκτητα σύνολα δεδομένων, τα οποία δε διατίθενται στο κοινό χωρίς χρέωση.

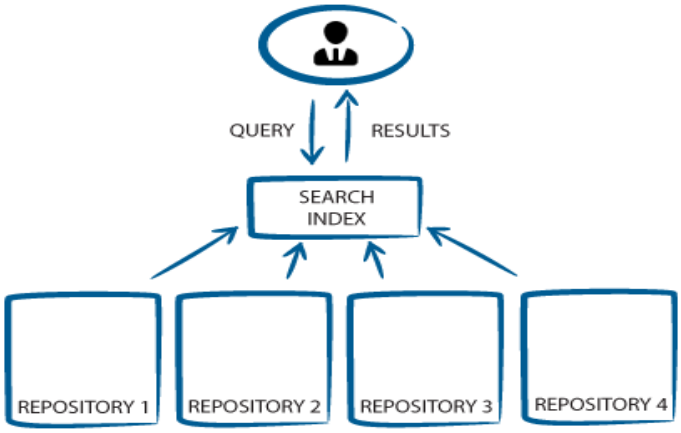
Μειονεκτήματα

Τα μειονεκτήματα που παρουσιάζει αυτό το μοντέλο έχουν σχέση με την αποδοτικότητα και την λειτουργικότητα του συστήματος αναζήτησης. Τέτοια είναι:

- το σύστημα συνήθως πρέπει να αναμένει την επιστροφή όλων των επιμέρους αποτελεσμάτων των μηχανών αναζήτησης. Συνεπώς, τίθενται ζητήματα απόδοσης όταν η ύπαρξη μιας αργής μηχανής αναζήτησης μπορεί να καθυστερήσει την όλη διαδικασία.
- κατα τη διαδικασία συγχώνευσης των αποτελεσμάτων, οι επιμέρους βαθμολογήσεις των αποτελεσμάτων πιθανότατα δεν χρησιμοποίουν την ίδια μέθοδο. Στην περίπτωση αυτή προτιμάται η απλή προβολή των αποτελεσμάτων σε ξεχωριστές λίστες ή η εφαρμογή κάποιας συγχώνευσης αποτελεσμάτων και ομαδοποίησή τους βάσει κάποιου πεδίου όπως η π.χ. ημερομηνια.
- όσον αφορά στο ερώτημα που τίθεται και στην πολυπλοκότητά του, αυτά πρέπει να προσαρμοστούν στις κοινές απαιτήσεις όλων των επιμέρους μηχανών αναζήτησης. Με τον τρόπο αυτό, περιορίζεται η δυνατότητα υποβολής εξελιγμένων ερωτημάτων. Βέβαια, μια λύση είναι η προσθήκη των λεγόμενων *Query parsers* στην αρχιτεκτονική, οι οποίοι θα εξειδικεύουν τα ερωτήματα για την κάθε μηχανή αναζήτησης.

INDEX-TIME MERGING

Μια πιο επαγγελματική λύση είναι αυτή που φαίνεται στο παρακάτω σχήμα. Σε αυτή την περίπτωση απαιτείται η ύπαρξη ενός κεντρικού *index* των δεδομένων που θα προσφέρονται για αναζήτηση.



Πλεονεκτήματα

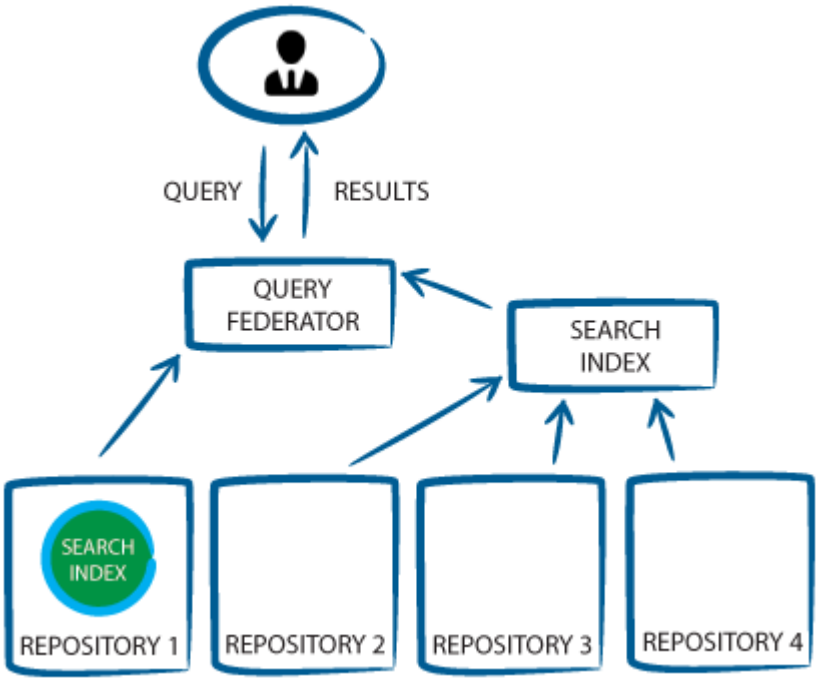
Με την αρχιτεκτονική που παρουσιάζεται στο παραπάνω σχήμα γίνεται δυνατή η χρήση εξειδικευμένων αλγορίθμων βαθμολόγησης των αποτελεσμάτων, προσαρμοσμένων στις ανάγκες της υπηρεσίας που θέλει να προσφέρει το σύστημα αναζήτησης. Σαφώς, σε αυτήν την περίπτωη υπάρχει καλύτερη συνάφεια μεταξύ αποτελεσμάτων και ερωτημάτων του χρήστη.

Μειονεκτήματα

Η δημιουργία του *indexing* του συστήματος αναζήτησης δεν είναι εύκολη διαδικασία, ειδικά όταν οι πηγές των δεδομένων είναι απομακρυσμένες ΒΔ με δυναμικό περιεχόμενο ή είναι ΒΔ που είναι προσβάσιμες μόνο με άδεια.

HYBRID FEDERATED SEARCH

Μια λύση που προτιμάται είναι η υβριδική αρχιτεκτονική που φαίνεται στο παρακάτω σχήμα. Στο μοντέλο αυτό γίνεται χρήση του *Query-federator* και του *indexer* ανάλογα με τις δυνατότητες των πηγών. Η δυσκολία σε αυτήν την αρχιτεκτονική βρίσκεται στην πολυπλοκότητα της υλοποίησης της επικοινωνίας μεταξύ των *Query-federator* και *indexer.*



Το ezDL – Ένα διαδραστικό σύστημα αναζήτησης

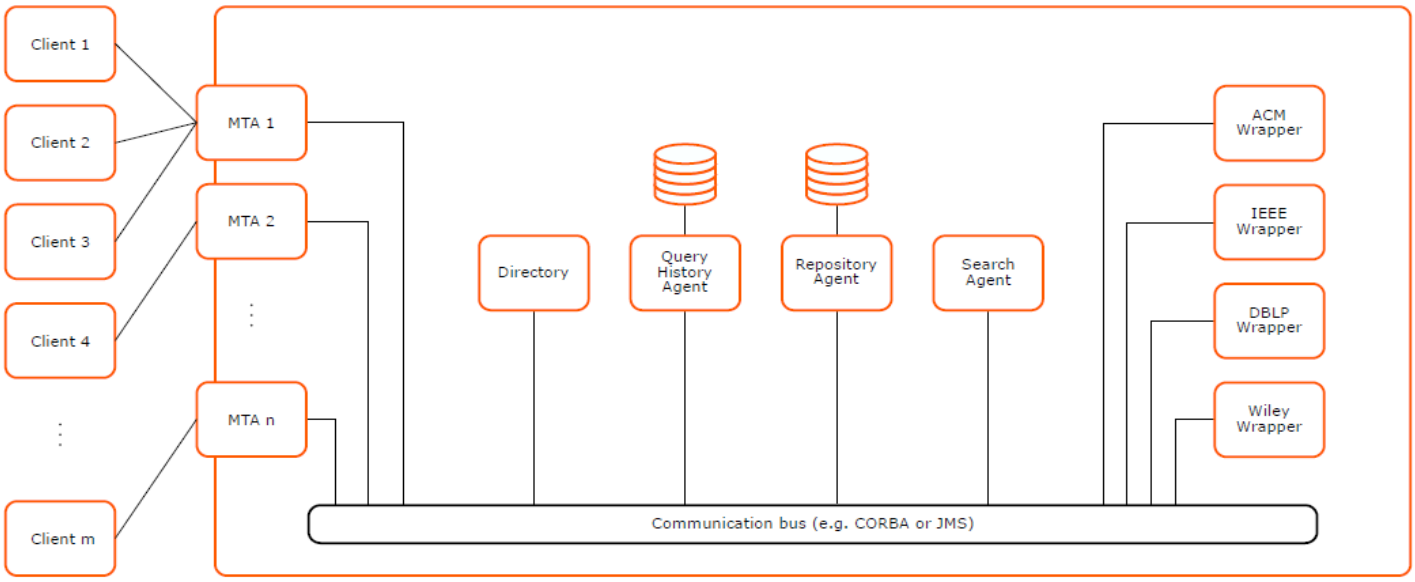
Το ***ezDL*** είναι ένα διαδραστικό σύστημα αναζήτησης ανοιχτού λογισμικού. Επηρεασμένο από το Daffodil, ένα project που είχε στόχο τη στρατηγική υποστήριξη κατά τη διάρκεια της διαδικασίας αναζήτησης σε ψηφιακά έγγραφα, το ezDL αποτελεί ένα framework με τα εξής βασικά χαρακτηριστικά:

- *αποτελεί ένα διαδραστικό εργαλείο αναζήτησης σε ετερογενείς συλλογές ψηφιακών εγγράφων. Ενσωματώνει τη φιλοσοφία της μετα-μηχανής αναζήτησης, συναθροίζει τα αποτελέσματα των επιμέρους πηγών και παρέχει μια σειρά από εργαλεία για την προβολή τους.*
- *αποτελεί μια πλατφόρμα ανάπτυξης διαδραστικών συστημάτων αναζήτησης με δυνατότητες προσαρμογής στις διάφορες ανάγκες της υπηρεσίας. Βασικά χαρακτηριστικά της αρχιτεκτονικής της πλατφόρμας του ezDL είναι η επεκτασιμότητα (extensibility) και η επαναχρησιμοποίηση (reusability) των βασικών λειτουργιών της.*
- *αποτελεί ένα σύστημα αξιολόγησης των αποτελεσμάτων της αναζήτησης με τη χρήση διαφόρων τεχνικών.*

Σήμερα, υπάρχουν πολλά επαγγελματικά εργαλεία που υποστηρίζουν κάποια από τα παραπάνω χαρακτηριστικά. Αυτό που κάνει το ***ezDL*** να ξεχωρίζει, είναι η ενοποίηση αυτών των χαρακτηριστικών σε ένα πολυεργαλείο.

Αρχιτεκτονική

Η αρχιτεκτονική του συστήματος ezDL βασίζεται στην αρχή της ελαχιστοποίησης των εξαρτήσεων μεταξύ των διαφόρων συστατικών του. Αυτό συμβάλλει στη σταθερότητα του όλου συστήματος, καθώς και στην ικανότητα ανεξάρτητης λειτουργίας των επιμέρους συστατικών. Όπως φαίνεται στο παρακάτω σχήμα, ένας πρώτος διαχωρισμός γίνεται μεταξύ των clients και του back-end του συστήματος Αυτό δίνει τη δυνατότητα της σύνδεσης και εξυπηρέτησης πολλών χρηστών ταυτόχρονα. Το ίδιο το back-end λειτουργεί ως ένα σύνολο οντοτήτων, των λεγόμενων “agents”, οι οποίοι επικοινωνούν μεταξύ τους και με τους clients παρέχοντας διαφορετικές λειτουργίες. Η ευκολία με την οποία προστίθεται ένας νέος agent συνεπάγεται αμέσως την πρόσθεση μιας νέας λειτουργίας στο σύστημα και φανερώνει την ικανότητα επεκτασιμότητας του ezDL. Τέλος, ο client αποτελείται από το κύριο πρόγραμμα και από ένα σύνολο ανεξάρτητων συστατικών, τα λεγόμενα “tools”. Τα εργαλεία αυτά επικοινωνούν μεταξύ τους με τη χρήση μηνυμάτων και δίνουν τη δυνατότητα πρόσθεσης επιπλέον λειτουργιών στο front-end κομμάτι του ezDL. Παρακάτω, θα γίνει αναφορά σε επιπλέον λεπτομέρειες για το κομμάτι του back-end και του front-end.



Το Back-end

Το back-end αποτελεί τον πυρήνα του συστήματος ezDL και περιλαμβάνει τα κύρια συστατικά που παρέχουν τις διάφορες υπηρεσίες του. Τα συστατικά αυτά, οπώς προαναφέρθηκε, ονομάζονται agents, ορολογία που προκύπτει από το γεγονός ότι ακολουθούν την *agent-based architecture.* Οι agents επικοινωνούν χρησιμοποιώντας ένα κοινό δίαυλο επικοινωνίας αποστέλλοντας μηνύματα μεταξύ τους.

Στο κομμάτι του back-end γίνεται η επικοινωνία με τη ΒΔ, καθώς και η υλοποίηση των λειτουργιών που σχετίζονται με αυτήν. Κάποιες από αυτές είναι η καταχώρηση στοιχείων χρηστών όπως το *username* και το *password*, καθώς επίσης και το ιστορικό αναζητήσεων του κάθε χρήστη. Στο σημείο αυτό, σημαντικός είναι ο ρόλος του *agent Directory,* ο οποίος είναι υπεύθυνος για την εκκίνηση του όλου συστήματος, αλλά και για την καταγραφή των ενεργών agents και των *resources* του συστήματος ανα πάσα στιγμή. Ο μηχανισμός της μετα-μηχανής αναζήτησης, η πιστοποίηση των χρηστών, η βάση γνώσης των συλλεγόμενων εγγράφων και οι wrappers είναι μέρη κάποιων agents που βρίσκονται στο back-end του ezDL.

Δεδομένου ότι κάθε λειτουργικότητα που παρέχεται από το ezDL βρίσκει την υλοποίησή της σε κάποιο agent, ο τερματισμός ενός agent δε συνεπάγεται τον τερματισμό του όλου συστήματος αλλά της συγκεκριμένης μόνο λειτουργικότητας. Αυτό κάνει το σύστημα πιο ευέλικτο και αυξάνει την ευρωστία *(robustness)* του. Σε πολλές περιπτώσεις, η δυνατότητα χρήσης πολλαπλών agents επιτρέπει τη διαχείριση του φόρτου εργασίας του συστήματος *(load balancing)* ή μπορεί να λειτουργήσει ως μηχανισμός ασφαλείας *(fail-safe mechanism)*. Ωστόσο, σε περιπτώσεις όπως αυτή του agent Directory, η οποιαδήποτε δυσλειτουργία μπορεί να προκαλέσει τον τερματισμό ολόκληρου του συστήματος.

Στα αριστερά του παραπάνω σχήματος, φαίνονται οι *MTA agents* που λειτουργούν ως σύνδεσμοι μεταξύ των clients και του back-end. Κάθε MTA *(Message Transfer Agent)* είναι υπευθυνος για την πιστοποίηση των χρηστών και για τη μεταφορά μηνυμάτων από και προς το back-end για το client που κάνει τη σύνδεση. Για παράδειγμα, αν ένας client αιτηθεί μια αναζήτηση βάσει κάποιου

ερωτήματος, το ερώτημα μεταδίδεται από τον MTA agent στον Search Agent μέσω ειδικού μηνύματος. Με τον τρόπο αυτό γίνεται σαφής διαχωρισμός των διαφόρων υπηρεσιών που προσφέρει το ezDL, χωρίς ο client να γνωρίζει κάτι για αυτές. Επίσης, δίνεται η δυνατότητα εκκίνησης πολλαπλών Search Agents, ανάλογα με το φόρτο του συστήματος, χωρίς κάποια επιπλέον ενέργεια του χρήστη. Αν και στην παρούσα υλοποίηση οι MTA agents χρησιμοποιούν ένα binary πρωτόκολλο μέσω μιας TCP σύνδεσης για την επικοινωνία, η σχεδίαση της αρχιτεκτονικής, όπως την περιγράψαμε πιο πάνω, επιτρέπει την μελλοντική ύπαρξη διαφορετικών υλοποιήσεων MTA agents. Για παράδειγμα, θα μπορούσαν να χρησιμοποιούν το πρωτόκολλο SOAP για την επικοινωνία μεταξύ του back-end και των clients.

Ο *Directory agent* είναι ένας agent ειδικού σκοπού και, ως εκ τούτου, το σύστημα διαθέτει μόνο ένα στιγμιότυπό του. Αποστολή του *Directory* είναι, μεταξύ άλλων, η καταγραφή οποιουδήποτε άλλου agent και των υπηρεσιών του, ώστε να είναι δυνατή η δρομολόγηση του κάθε αιτήματος στην κατάλληλη υπηρεσία.

Στη δεξιά πλευρά του σχήματος παρουσιάζονται οι agents, οι οποίοι επιτυγχάνουν την τελική σύνδεση με τις τοπικές ή απομακρυσμένες υπηρεσίες (π.χ. υπηρεσίες ανάκτησης δεδομένων ψηφιακών εγγράφων). Οι agents αυτοί μεταφράζουν το ερώτημα που τίθεται από τον client στην κατάλληλη μορφή που χρειάζεται η εκάστοτε υπηρεσία, λαμβάνουν τα αποτελέσματα του ερωτήματος αυτού και τα επιστρέφουν στην κατάλληλη μορφή που αναμένεται από το σύστημα του ezDL.

Για να καταστεί σαφής η ροή εργασίας *(work-flow)* κατά την εκτέλεση ενός ερωτήματος, ακολουθεί αναφορά στα βήματα που εκτελούνται κατά τη διαδικασία.

1. Ο client αποστέλλει το αίτημα ερωτήματος στον MTA agent, με τον οποίο είναι συνδεδεμένος. Το αίτημα περιέχει το ερώτημα και μια λίστα από υπηρεσίες, στις οποίες αναμένεται αυτό να εκτελεστεί.

2. Ο MTA agent προωθεί το αίτημα στον Search agent. Αυτό το σημείο είναι κατάλληλο για να γίνει διαχείριση του φόρτου εργασίας του συστήματος. Ο διαμοιρασμός του φόρτου μπορεί να γίνει με την χρήση παραπάνω στιγμιότυπων *Search agents*.

3. Ο *Search agent* ζητάει από το *Directory agent* τα ονόματα των *agents* που αποτελούν τους συνδετικούς κρίκους με την κατάλληλη υπηρεσία. Μετά την παραλαβή της λίστας των *agents*, ο *Search agent* στέλνει το ερώτημα σε κάθε έναν από αυτούς.

4. Ο κάθε agent που παραλαμβάνει το ερώτημα, το μετατρέπει σε ερώτημα συμβατό με την υπηρεσία, για την οποία είναι υπεύθυνος και το αποστέλλει. Μετά την παραλαβή των αποτελεσμάτων από την υπηρεσία, δημιουργεί μια συλλογή με αυτά και τα στέλνει πίσω στον Search agent.

5. Ο Search agent συλλέγει όλα τα σύνολα αποτελεσμάτων από όλους τους agents, τις συγχωνεύει σε μια ενιαία συλλογή, διαγράφει τις διπλοεγγραφές και τις αξιολογεί δίνοντάς τους κάποιο βαθμό. Η αξιολόγηση των αποτελεσμάτων γίνεται είτε με το αρχικό *RSV* σύστημα είτε με τη λειτουργία που προσφέρει το *Lucene.* Ο *Search agent* προωθεί, επίσης, τη συλλογή των εγγράφων στον *Repository agent*.

6. Η τελική συλλογή αποστέλλεται στον MTA agent που αιτήθηκε την αναζήτηση και αυτός με τη σειρά του την στέλνει στον client που είναι συνδεδεμένος.

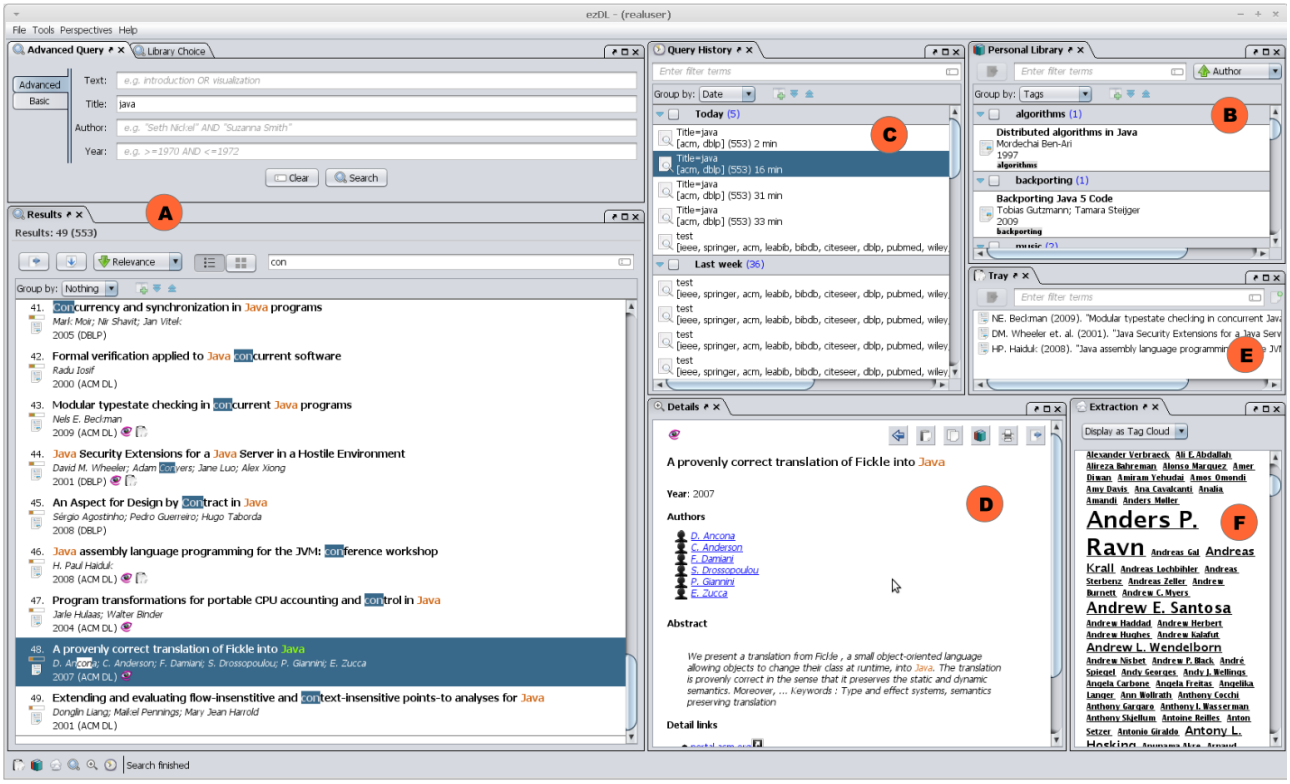
To Front-end

Υπάρχουν διαφορετικές υλοποιήσεις του front-end για το ***ezDL***. Παρακάτω θα γίνει μια σύντομη περιγραφή της desktop εφαρμογής, δεδομένου ότι, αυτή τη στιγμή, αυτή αποτελεί τον πιο συχνό και ευρείας χρήσεως client του ezDL.

Εργαλεία και όψεις

Ένα εργαλείο αποτελείται από ένα σύνολο αλληλοεξαρτώμενων λειτουργιών. Με τη χρήση πολλών διαφορετικών *views*, τα εργαλεία μπορούν να τοποθετηθούν με ποικίλους τρόπους και να προσαρμοστούν στις εκάστοτε ανάγκες του χρήστη. Η παραμετροποίηση των *views* σε ένα συγκεκριμένο *layout* ονομάζεται *perspective* και αποθηκεύεται για τον κάθε client.

Στην παρακάτω εικόνα παρουσιάζονται το γραφικό περιβάλλον και τα κύρια εργαλεία του client.



Ακολουθεί μια σύντομη περιγραφή των εργαλείων που απεικονίζονται παραπάνω

- Το εργαλείο (A) είναι το ***Search Tool*** και αποτελείται από τις διαφορετικές φόρμες υποβολής των ερωτημάτων, το view της λίστας με τις διαθέσιμες υπηρεσίες αναζήτησης και το view για την προβολή των τελικών αποτελεσμάτων της αναζήτησης. Με το εργαλείο αυτό, δίνεται η δυνατότητα ταξινόμησης και ομαδοποίησης των αποτελεσμάτων βάσει των πεδίων τους. Επίσης, ο χρήστης μπορεί να αναζητήσει ένα συγκεκριμένο έγγραφο φιλτράροντας όλα τα αποτελέσματα με λέξεις - κλειδιά και να εξάγει κάποιο αποτέλεσμα ως

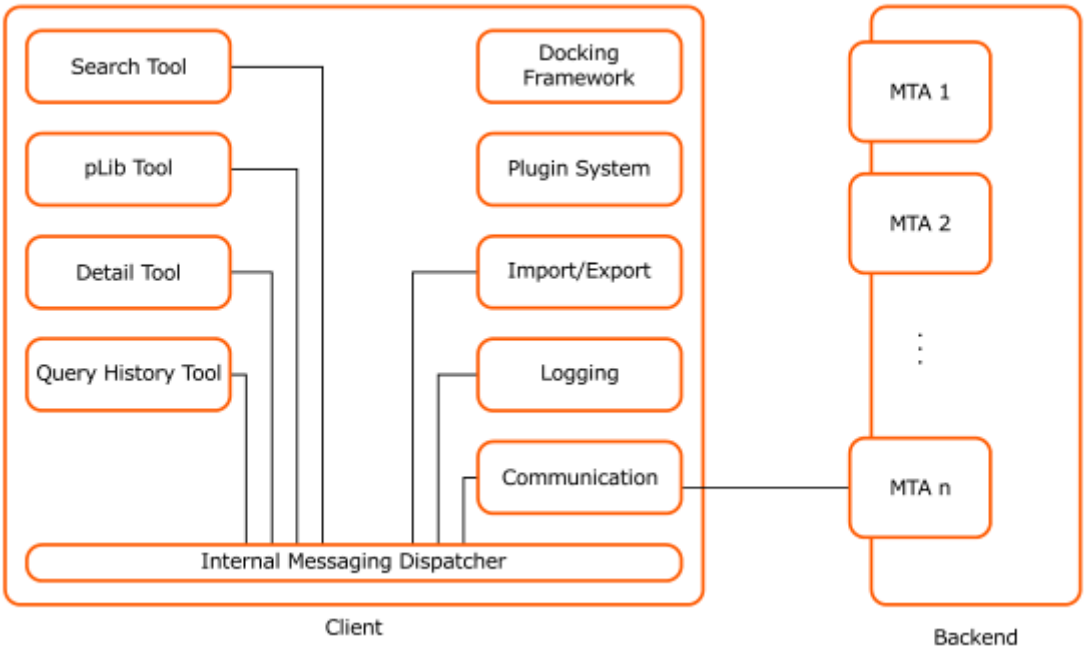
αρχείο κειμένου.

- Το εργαλείο (F) είναι το ***Extraction Tool***, το οποίο χρησιμοποιείται για την οπτικοποίηση των πιο συχνά εμφανιζόμενων όρων στα αποτελέσματα που υπάρχουν στο Results view. Η οπτικοποίηση μπορεί να γίνει με τη χρήση μιας λίστας ή/και ενός ραβδογράμματος και να παρουσιαστεί με την μορφή ενός σύννεφου όρων *(cloud term)*.
- Το εργαλείο (B) είναι το ***Personal Library***, το οποίο επιτρέπει την αποθήκευση των ερωτημάτων και των εγγράφων από τα αποτελέσματα που έχει υποβάλει ο χρήστης. Ο χρήστης μπορεί να δημιουργήσει μια προσωπική συλλογή από αναφορές και να οργανώσει τη δική του βιβλιοθήκη από έγγραφα, γνωρίζοντας την ημερομηνία, το ερώτημα και άλλες σχετικές πληροφορίες με αυτά. Επιπλέον, έχει τη δυνατότητα να προσθέσει δικές του ταμπέλες που θα τον βοηθήσουν στην οργάνωση της συλλογής του.
- Το εργαλείο (C) είναι το ***Search History*** και χρησιμοποιείται για την αποθήκευση των προγενέστερων ερωτημάτων που έχει υποβάλει ο χρήστης. Αποτελείται από μια λίστα, στην οποία αναγράφονται οι όροι του ερωτήματος, η ημερομηνία κατά την οποία τέθηκε το ερώτημα και οι υπηρεσίες αναζήτησης που χρησιμοποιήθηκαν. Επίσης, φιλτράροντας με λέξεις - κλειδιά, δίνεται η δυνατότητα ταξινόμησης των ερωτημάτων βάσει της ημερομηνίας στην οποία υποβλήθηκαν, καθώς και η δυνατότητα εύρεσης κάποιου συγκεκριμένου ερωτήματος.
- Το εργαλείο (D) είναι το ***Detail View*** και είναι αυτό στο οποίο προβάλλονται οι επιπλέον πληροφορίες ενός αποτελέσματος όταν αυτές ζητηθούν. Τέτοιου είδους πληροφορίες μπορεί να σχετίζονται με μια σύντομη περιγραφή, με κάποια μετα-δεδομένα του επιλεγόμενου εγγράφου και με έναν υπερσύνδεσμο που δείχνει στην αρχική πηγή του εγγράφου.
- Το εργαλείο (E) είναι το ***Tray*** και χρησιμοποιείται για την προσωρινή αποθήκευση κάποιων εγγράφων κατα τη διάρκεια ενός *session* του χρήστη.

Επικοινωνία με Front και Back-end

Η επικοινωνία μεταξύ των tools πραγματοποιείται με τη χρήση μηνυμάτων μέσω μιας εσωτερικής υποδομής επικοινωνίας. Ο ***Internal Messaging Dispatcher*** αποτελεί το συνδετικό κρίκο μεταξύ όλων των εργαλείων του client. Όταν ένα εργαλείο στέλνει ένα μήνυμα, αυτό φτάνει στον Dispatcher και αναμεταδίδεται σε όλα τα εργαλεία. Το μόνο που απαιτείται είναι να έχει προηγηθεί η δήλωση του ενδιαφέροντος για το συγκεκριμένο μήνυμα στον εσωτερικό κώδικα του tool. Για να επικοινωνήσει κάποιο tool με το Back-end αρκεί να αποστείλει μήνυμα με μια συγκεκριμένη μορφή που έχει δηλωθεί στο ezDL. Το συστατικό ***Communication*** που φαίνεται παρακάτω είναι υπεύθυνο για την παραλαβή αυτών των μηνυμάτων και την αποστολή αυτών στον συνδεδεμένο *MTA* του *client*. Παράδειγμα τέτοιας λειτουργίας πραγματοποιείται με την υποβολή ερωτήματος από το *Search Tool*.

Παρακάτω παρουσιάζεται η αρχιτεκτονική του client *ezDL* και της υποδομής επικοινωνίας του.



Επεξεργασία ερωτημάτων

Το ezDL παρέχει κάποια σύνταξη για τα ερωτήματα που υποβάλλει ο χρήστης επιτρέποντας την δημιουργία πιο σύνθετων ερωτημάτων. Οι όροι του ερωτήματος μπορούν να υποβληθούν με χρήση λογικών πράξεων AND, OR και NOT και με τελεστές απόστασης τύπου NEAR. Για παράδειγμα η σύνταξη

```
term1 NEAR/2 term2
```

σημαίνει ότι ο όρος term1 πρέπει να βρίσκεται σε απόσταση λιγότερη ή ιση με 2 από τον όρο term2. Η απόσταση υπολογίζεται με την ύπαρξη λέξεων ανάμεσα στους όρους

Το ερώτημα περιγράφεται απο το σύνολο των όρων του και από τους τελεστές που έχουν οριστεί για αυτούς. Το ezDL χρησιμοποιεί μια δενδρική μορφή αναπαράστασης των όρων του κάθε ερωτήματος και παρέχει μια σειρά από λειτουργίες για την επεξεργασία τους. Για παράδειγμα, αν το σύστημα παρατηρήσει μια παύση κατα την πληκτρολόγηση των όρων, οι όροι μεταφέρονται στο σύστημα υποστήριξης και προτροπής και, αφού εξεταστούν σε κλάσματα δευτερολέπτων, το σύστημα προτρέπει μια λίστα με πιθανούς εναλλακτικούς όρους. Οι εναλλακτικές επιλογές μπορεί να είναι γραμματική διόρθωση, επιλογή και διαστολή συνωνύμων.

Τεχνολογίες

Xpath

Η XPath είναι μια γλώσσα εκτέλεσης ερωτημάτων σε XML έγγραφα. Η πρώτη έκδοση (Xpath 1.0) προτάθηκε από τον W3C το 1997 θεμελιωμένη στα πρότυπα XLST και Xpointer. Η δεύτερη έκδοση υλοποιήθηκε το 2007 και προσέθεσε καινούριες λειτουργίες επιλογής και επεξεργασίας των αντίστοιχων κόμβων.

Η XPath δημιουργήθηκε για την εύκολη επιλογή κόμβων που ακολουθούν κάποια κριτήρια και ανήκουν στη δενδρική δομή ενός XML εγγράφου. Η βασικότερη έκφραση της γλώσσας Xpath είναι αυτή του μονοπατιού, η οποία ορίζει τον τρόπο προσπέλασης ενός ή παραπάνω κόμβων του XML εγγράφου. Το μονοπάτι καθορίζεται από την αλληλουχία κόμβων, οι οποίοι διαχωρίζονται με τον ειδικό χαρακτήρα / (slash).

Οι δυνατότητες της γλώσσας Xpath δεν περιορίζονται μόνο στην επιλογή των ζητούμενων κόμβων, αλλά και σε λειτουργίες όπως η ταύτιση και η διαχείριση αλφαριθμητικών γραμματοσειρών και η εφαρμογή αριθμητικών εκφράσεων στους επιλεγμένους κόμβους. Μάλιστα, στη δεύτερη έκδοση προστέθηκαν και δυνατότητες εκφράσεων βρόγχων (loop expression), συνθηκών ελέγχου περιπτώσεων (if-then-else) και δημιουργίας συναρτήσεων. Ωστόσο, η πλειοψηφία του προγραμματιστικού κοινού αναγνωρίζει ως μειονέκτημα την αλλαγή που επήλθε στη δεύτερη έκδοση, σχετικά με την αδυναμία δημιουργίας εγγράφου XML από κάποιο ερώτημα. Η λειτουργία αυτή υπήρχε στην πρώτη έκδοση του Xpath, ενώ σήμερα η δυνατότητα δημιουργίας εγγράφου XML από κάποιο ερώτημα παρέχεται από τη γλώσσα Xquery.

Παρατίθεται ένα απλό παράδειγμα χρήσης της γλώσσας Xpath.

Έστω ότι έχουμε το XML έγγραφο :

```
<portal>
     <cameras>
          <digital>
               <cam brand="Canon" model="A60" price="550"></cam>
          </digital>
          <SLR>
               <cam brand="Canon" model="EOS-3" price="980"></cam>
          </SLR>
     </cameras>
</portal>
```

Ένα μονοπάτι προσπέλασης όλων των ψηφιακών μηχανών είναι:
`/cameras/digital`

Ενώ για να προσπελαστεί η ιδιότητα τιμή κάθε SLR μηχανής, το μονοπάτι είναι:
`/cameras/SLR/cam/@price`

Ωστόσο, επειδή τις περισσότερες φορές αυτό που χρειάζεται για την επεξεργασία ενός XML εγγράφου είναι η επιλογή των κόμβων εκείνων που ικανοποιούν συγκεκριμένες συνθήκες, η XPath ορίζει προθέματα σε κάθε κόμβο που περιλαμβάνεται στο μονοπάτι. Ως προθέματα ορίζονται οι Boolean εκφράσεις που αποτιμώνται σε ένα κόμβο. Ένα τέτοιο παράδειγμα είναι το παρακάτω, στο οποίο επιλέγονται οι SLR κάμερες μάρκας Canon με τιμή μεγαλύτερη από 100 Ευρώ.

/cameras/SLR/cam[@brand = 'Canon' and @price >= 100]

Mercurial SCM

Το Mercurial SCM (Source Content Management) είναι ένα από τα δημοφιλέστερα Συστήματα Διαχείρισης Εκδόσεων (Version Control System) που χρησιμοποιούνται σήμερα από την προγραμματιστική κοινότητα. Δημιουργήθηκε από τον Matt Madkall το 2005 και διανέμεται υπό την άδεια GNU General Public Licence v2.

Τα Συστήματα Διαχείρισης Εκδόσεων έχουν αποκτήσει ενεργό ρόλο στη διαδικασία ανάπτυξης λογισμικού, καθώς συνιστούν εν μέρει κάποια από τα βασικότερα εργαλεία οργάνωσης και διαχείρισης της ίδιας της διαδικασίας. Οι κύριες λειτουργίες ενός VCS αποσκοπούν στην αποθήκευση διαφορετικών εκδόσεων λογισμικού - είτε αυτό αφορά αρχεία κειμένου είτε εκτελέσιμα αρχεία - στην καταγραφή των αλλαγών που πραγματοποιήθηκαν κατά τη μετάβαση από τη μία έκδοση στην άλλη, στον προσδιορισμό του χρήστη που πραγματοποίησε τις αλλαγές, αλλά και του χρόνου κατά τον οποίο αυτές έλαβαν χώρα. Η ανταλλαγή εκδόσεων του λογισμικού μεταξύ των μελών μιας ομάδας ανάπτυξης λογισμικού, καθώς και η επαναφορά σε προηγούμενες εκδόσεις, είναι εφικτές, όταν αυτό θεωρηθεί απαραίτητο.

Κύρια χαρακτηριστικά του VCS είναι η ταχύτητα, η ευκολία χρήσης του και η δυνατότητα επεκτασιμότητάς του. Τα παραπάνω στοιχεία βασίζονται στο σχεδιασμό του συστήματος, καθώς και στις γλώσσες (Python, C) που χρησιμοποιήθηκαν για την υλοποίησή του. Επίσης, φημίζεται για τη δυνατότητα πρόσθεσης επιπλέον βιβλιοθηκών (extensions) και για την προσαρμογή του εργαλείου στις προσωπικές ανάγκες του κάθε προγραμματιστή. Για παράδειγμα, το Shelve Extension επιτρέπει τον παραγκωνισμό κάποιου κώδικα από το working directory και την επαναφορά του οποιαδήποτε στιγμή.

Το Mercurial ανήκει στην κατηγορία των Κατανεμημένων Συστημάτων Διαχείρισης Εκδόσεων (Distributed Version Control Systems), καθιστώντας το παραπάνω από επαρκές στις σύγχρονες ανάγκες ταχείας ανάπτυξης λογισμικού. Έτσι, κάθε προγραμματιστής έχει άμεση και γρήγορη πρόσβαση σε όλο το ιστορικό των αλλαγών, ακόμα και αν δεν είναι συνδεδεμένος στο δίκτυο, αφού οι πληροφορίες των συνόλων αλλαγών (changesets) βρίσκονται τοπικά στο μηχάνημά του. Η διαφορά της εντολής push και commit δίνει στον προγραμματιστή την επιπλέον δυνατότητα να πειραματίζεται με αλλαγές στον κώδικά του, εωσότου είναι έτοιμος να υποβάλει μια βελτιωμένη έκδοση. Τέλος, επιτρέπει τη λειτουργία της συγχώνευσης (merge) διαφορετικών εκδόσεων, χρησιμοποιώντας τη μέθοδο της συγχώνευσης τριών βημάτων (three way merge algorithm).

Παρακάτω περιγράφονται κάποιες από τις συνήθεις έννοιες που συναντάμε στο Mercurial.

Repository: Πρόκειται για το χώρο, στον οποίο εγγράφονται τα αρχεία. Κατά τη διαδικασία του commit ο κάθε προγραμματιστής αποθηκεύει τις αλλαγές στο δικό του αποθετήριο (repository) μαζί με τα απαραίτητα meta-data της καινούριας έκδοσης.

Working directory: Είναι ο κατάλογος που διαβάζει και επεξεργάζεται ο προγραμματιστής. Μπορεί να περιέχει αρχεία από προηγούμενη έκδοση ή κάποια που δεν έχουν γίνει ακόμα commit. Πολλές φορές καλείται και working copy.

Revision: Σε κάθε repository αποθηκεύονται δεκαδικοί αριθμοί των εκδόσεων του λογισμικού μετά από οποιαδήποτε αλλαγή. Οι δεκαδικοί αυτοί αριθμοί αντιπροσωπεύουν μια διαφορετική έκδοση

του λογισμικού(revision).

Branch: Μια συνηθισμένη λειτουργία των VCS είναι η δημιουργία των branches. Κάθε branch έχει την ανεξαρτησία του όσον αφορά στην εξέλιξη των revision που θα το απαρτίζουν. Κατά τη δημιουργία ενός branch, η γραμμική πορεία της ανάπτυξης λογισμικού διαχωρίζεται, ενώ δίνεται και η δυνατότητα συνένωσης αυτών με την εντολή merge.

Merge: Πρόκειται για τη διαδικασία συνένωσης δυο branch και η επανάθεση της γραμμικής πορείας ανάπτυξης του λογισμικού. Συνήθως, χρησιμοποιούνται επιπλέον εργαλεία για την εύρεση και επίλυση διαφορών μεταξύ των δυο branch.

Commit: Είναι η υπεύθυνη εντολή για την εγγραφή των αλλαγών στο repository.

Pull/Push/Update: Δεδομένου ότι τα repositories στο Mercurial είναι αυτάρκη αποθετήρια (self-contained), για να γίνει ανταλλαγή εκδόσεων κώδικα, ουσιαστικά θα πρέπει να γίνει ανταλλαγή πληροφορίας μεταξύ των repositories. Αυτό επιτυγχάνεται με την εντολή pull και push, ορίζοντας την κατεύθυνση αντιγραφής των δεδομένων, ενώ η ενημέρωση του working directory επιτυγχάνεται με την εντολή update.

Apache Maven

Το Apache Maven είναι ένα ολοκληρωμένο εργαλείο διαχείρισης και αυτοματοποίησης σύνθετων διαδικασιών μεταγλώττισης, παραγωγής και ανάπτυξης προγραμμάτων. Αν και συχνά αναφέρεται ως μια εναλλακτική λύση του εργαλείου Ant, το Maven αποτελεί μια σουίτα συνεκτικών λειτουργιών του κύκλου παραγωγής ενός λογισμικού. Χάρη στην πληθώρα των επιπλέον συστατικών λογισμικού που μπορούν να προστεθούν στο κυρίως πρόγραμμα, τα λεγόμενα plug-ins, το Maven παρέχει μια ολοκληρωμένη λύση *επαλήθευσης, μεταγλώττισης, ελέγχου, πακεταρίσματος, αναφοράς* και *ανάπτυξης* προγραμμάτων.

Οι ρυθμίσεις για την εκάστοτε λειτουργία του εργαλείου Maven καθορίζονται σε ένα xml αρχείο, που υπάρχει σε κάθε Maven project. Αυτές μπορεί να αφορούν στο είδος των plug-ins που θα χρησιμοποιηθούν, στις εκδόσεις των διαφόρων συστατικών του λογισμικού, στον προσδιορισμό των paths και σε οποιαδήποτε ρύθμιση που μπορεί να προσαρμόζει τις υπάρχουσες προεπιλεγμένες.

Το Maven υποστηρίζει την έννοια του *componentization,* καθώς δίνει τη δυνατότητα σε κάποιον να δημιουργήσει ένα multi-module project. Έτσι, τοποθετώντας sub-projects σε υποφακέλους κάτω από το φάκελο του κυρίως project, δημιουργείται μια δομή από ένα σύνολο ιεραρχημένων projects. Με τον τρόπο αυτό, δίνεται η δυνατότητα μαζικής διαχείρισής τους, καθώς και αυτοματοποιημένης εκτέλεσης εντολών *mvn* για όλα τα υπο-project. Η δομή αυτή παρουσιάζεται παρακάτω.

```
<project>
   <!--  .... -->
  <modules>
    <module>ezdl</module>
    <module>examples</module>
    <module>framework</module>
    <module>gframedl</module>
    ...
</project>
```

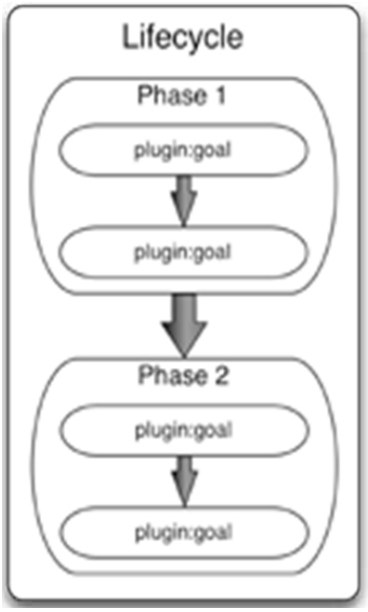
Κάθε Maven project παράγει ένα αρχείο τύπου JAR, WAR ή EAR προσδιορίζοντας τη μοναδικότητά του με μια συλλογή από πεδία που ονομάζονται *Artifact Vector.* Τα πεδία αυτά αναφέρονται στο *groupId* του, στο *artifactId* του, στην έκδοσή του και στον τύπο του αρχείου που θα παραχθεί.

(groupid:artifactid:version:type:scope)

Το *Artifact Vector* χρησιμοποιείται στην αναφορά των εξαρτήσεων που μπορεί να έχει ένα project. Οι εξαρτήσεις ορίζονται στο pom.xml και παρέχουν πληροφορία στο μηχανισμό του Maven, ώστε να αναζητήσει σε τοπικά και απομακρυσμένα αποθετήρια τις εξαρτήσεις αυτές. Σε πολύπλοκα δομημένα projects οι ίδιες οι εξαρτήσεις μπορεί να έχουν δευτερεύουσες εξαρτήσεις. Το Maven

διαθέτει μηχανισμό αναζήτησης των απαραίτητων εξαρτήσεων και διαχείρισης αυτών, αφού τους τοποθετήσει τοπικά. Μάλιστα, δίνεται η χρήσιμη δυνατότητα να ρυθμίζεται η συγκεκριμένη έκδοση της κάθε εξάρτησης. Για κάθε εξάρτηση είναι δυνατό να καθοριστεί το scope της ρυθμίζοντας έτσι τη σχέση της με το τελικό πακεταρισμένο artifact. Έτσι, αν, για παράδειγμα, το *scope* ρυθμιστεί να είναι ίσο με *test,* τότε υποδηλώνεται ότι η εξάρτηση θα χρησιμοποιηθεί μόνο στη διαδικασία του test και όχι στο τελικό πακετάρισμα.

Ο κύκλος εκτέλεσης του Maven διαιρείται σε τέσσερις ιεραρχικά ενφωλιασμένες κατηγορίες. Από την πιο αφαιρετική έως την πιο συγκεκριμένη, οι κατηγορίες είναι οι ακόλουθες και η μεταξύ τους σχέση φαίνεται στο παρακάτω σχήμα: Life-cycle, Phase, Plugin και Goal.



Η προσέγγιση αυτής της κατηγοριοποίησης επιχειρεί να πλαισιώσει τα βήματα που ακολουθούνται σε οποιοδήποτε λογισμικό, ανεξαρτήτως γλώσσας, κατά τη διαδικασία μεταγλώττισης και ανάπτυξής τους.

Ο κύκλος ζωής του Maven αποτελείται από τις φάσεις, μια αλληλουχία βημάτων που μαζί ολοκληρώνουν το έργο παραγωγής του λογισμικού. Κάθε φάση είναι συνδεδεμένη με ένα *plugin* και κάθε *plugin* περιέχει έναν ή περισσότερους στόχους. Μέσω της γραμμής εντολών δίνεται η δυνατότητα εκτέλεσης μιας φάσης ή ενός στόχου. Αν η εντολή αναφέρεται στην εκτέλεση κάποιου στόχου, τότε θα εκτελεστεί αυτός και μόνο αυτός. Από την άλλη, αν ζητηθεί ρητή εκτέλεση μιας φάσης, τότε θα εκτελεστούν όλοι οι στόχοι και οι προγενέστερες φάσεις του κύκλου ζωής του λογισμικού.

Παρακάτω ακολουθεί μια συνοπτική περιγραφή των ενσωματωμένων κύκλων ζωής (lifecycles) του Maven.

Clean life-cycle: Διαγράφει όλα τα υπάρχοντα *artifacts* που βρίσκονται στο *output directory* του λογισμικού.

Default life-cycle: Ο κύκλος αυτός περιέχει τις πιο συχνά χρησιμοποιούμενες φάσεις της

παραγωγής και ανάπτυξης του λογισμικού. Τέτοιες είναι οι φάσεις του *validate,compile, test, test-compile, package, verify, install, deploy κ.α.*

Site life-cycle: Ο κύκλος αυτός παράγει μια ιστοσελίδα, η οποία παρέχει λεπτομερείς πληροφορίες και αναφορές για τα modules που βρίσκονται στο pom.xml αρχείο. Επίσης, δίνεται η δυνατότητα φόρτωσης ενός self-standing web server για τη φόρτωση αυτού του site.

MySQL

Η MySQL είναι ένα από τα πιο γνωστά Σχεσιακά Συστήματα Διαχείρισης Βάσεων Δεδομένων (Relation Database Management System). Όπως σε κάθε ΣΔΒΔ, στόχος είναι η εύκολη και γρήγορη ανάκτηση και διαχείριση δεδομένων που αποθηκεύονται στη βάση. Κάθε ΒΔ βασίζεται σε ένα μοντέλο δεδομένων, βάσει του οποίου ορίζονται οι συσχετίσεις μεταξύ των πινάκων, καθώς και οι περιορισμοί που ενδέχεται να έχουν τα πεδία τους.

Για τη διαχείριση των δεδομένων της εκάστοτε ΒΔ, η MySQL χρησιμοποιεί τη διαδεδομένη γλώσσα ερωτημάτων SQL (Structure Query Language). Βασισμένη στη σχεσιακή άλγεβρα, η SQL αποτελεί διεθνές πρότυπο ISO. Σήμερα είναι η πιο ευρέως χρησιμοποιούμενη γλώσσα για τις σχεσιακές ΒΔ και παρέχει μια πληθώρα δυνατοτήτων, κάποιες από τις οποίες θα αναφερθούν παρακάτω:

- *τη δημιουργία, τη διαγραφή και την μεταβολή των πινάκων καθώς και των σχετικών περιορισμών που μπορεί να διέπουν τα διάφορα πεδία τους.*
- *τη σύνταξη ερωτημάτων υπολογισμού και ανάκτησης, μεταβολής και διαγραφής δεδομένων των πινάκων μιας ΒΔ.*
- *τη δημιουργία και διαγραφή όψεων σε μια ΒΔ*
- *τη δημιουργία διαδικασίων σε μια ΒΔ (stored procedures)*
- *τον ορισμό δικαιωμάτων πρόσβασης σε πίνακες, διαδικασίες(procedures) και όψεις(views).*

Χαρακτηριστικά της MySQL

Ταχύτητα: Το MySQL θεωρείται ένα πολύ γρήγορο Σύστημα Διαχείρισης Βάσεων Δεδομένων. Τα συμπεράσματα αυτά δε βασίζονται μόνο σε μαρτυρίες χρηστών, αλλά προκύπτουν και από μια σειρά benchmark tests που έχουν γίνει. Με τη χρήση πολλαπλών νημάτων επιτυγχάνεται η αξιοποίηση όλων των πόρων του διακομιστή. Επιπλέον, είναι γνωστό ότι χρησιμοποιεί πολύ γρήγορα B-trees με συμπίεση στους δείκτες για ταχείες ανακτήσεις δεδομένων σε μεγάλους πίνακες.

Επεκτασιμότητα: Το MySQL μπορεί να ελέγξει την πρόσβαση πολλών χρηστών ταυτόχρονα, αλλά και τις συναλλαγές που αυτοί θα έχουν με κάποια ΒΔ. Επίσης, δίνεται η δυνατότητα πολλαπλών συνδέσεων σε μια ΒΔ, χωρίς να δημιουργούνται αντίγραφά της. Υποστηρίζει τεράστιες ΒΔ, μεγέθους της τάξης των 200 χιλιάδων πινάκων και 50 εκατομμυρίων εγγραφών. Βασίζεται στην σχεδίαση πολλαπλών επιπέδων για την εύκολη προσθήκη επιπλέον modules.

Ασφάλεια: Ένα επιπλέον χαρακτηριστικό του MySQL σχετίζεται με την παροχή προνομίων και κωδικών ασφαλείας για τη σύνδεση των χρηστών στο διακομιστή και με τη χρήση κρυπτογράφησης καθ' όλη τη μεταφορά δεδομένων για κάθε σύνδεση στη ΒΔ.

Συμβατότητα: Οι εφαρμογές-πελάτες (clients) έχουν τη δυνατότητα να συνδεθούν χρησιμοποιώντας διαφορετικά πρωτόκολλα σε διαφορετικες πλατφόρμες. Για παράδειγμα, υπάρχει

η δυνατότητα σύνδεσης με TCP/IP socket ή με Unix domain socket σε πλατφόρμες Unix.

Τοπική προσαρμογή: Τα δεδομένα μπορούν να σωθούν σε διαφορετικά character set και αποκτούν τη δυνατότητα λειτουργιών όπως είναι η ταξινόμηση ή/και η επιλογή εγγραφών. Ο διακομιστής μπορεί να προβάλλει τα μηνύματα λάθους σε πολλές γλώσσες. Τέλος, παρέχεται η δυνατότητα της δυναμικής αλλαγής της ώρας, ανάλογα με τη ζώνη-ώρας, στην οποία βρίσκεται ο εκάστοτε πελάτης.

Υλοποίηση

Το κομμάτι που αφορά στην υλοποίηση μπορεί να διακριθεί σε δύο μέρη. Στο πρώτο μέρος θα αναλυθεί η ανάπτυξη ενός *agent* του συστήματος ezDL τύπου ***Wrapper***, ενώ στο δεύτερο μέρος θα αναλυθεί η ανάπτυξη ενός εργαλείου για τον *client*, συγκεκριμένα του ***Translation Tool***. Και στις δύο περιπτώσεις, η ανάπτυξη έχει ως κύριο στόχο την *επεκτασιμότητα* του συστήματος. Στην περίπτωση του *Wrapper*, γίνεται η προσθήκη νέων πόρων για το παρόν σύστημα αναζήτησης, ενώ στην περίπτωση του εργαλείου *Translation Tool*, γίνεται η προσθήκη μιας επιπλέον λειτουργίας όσον αφορά στην υποβολή των ερωτημάτων.

Κατηγορία Wrapper

Το *component* αυτό αποτελεί μέρος του back-end του συστήματος ezDL και αποσκοπεί στη μεσολάβηση μεταξύ του συστήματος και της υπηρεσίας για την οποία θα οριστεί αρμόδιο. Οι κύριες λειτουργίες ενός *Wrapper* είναι η μορφοποίηση του ερωτήματος που θα υποβληθεί τελικά σε κάποια υπηρεσία και η εξαγωγή της ζητούμενης πληροφορίας από τα δεδομένα που αυτή θα επιστρέψει.

Για την υποβολή ενός ερωτήματος που εμπεριέχει κάποιους όρους, το ezDL δημιουργεί ένα αντικείμενο τύπου *Query*. Δεδομένου ότι το *Query* μπορεί να έχει διαφορετικά πεδία, όπως, για παράδειγμα, το όνομα ενός συγγραφέα ή τον τίτλο ενός βιβλίου, τα πεδία αυτά συγκεντρώνονται σε μια δενδρική δομή, στην οποία κάθε φύλλο του δένδρου αντιστοιχεί σε διαφορετικό πεδίο. Κάθε *Wrapper* λαμβάνει το ίδιο Query και ο τρόπος με τον οποίο θα συντάξει το τελικό ερώτημα που θα υποβληθεί στην εκάστοτε υπηρεσία.έγκειται στην αρμοδιότητά του. Στην περίπτωση που περιγράφεται παρακάτω, οι όροι της δεντρικής δομής χρησιμοποιούνται για τη δημιουργία ενός HTTP ερωτήματος. Αν και αυτό μπορεί να φαίνεται εύκολο όταν οι όροι είναι απλές λέξεις, ένα αντικείμενο τύπου *Query* μπορεί να αποθηκεύσει *Boolean* σχέσεις μεταξύ των όρων και αντίστοιχα αυτές να καταχωρηθούν στο ερώτημα του HTTP. Για την επίτευξη αυτών των σχέσεων, η δενδρική δομή αποθηκεύει κόμβους *Query Nodes* τύπου *"And"* και *"Or"*.

Τέλος, αν το αίτημα του HTML είναι συντακτικά σωστό και αν η υπηρεσία στην οποία υποβάλλεται το ερώτημα είναι ενεργή, ο *Wrapper* λαμβάνει την αντίστοιχη απάντηση. Η απάντηση που λαμβάνει ένας *Wrapper* μπορεί είναι είτε ένα XML είτε ένα HMTL αρχείο. Στην υλοποίηση που αναλύεται παρακάτω, ο *Wrapper* επιστρέφει ένα XML έγγραφο. Τα δεδομένα του εγγράφου αυτού απεικονίζονται σε κατάλληλα πεδία και αποθηκεύονται σε ένα αντικείμενο τύπου *WrapperDLObjectList,* μέσω του οποίου επιστρέφονται στον *client* για να γίνει η προβολή τους στον χρήστη. Η διαδικασία αυτή της εξαγωγής των δεδομένων από το έγγραφο XML ονομάζεται *parsing* και γίνεται με τη χρήση της γλώσσας XPATH και βοηθητικών βιβλιοθηκών της Java.

Απαιτήσεις

Παρακάτω περιγράφονται οι μη λειτουργικές και οι λειτουργικές απαιτήσεις ενός προγράμματος Wrapper γενικά, και του ClefIP IPC Wrapper ειδικότερα.

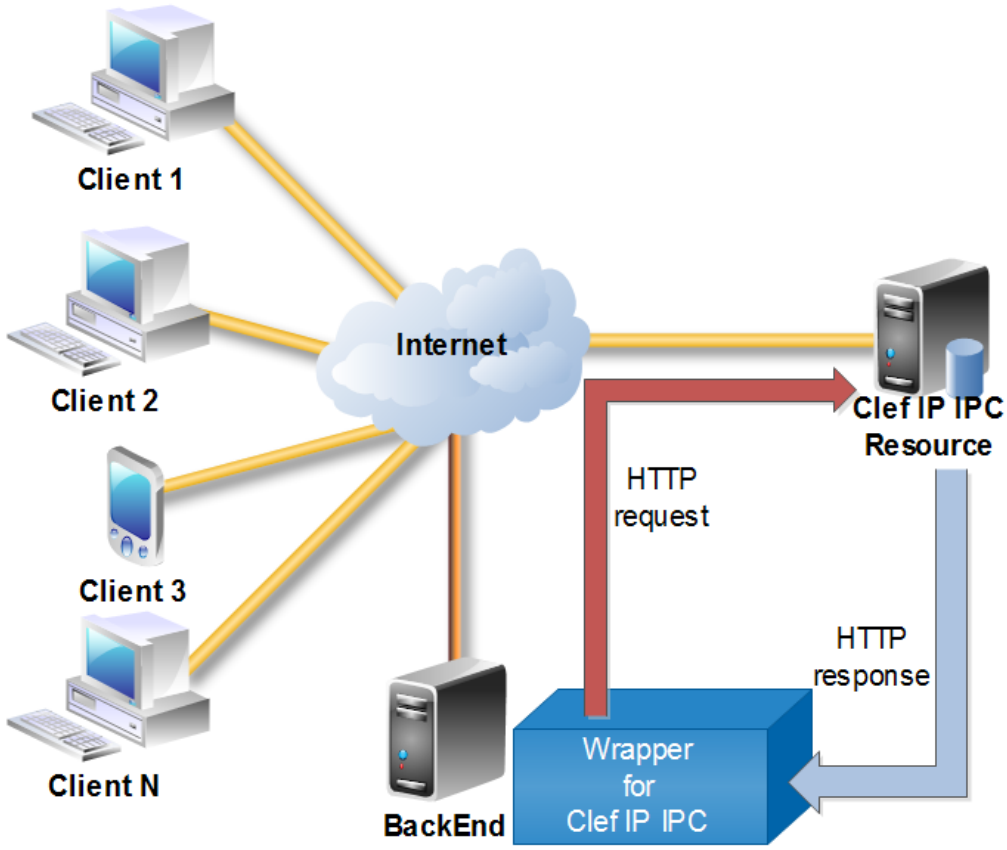
Μη λειτουργικές απαιτήσεις				
#	**Κατηγορία**	**Υποκατηγορία**	**Υπο-υποκατηγορία**	**Περιγραφή**
1	Quality Of Service	Reliability	Integrity	Ο Wrapper θα πρέπει να επιστρέφει αναφορά και στις περιπτώσεις που η αντίστοιχη υπηρεσία δεν είναι διαθέσιμη.
2			Integrity	Το σύστημα θα πρέπει να μπορεί να επιστρέψει μετά από οποιαδήποτε απρόοπτη λήξη του Wrapper κρατώντας ένα ιστορικό με τα ερωτήματα που έχουν ήδη υποβληθεί.
3			Independence	Ο Wrapper θα πρέπει να λειτουργεί ανεξάρτητα από τους υπόλοιπους wrappers. Η συλλογή αποτελεσμάτων του δεν πρέπει να επηρεάζει και να επηρεάζεται απο τις συλλογές των υπολοίπων wrapper.
4			Independence	Η αποτυχία επιστροφής δεδομένων σε κάποια χρονική στιγμή, δεν πρέπει να καθιστά αδύνατη την λειτουργία του wrapper για τις μετέπειτα υποβολές ερωτημάτων.
5			Independence	Κάθε Wrapper θα πρέπει να μπορεί να κάνει εκκίνηση όχι μόνο κατά την εκκίνηση του κεντρικού συστήματος αλλά και οποιαδήποτε άλλη στιγμή.
6				Το σύστημα θα πρέπει να μπορεί να κρατάει ένα log αρχείο με όλα τα σφάλματα και τις προειδοποιήσεις που πραγματοποιούνται κατά τη διαδικασίας αναζήτησης.
7		Performance	Time	Η διαδικασία υποβολής ερωτημάτων από το Wrapper θα πρέπει να έχει κάποιο άνωτατο χρονικό όριο ώστε να μην καθυστερεί η όλη διαδικασία αναζήτησης.

8			Memory	Ο Wrapper θα πρέπει να μπορεί να λειτουργεί για πολύ καιρό χωρίς να χρειάζεται επανεκκίνησή του.
9		Scalability		Το σύστημα θα πρέπει να είναι ικανό να αντεπεξέρχεται στην αύξηση του πλήθους των wrapper χωρίς αυτό να επηρεάζει την απόδοσή του.
10		Extensibility		Θα πρέπει να μπορεί να εξάγει τα αποτελέσματα με διαφορετικούς τρόπους ανάλογα με τις επιστροφές της εκάστοτε υπηρεσίας. Χρήση XPATH γλώσσας ή με SAXHandler.

Λειτουργικές απαιτήσεις		
#	**Κατηγορία**	**Περιγραφή**
1	Parametrization	Ο Wrapper ClefIPIPC θα πρέπει να είναι ικανός να επιστρέφει αποτελέσματα IPCs για διαφορετικού τύπου level μεταξύ των τιμών 3, 4, 5.
2	Parametrization	Θα πρέπει να υποστηρίζει Boolean εκφράσεις μεταξύ των όρων του ερωτήματος, και αυτές αντίστοιχα να μεταφράζονται στο HTTP request.
3	Parametrization	Το πλήθος των αποτελεσμάτων θα είναι ακριβώς όσο είναι όταν το ερώτημα υποβάλλεται από την ιστοσελίδα της υπηρεσίας.
4	Time	Ο χρόνος του κύκλου ανάκτησης δεδομένων θα διαρκεί λιφότερο από 20 δευτερόλεπτα. Ειδάλλως θα στέλνεται κατάλληλο μήνυμα στον client.
5	Data Format	Τα αποτελέσματα θα πρέπει να αποθηκεύονται σε Unicode Strings.
6	Data Format	Κατά την εξαγωγή δεδομένων θα πρέπει να αποθηκεύονται τα πεδία: τίτλος και IPC της εκάστοτε εγγραφής.
7	Data Format	Τα αποτελέσματα πρέπει να αποθηκεύονται σε αντικείμενο τύπου *WrapperDLObjectList* πριν σταλλούν στον client.
8	Representation	Θα πρέπει να ενημερωθεί η λίστα με τους Wrapper για την οποιαδήποτε επιπλέον λειτουργία του προσφέρεται από τον Wrapper όσον αφορά στα πεδία Capabilities.

Αρχιτεκτονική υψηλού επιπέδου

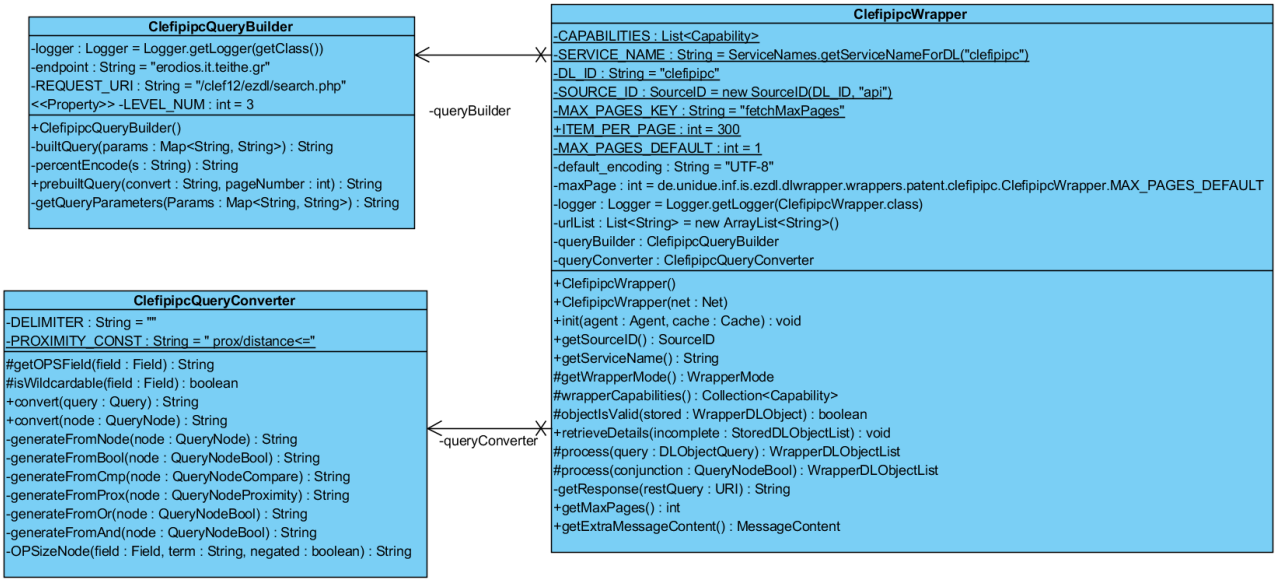
Στο παρακάτω διάγραμμα παρουσιάζεται μια αφαιρετική αναπαράσταση του συστήματος ezDL, καθώς και η τοποθεσία στην οποία βρίσκεται ο wrapper σε σχέση με το όλο σύστημα.



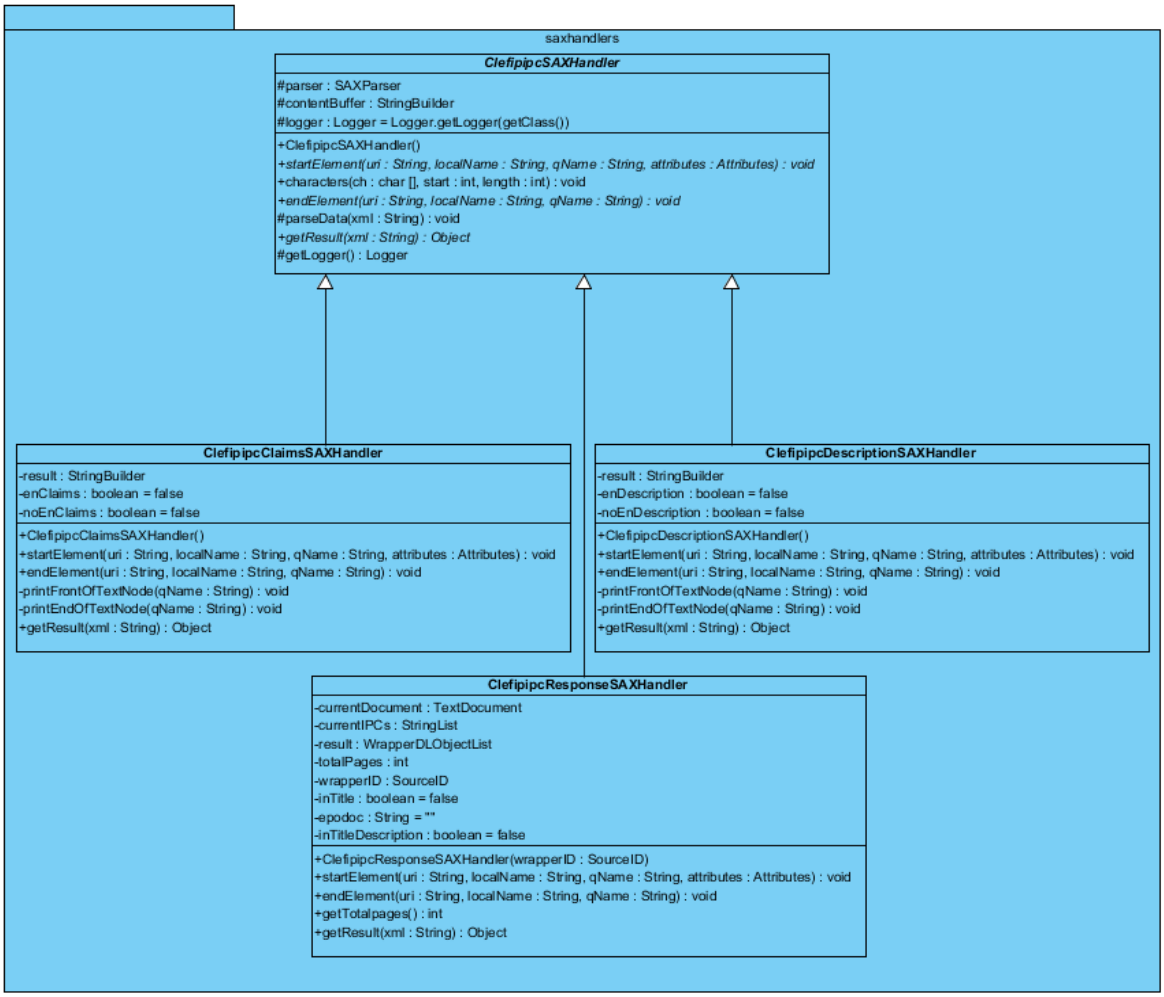
Διάγραμμα κλάσεων

Στο διάγραμμα κλάσεων ξεχωρίζουν οι τρεις κλάσεις **ClefipipcWrapper**, **ClefipipcQueryBuilder** και **ClefipipcQueryConverter**. Η πρώτη κλάση είναι η κύρια κλάση του Wrapper και είναι αρμόδια για την δημιουργία του στιγμιότυπου του Wrapper και το συγχρονισμό των λειτουργιών που αυτός θα διετελέσει.

Η βασική μέθοδος του Wrapper είναι η ***process,*** η οποία λαμβάνει ως όρισμα ένα αντικείμενο τύπου **QueryNodeBool** και επιστρέφει ένα αντικείμενο τύπου **WrapperDLObjectList.** Το **QueryNodeBool** περιέχει τους όρους που θα λάβει το στιγμιότυπο **ClefipipcQueryConverter,** το οποίο στη συνέχεια θα μετασχηματίσει τα ερωτήματα του ezDL σε αντίστοιχα ερωτήματα, συμβατά με τα αντίστοιχα Clef IP IPC υπηρεσίας. Για παράδειγμα ο κόμβος AND με δύο όρους αναπαρίσταται με ένα String που μεταξύ των δυο όρων θα έχει το " AND ". Στην ίδια μέθοδο το στιγμιότυπο **queryBuilder** της κλάσης **ClefipipcQueryBuilder** έχει αρμοδιότητα να συνθέσει το τελικό HTTP request.



Μετά την παραλαβή των αποτελεσμάτων, σειρά έχει η εξαγωγή των δεδομένων που ορίστηκαν από τις λειτουργικές απαιτήσεις του προγράμματος. Ο Wrapper παραλαμβάνει ένα XML αρχείο και, χρησιμοποιώντας συνδυασμό από στιγμιότυπα των κλάσεων που επεκτείνουν την **abstract** κλάση **ClefipipcSAXHandler,** διατρέχει τα ζητούμενα nodes και εξάγει τους τίτλους και τα IPCs των εγγραφών.



Κατηγορία front-end tool

Τα εργαλεία του front-end ποικίλουν ως προς τη συνεισφορά τους στο σύστημα αναζήτησης του ezDL. Κάθε ένα από αυτά μπορεί να προσφέρει διαφορετική λειτουργικότητα, γεγονός που κάνει το σύστημα ευέλικτο και προσαρμόσιμο στις ανάγκες του εκάστοτε χρήστη. Ο μηχανισμός επικοινωνίας μεταξύ των *components* του front-end επιτρέπει την εύκολη ενσωμάτωση νέων εργαλείων. Μάλιστα, αυτά μπορούν να αλληλοσυνεργάζονται και να παρέχουν πιο σύνθετες λειτουργίες.

Χωρίς την ύπαρξη των εργαλείων, τα αποτελέσματα θα αποτελούσαν μια τυποποιημένη λίστα από εγγραφές. Λειτουργίες όπως η ταξινόμηση, η ομαδοποίηση και η εύρεση λέξεων-κλειδιών μπορούν να εξοικονομήσουν πάρα πολύ σημαντικό χρόνο στο χρήστη και να βελτιώσουν τη διαδικασία της αναζήτησης. Πιο περίπλοκα και ενδιαφέροντα εργαλεία είναι αυτά που σχετίζονται με την οπτικοποίηση των δεδομένων και την εξαγωγή γνώσης από αυτά, μέσω των τεχνικών *clustering και entity extraction.*

Translation Tool

Η υλοποίηση του εργαλείου μετάφρασης όρων πραγματοποιήθηκε σε τρεις φάσεις. Στην πρώτη φάση, υλοποιήθηκε ένα αυτόνομο εργαλείο του front-end, ικανό να δέχεται κείμενο και να το μεταφράζει σε μια σειρά από επιλεγόμενες γλώσσες. Στη φάση αυτή χρησιμοποιήθηκε η online υπηρεσία μετάφρασης *Patentscope* με χρήση υποβολής αιτημάτων HTTP, τα οποία εμπεριέχουν τους όρους που πρέπει να μεταφραστούν. Στη δεύτερη φάση, κρίθηκε αναγκαία η προσθήκη νέων πηγών μετάφρασης, όπως αυτή της υπηρεσίας *Bing* της *Microsoft*, καθώς και μιας υπηρεσίας μετάφρασης για μεγαλύτερου μεγέθους κείμενα, η οποία διατίθεται από την *Patentscope.* Η προσθήκη αυτή αυξάνει την αξιοπιστία του εργαλείου, αφού πλέον η μετάφραση των όρων δεν εξαρτάται μόνο από μια πηγή. Επίσης, η υπηρεσία *Patentscope* για μεγαλύτερο μέγεθος κειμένου δίνει μια επιπλέον λειτουργικότητα στο εργαλείο που δεν υπήρχε πριν. Μάλιστα, η τελευταία υπηρεσία επιτρέπει την μετάφραση όρων επιλέγοντας το domain, στο οποίο αυτά ανήκουν. Έτσι, αν για παράδειγμα οι όροι που θέλει ο χρήστης να μεταφράσει σχετίζονται με τον τομέα της Αυτοκινητοβιομηχανίας, η μετάφρασή τους θα έχει μεγαλυτερο βαθμό εγκυρότητας, εφόσον μπορεί να γίνει στοχευμένα για αυτό το domain. Τέλος, στην τρίτη φάση το εργαλείο μετάφρασης έπρεπε να ενσωματωθεί στις λειτουργίες αναζήτησης του ezDL. Η φάση αυτή εμπεριέχει την διαδικασία του *integration* του εργαλείου με το όλο σύστημα και του τελικού ελέγχου του συστήματος μετά την ενσωμάτωσή του.

Απαιτήσεις

Παρακάτω περιγράφονται οι μη λειτουργικές και οι λειτουργικές απαιτήσεις ενός εργαλείου του front-end γενικά, και του εργαλείου **Translation Tool** ειδικότερα.

Μη λειτουργικές απαιτήσεις				
#	**Κατηγορία**	**Υποκατηγορία**	**Υπο-υποκατηγορία**	**Περιγραφή**
1	Quality Of Service	Reliability	Integrity	Το εργαλείο θα πρέπει να μπορεί να επεξεργάζεται τις εγγραφές όταν υπάρχουν αποτελέσματα για επεξεργασία.
2			Integrity	Το εργαλείο μετάφρασης θα πρέπει να είναι ικανό να ενημερώνει αν η υπηρεσία με την οποία συνδέεται δεν είναι διαθέσιμη εκείνη τη στιγμή.
3			Independence	Θα πρέπει να λειτουργεί ανεξάρτητα από τα υπόλοιπα εργαλεία, εκτός αν ζητηθεί ρητά κάτι διαφορετικό. Κοινώς, θα πρέπει να επιστρέφει μεταφρασμένους όρους ανεξάρτητα από την ολική χρήση του συστήματος.
4			Independence	Η αποτυχία επεξεργασίας ερωτημάτων σε κάποια χρονική στιγμή, δεν πρέπει να καθιστά αδύνατη την λειτουργία του εργαλείου για τις μετέπειτα επεξεργασίες ερωτημάτων.
5			Usability	Κάθε εργαλείο θα πρέπει να μπορεί να αλλάζει χρήση της υπηρεσίας που χρησιμοποιεί.
6			Usability	Όλα τα εργαλεία θα πρέπει να μπορούν να δηλωθούν είτε ως ενεργά είτε ως ανενεργά και αντίστοιχα να επεξεργάζονται ή όχι τα οποιαδήποτε δεδομένα παραλαμβάνονται.
7			Usability	Το γραφικό περιβάλλον θα πρέπει να ακολουθεί τη μορφή του όλου συστήματος.
8		Performance	Time	Η διαδικασία μετάφρασης των ερωτημάτων θα πρέπει να έχει κάποιο ανώτατο χρονικό όριο ώστε να μην καθυστερεί η όλη διαδικασία

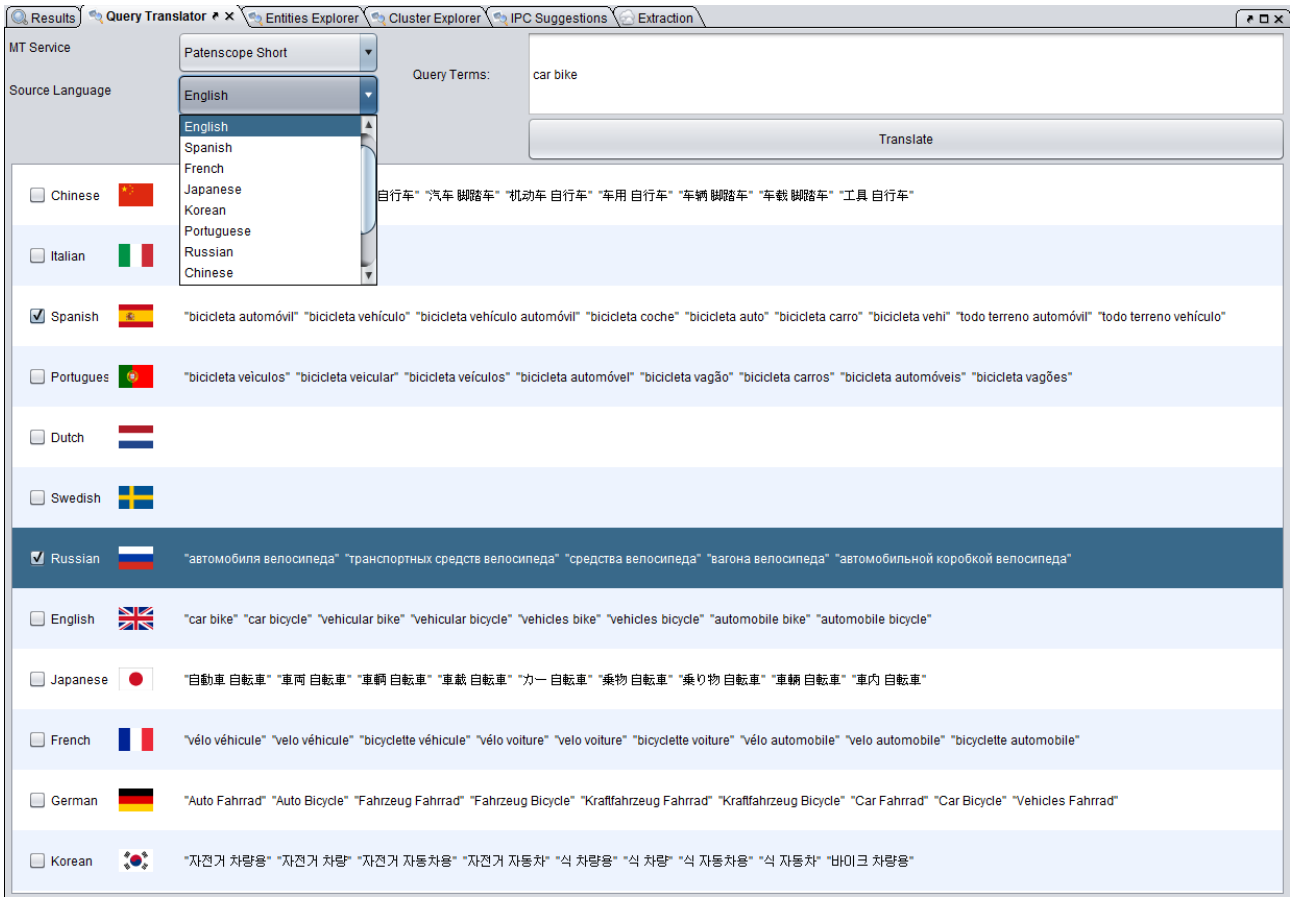
				αναζήτησης.
9		Scalability		Το σύστημα θα πρέπει να είναι ικανό να προσθέτει εύκολα νέα εργαλεία.
10		Extensibility		Το εργαλείο μετάφρασης θα πρέπει να δίνει τη δυνατότητα αύξησης επιπλέον υπηρεσιών μετάφρασης.
11		Extensibility		Θα πρέπει να είναι εύκολη η πρόσθεση επιπλέον γλωσσών στις οποίες θα γίνεται η μετάφραση.

Λειτουργικές απαιτήσεις		
#	**Κατηγορία**	**Περιγραφή**
1	Parametrization	Το εργαλείο μετάφρασης θα πρέπει να επιλέγει μια από τις υπηρεσίες μετάφρασης με τη χρήση ενός **combo-box**.
2	Parametrization	Για κάθε υπηρεσία μετάφρασης θα πρέπει να ορίζεται η αρχική γλώσσα στην οποία υποβάλλονται οι όροι.
3	Parametrization	Στην υπηρεσία μετάφρασης Patenscope-long θα πρέπει να δίνεται η δυνατότητα επιλογής ζεύγους γλωσσών.
4	Parametrization	Στην υπηρεσία μετάφρασης Patenscope-long θα πρέπει να δίνεται η δυνατότητα επιλογής του domain αναζήτησης. Τα επιτρεπτά domain θα εμφανίζονται μόνο σε αυτή την περίπτωση και η επιλογή τους θα γίνεται με τη χρήση ενός **combo-box**.
5	Time	Ο χρόνος επιστροφής των μετεφρασμένων όρων δε θα πρέπει να ξεπερνάει τα 20 δευτερόλεπτα. Σε διαφορετική περίπτωση θα καθυστερεί την διαδικασία αναζήτησης.
6	Data Format	Οι επιτρεπτές γλώσσες μετάφρασης ορίζονται σε μια κλάση **Enum,** για να μην υπάρξει ασυμβατότητα μεταξύ των υπηρεσίων μετάφρασης.
7	Data Format	Τα ζεύγη γλωσσών της υπηρεσίας Patenscope-long θα πρέπει να ορίζονται σε μια κλάση **Enum**.
8	Data Format	Τα επιτρεπτά domain της υπηρεσίας Patenscope-long θα πρέπει να ορίζονται σε μια κλάση **Enum**.
9	Data Format	Οι μεταφρασμένοι όροι θα πρέπει να είναι αποθηκευμένοι ως **Unicode String**, με μέγιστο τους 255 χαρακτήρες.
10	Error Handling	Τα ερωτήματα που υποβάλλονται στις υπηρεσίες μετάφρασης θα πρέπει να κάνουν **throw**

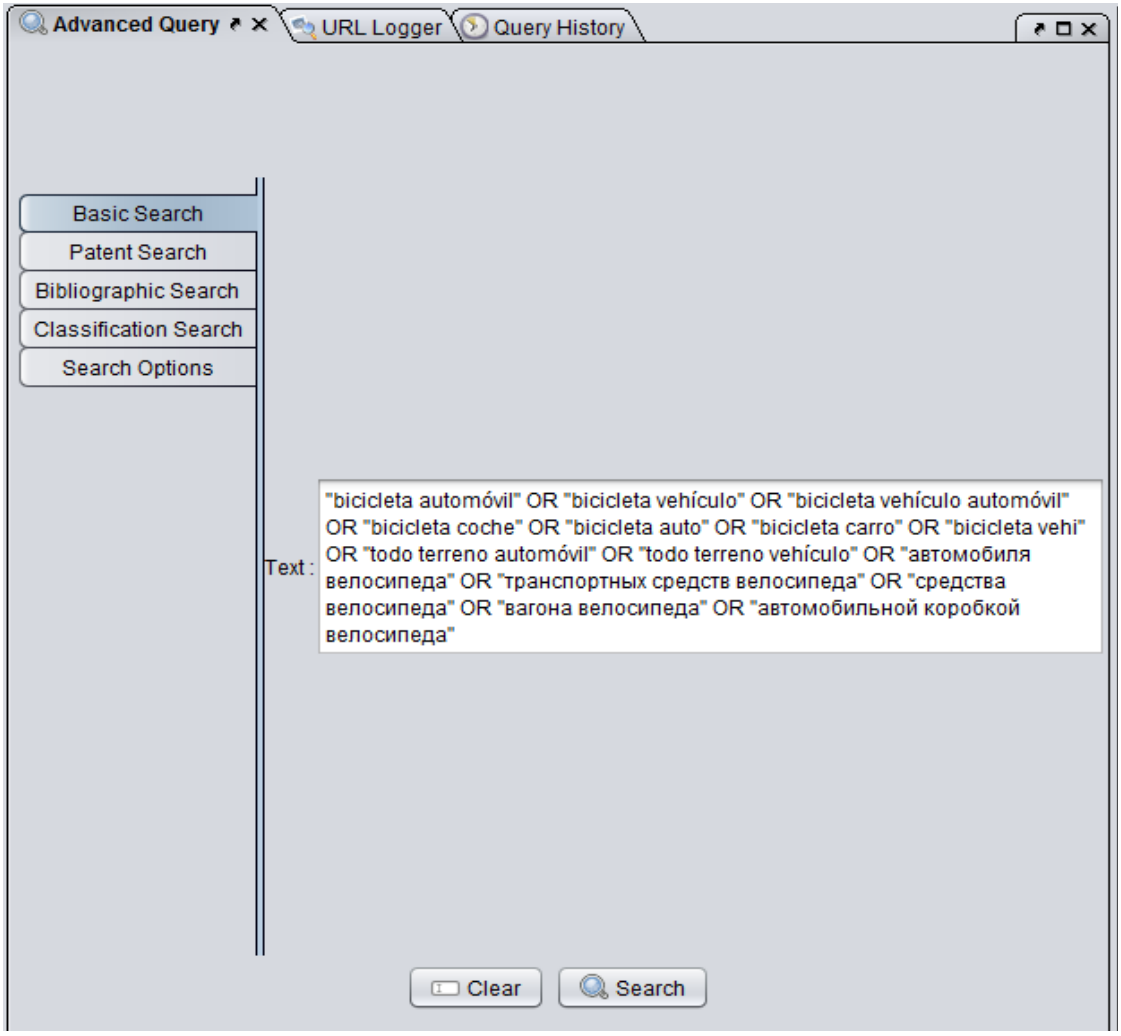
		NonMatchingLanguagePairException στην περίπτωση ύπαρξης κάποιου λάθους.
11	Error Handling	Αν κάποια από τις υπηρεσίες δεν είναι διαθέσιμη να γίνεται αντίστοιχο **throw Exception.**
12	Representation	Θα πρέπει να ενημερωθεί η λίστα με τους μετάφρασμένους όρους για την κάθε γλώσσα μετά από την κλήση της μετάφρασης.
13	Representation	Η λίστα θα πρέπει να αλλάζει δυναμικά μέγεθος ανάλογα με το πλήθος των μετάφρασμένων όρων.
14	Representation	Για κάθε γλώσσα θα πρέπει να υπάρχει ο τίτλος της και μια εικόνα για την εθνική σημαία της χώρας που χρησιμοποιεί αυτή την γλώσσα.
15	Representation	Η λίστα της αναπαράστασης των γλωσσών μετάφρασης και των μετεφρασμένων όρων θα πρέπει να βασίζεται στην κλάση η οποία θα κάνει επέκταση της κλάσης **CheckBoxListModel** του **ezDL.**
16	Interaction	Η λήψη των όρων οι οποίοι πρέπει να μεταφραστούν να γίνεται με τη χρήση ειδικου Event, του **TranslationEvent** το οποίο κάνει επέκταση το **GframeEvent** του **ezDL.**
17	Interaction	Θα πρέπει να συνδέεται το **Query-View** του συστήματος με τις επιλεγόμενες γλώσσες έτσι ώστε οι μετεφρασμένοι όροι που έχουν επιλεγεί να τοποθετούνται στο ερώτημα που θα υποβληθεί.
18	Compatibility	Αν τα υποβληθέντα ερωτήματα εμπεριέχουν Boolean τελεστές αυτοί δεν πρέπει να μεταφράζονται διότι δεν έχουν σημασιολογική αξία.
19	Compatibility	Ο συνδυασμός των μετάφρασμένων όρων θα πρέπει να δημιουργεί ένα σύνθετο ερώτημα με τη βοήθεια τελεστών OR.
20	Compatibility	Ο συνδυασμός των μετεφρασμένων όρων από τις διαφορετικές γλώσσες θα πρέπει να δημιουργεί ένα σύνθετο ερώτημα με τη βοήθεια τελεστών OR.

Λειτουργία του Translation Tool

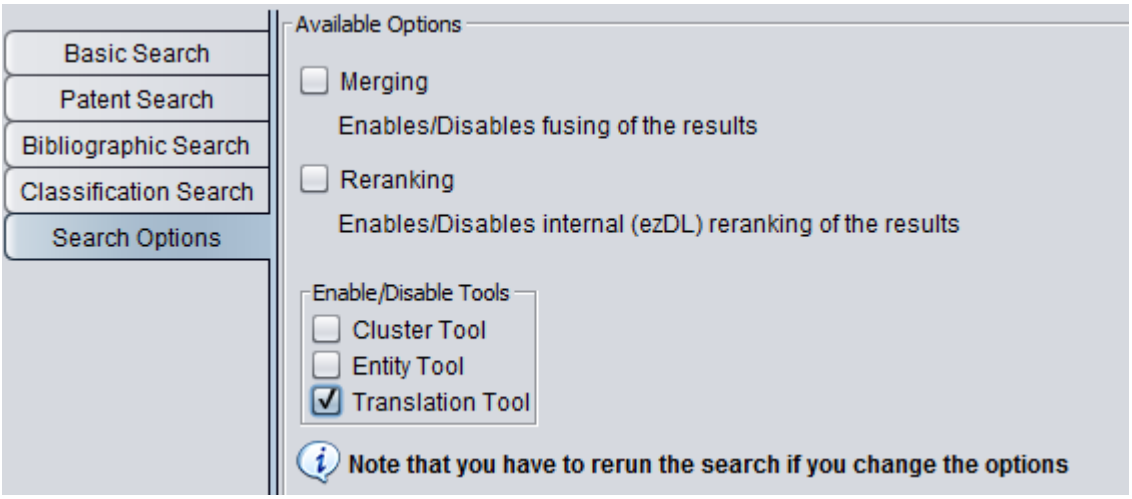
Παρακάτω απεικονίζεται το γραφικό περιβάλλον του εργαλείου μετάφρασης. Αριστερά έχουμε δύο **combo-boxes,** ένα για την επιλογή της υπηρεσίας μετάφρασης και ένα για τη γλώσσα των υποβαλλόμενων όρων. Το **text-field** και το κουμπί **Translate** στα δεξιά δίνουν την ελευθερία στον χρήστη να χρησιμοποιεί το εργαλείο μετάφρασης ανεξάρτητα από την ολική λειτουργία του συστήματος αναζήτησης.



Για να γίνει η αλληλεπίδραση του εργαλείου μετάφρασης με το όλο υπόλοιπο σύστημα, ο χρήστης επιλέγει τις γλώσσες που επιθυμεί στα **check-boxes** που υπάρχουν στην λίστα. Σε αυτή την περίπτωση, οι όροι των επιλεγμένων γλωσσών δημιουργούν ένα σύνθετο ερώτημα, το οποίο τοποθετείται στο **text-field** του **Query-View,** όπως φαίνεται παρακάτω. Ωστόσο, η υποβολή του σύνθετου ερωτήματος γίνεται με το πάτημα του κουμπιού **Search.**



Ο χρήστης μπορεί να ρυθμίσει αν το εργαλείο μετάφρασης θα λειτουργεί για κάθε αναζήτηση. Τότε σε κάθε αναζήτηση γίνεται η αυτόματη μετάφραση των όρων που υπάρχουν στο **text-field** του **Query-View,** η δημιουργία του σύνθετου ερωτήματος και η τελική υποβολή του στις υπηρεσίες αναζήτησης. Η επιλογή της ρύθμισης συνεχούς λειτουργίας του εργαλείου μετάφρασης γίνεται με την επιλογή ενός **check-box** στα **Search Options.**



Συμπεράσματα

Στην παρούσα πτυχιακή εργασία, επισημάνθηκε καταρχάς η ανάγκη ύπαρξης επαγγελματικών συστημάτων αναζήτησης, δεδομένου ότι οι κλασικές μηχανές αναζήτησης δεν μπορούν να ικανοποιήσουν πάντοτε τις απαιτήσεις του χρήστη. Στις περιπτώσεις όπου η ζητούμενη πληροφορία βρίσκεται στο ***Deep Web*** και σε εκείνες όπου η απλή αναπαράσταση των δεδομένων δεν είναι επαρκής, η χρήση επαγγελματικών συστημάτων αναζήτησης αποτελεί μονόδρομο για τη λειτουργία της αναζήτησης.

Επιπλέον, μελετήθηκε η πλατφόρμα ανάπτυξης διαδραστικών συστημάτων αναζήτησης ***ezDL*** και η αρχιτεκτονική στην οποία αυτή βασίζεται. Η αρχιτεκτονική του ezDL επιτρέπει την *επεκτασιμότητα* και την *παραμετροποίηση* του συστήματος, προσαρμόζοντάς το στις ανάγκες του εκάστοτε χρήστη. Η *επεκτασιμότητα* μπορεί να γίνει σε επίπεδο λειτουργιών και αναπαραστάσεων με την ενσωμάτωση εργαλείων στο *front-end* του συστήματος ezDL, αλλά και με την πρόσθεση νέων πηγών στο *back-end* του συστήματος. Η *παραμετροποίηση* του συστήματος μπορεί να γίνει με τον ορισμό κατηγοριών των πηγών και με την αλλαγή της λειτουργίας των *agents-wrappers*. Στην πρώτη περίπτωση, δίνεται η δυνατότητα συγκεκριμενοποίησης του πεδίου αναζήτησης για το οποίο προορίζεται το εκάστοτε σύστημα αναζήτησης. Με αυτό τον τρόπο, το ίδιο σύστημα αναζήτησης μπορεί να χρησιμοποιηθεί για αναζητήσεις σε τομείς όπως η *Ιατρική,* η *Βιβλιογραφία,* η *Αυτοκινητοβιομηχανία,* οι *Πατέντες κ.α.* Στην δεύτερη περίπτωση, δίνεται η δυνατότητα παραμετροποίησης των *agents-wrappers*. Έτσι, το σύστημα αναζήτησης ezDL εύκολα προσαρμόζεται στις αλλαγές που προκύπτουν στις αντίστοιχες υπηρεσίες αναζήτησης της κάθε πηγής.

Έπειτα, αναλύθηκε η διαδικασία ανάπτυξης ενός προγράμματος τύπου *agent-wrapper*, καθώς και η διαδικασία ανάπτυξης ενός *tool* του *front-end* του συστήματος. Και στις δύο περιπτώσεις περιγράφονται οι λειτουργικές και οι μη λειτουργικές απαιτήσεις που καταγράφηκαν κατά τη διαδικασία της ανάπτυξης του λογισμικού. Οι μη λειτουργικές απαιτήσεις αποτελούν υποκατηγορίες απαιτήσεων τύπου *Quality of Service,* οι οποίες, στην περίπτωσή μας, είναι ύψηλης προτεραιότητας.

Το πρόγραμμα ***Clef-IP IPC***, είναι ένας *agent-wrapper* εξειδικευμένος στην υποβολή ερωτημάτων και στην επιστροφή αποτελεσμάτων *IPCs* από μια συλλογή 3.1 εκατομμυρίων πατεντών. Η υπηρεσία στην οποία απευθύνεται ο *wrapper Clef-IP IPC* επεξεργάζεται τους όρους του υποβληθέντος ερωτήματος και, αφού αξιολογήσει τις πατέντες με τη χρήση ενός αλγορίθμου επιλογής (CORI), επιστρέφει τα IPCs και τον τίτλο της κάθε πατέντας. Στο κεφάλαιο αυτό παρουσιάζεται το διάγραμμα κλάσεων και η αρχιτεκτονική του *agent-wrapper Clef-IP IPC*. Επίσης, γίνεται επεξήγηση της κάθε λειτουργίας του wrapper και προσδιορίζεται η κλάση, στην οποία αυτή πραγματοποιείται.

Το εργαλείο μετάφρασης ***Translation Tool*** αποτελεί μια υλοποίηση ενός εργαλείου του *front-end.* Ο κύριος στόχος αυτού του εργαλείου είναι η μετάφραση των όρων ενός ερωτήματος που πρόκειται να υποβληθεί στις μηχανές αναζήτησης σε έναν αριθμό από γλώσσες. Αμέσως μετά τη διαδικασία μετάφρασης, οι μεταφρασμένοι όροι χρησιμοποιούνται στην σύνταξη ενός σύνθετου ερωτήματος με τη χρήση του τελεστή OR. Τέλος, το σύνθετο αυτό ερώτημα υποβάλλεται στις επιλεγμένες

υπηρεσίες του συστήματος αναζήτησης. Η ιδιαιτερότητα αυτού του εργαλείου έγκειται στη δυνατότητα της λειτουργίας του ως ένα ανεξάρτητο εργαλείο, αλλά και στην επιλογή της λειτουργίας του ως μέρος της συνολικής διαδικασίας της αναζήτησης. Λόγω της αρχιτεκτονικής του *front-end* του συστήματος *ezDL,* η πρόσθεση νέων εργαλείων και η δημιουργία επικοινωνίας τους με άλλα εργαλεία είναι μια εύκολη διαδικασία. Αυτό γίνεται άμεσα αντιληπτό από τη ροή λειτουργίας του εργαλείου μετάφρασης και από την αλληλεπίδρασή του με το υπόλοιπο σύστημα του *ezDL*.

Συμπερασματικά, το ***ezDL*** αποτελεί μια επαγγελματική λύση για οποιονδήποτε επιθυμεί να δημιουργήσει ένα σύστημα αναζήτησης. Η επέκτασή του σε επίπεδο πηγών αλλά και σε επίπεδο εργαλείων είναι μια εύκολη διαδικασία, αρκεί να γίνει από προγραμματιστές. Μια μελλοντική σκέψη είναι η ύπαρξη ενός εργαλείου το οποίο να επιτρέπει την προσθήκη νέων εργαλείων. Η προσθήκη ενός νέου εργαλείου θα μπορεί να γίνει με την υποβολή ενός αρχείου περιγραφής των μηνυμάτων που αυτό δέχεται και αποστέλλει. Αυτό βασίζεται στην ιδέα της γενικής λειτουργίας των εργαλείων και στην αρχιτεκτονική του *front-end* του συστήματος *ezDL.* Πολλά από τα εργαλεία θα μπορούσαν να αποτελούν διαδικτυακές εφαρμογές που βρίσκονται υλοποιημένες κάπου στο Internet και η χρήση των οποίων θα επέκτεινε την λειτουργικότητα του συστήματος αναζήτησης. Το μόνο που θα έμενε λοιπόν, θα ήταν ο ορισμός της επικοινωνίας τους με το υπόλοιπο σύστημα αναζήτησης του ezDL, πράγμα που θα μπορούσε να γίνει μέσω ενός αρχείου. Ομοίως, θα μπορούσαν να οριστούν νέες πηγές αναζήτησης και η αυτόματη δημιουργία *agents-wrappers*. Εφόσον αυτό που διαφοροποιεί τους *agent-wrappers* είναι η διεύθυνση της αντίστοιχης υπηρεσίας και η ρύθμιση του τρόπου εξαγωγής των ζητούμενων δεδομένων, αυτά θα μπορούσαν να οριστούν με τη υποβολή ενός αρχείου. Ίσως λοιπόν στο μέλλον η πρόσθεση νέων εργαλείων και νέων πηγών αναζήτησης να γίνεται πολύ πιο εύκολα, απλά με την χρήση ενός αρχείου.

Βιβλιογραφία

- **Randolph Hock.** The Extreme Searcher's Internet Handbook. *ISBN:0-910965-68-4*, Copyright © 2004 by Randolph E. Hock.
- **Randolph Hock.** Extreme Searcher's Guide to Web Search Engines *ISBN: 0910965471,* Copyright © 2001 by Randolph E. Hock.
- **Thomas Beckers, Sebastian Dungs, Norbert Fuhr, Matthias Jordan, Sascha Kriewel.** ezDL: An Interactive Search and Evaluation System, Daffodil 2013.
- **S. Kriewel, C.-P. Klas, A. Schaefer, and N. Fuhr.** Strategic support for user-oriented access to heterogeneous digital libraries, Daffodil 2004
- **Weiyi Meng.** Metasearch Engines, Department of Computer Science, State University of New York at Binghamton, Binghamton 2008.
- **Κουρουπέτρογλου Χρήστος.** Η σχεδίαση και η ανάπτυξη ενός Meta-search Engine. Τεχνολογικό Εκπαιδευτικό Ίδρυμα, Τμήμα Πληροφορικής, Θεσσαλονίκη 2002.
- **Steve Pederson.** Understanding the deep Web in 10 Minutes, BrightPlanet 2013.
- **Κωνσταντίνος Ντονάς.** Ταυτόχρονη αναζήτηση σε πολλαπλές πηγές δεδομένων με χρήση λογισμικού ανοιχτού κώδικα και εργαλείου εξαγωγής περιεχομένου απο ιστοσελίδες, Θεσσαλονίκη 2008.
- **Dong Nguyen, Thomas Demeester, Dolf Trieschnigg, Djoerd Hiemstra.** Federated Search in the Wild, 2012.
- **Devika K, Subu Surendran.** An Overview of Web Data Extraction Techniques, Department of Computer Science and Engineering, SCT College of Engineering, Kerala 2013.

Ιστοσελίδες

- Federated Search: The Options. http://www.searchtechnologies.com/federated-search.html
- Deep Web: A internet “Invisível”. http://patoatomico.com.br/deep-web-a-internet-invisivel/
- Meta-search Engine. http://en.wikipedia.org/wiki/Metasearch_engine
- Maven. http://maven.apache.org/
- Mercurial. http://mercurial.selenic.com/
- XPATH. http://en.wikipedia.org/wiki/XPath
- MySQL. http://www.mysql.com/