



ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Κατασκευή Μηχανής Αναζήτησης
για Διαδικτυακές Υπηρεσίες

Γεώργιος Μπατιστάτος

A.M. 20725

ΑΘΗΝΑ
ΣΕΠΤΕΜΒΡΙΟΣ 2011

ΧΑΡΟΚΟΠΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΜΑΤΙΚΗΣ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Κατασκευή Μηχανής Αναζήτησης για Διαδικτυακές Υπηρεσίες

Γεώργιος Μπατιστάτος

A.M. 20725

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Δημοσθένης Αναγνωστόπουλος, *Καθηγητής*

ΤΡΙΜΕΛΗΣ ΕΞΕΤΑΣΤΙΚΗ ΕΠΙΤΡΟΠΗ:

Δημοσθένης Αναγνωστόπουλος, *Καθηγητής*

Μαρία Νικολαΐδη, *Αναπληρώτρια Καθηγήτρια*

Ουρανία Χατζή, *Διδάσκουσα βάσει Π.Δ. 407/80*

Πρόλογος

Η εκπόνηση της παρούσας Πτυχιακής εργασίας πραγματοποιήθηκε στο Τμήμα Πληροφορικής και Τηλεματικής του Χαροκοπείου Πανεπιστημίου από τον Οκτώβριο του 2010 έως τον Οκτώβριο του 2011.

Επιθυμώ να εκφράσω τις ευχαριστίες μου σε όλους εκείνους που συνέβαλλαν άμεσα ή έμμεσα στην ολοκλήρωση της Πτυχιακής μου εργασίας και κατά συνέπεια των προπτυχιακών σπουδών μου.

Αρχικά, θα ήθελα να ευχαριστήσω τον εκλιπόντα Καθηγητή και πρώην Πρύτανη του Χαροκοπείου Πανεπιστημίου Γεώργιο Καραμπατζό. Το Τμήμα Πληροφορικής και Τηλεματικής αποτέλεσε δικό του όραμα, δίνοντας την ευκαιρία σε όλους εμάς τους φοιτητές να αποκτήσουμε υψηλού επιπέδου σπουδές σε ένα πρότυπο ακαδημαϊκό περιβάλλον.

Ιδιαίτερα θα ήθελα να ευχαριστήσω τον Πρύτανη του Χαροκοπείου Πανεπιστημίου και Πρόεδρο του Τμήματος Πληροφορικής και Τηλεματικής Καθηγητή Δημοσθένη Αναγνωστόπουλο, ο οποίος υπήρξε και ο επιβλέπων της παρούσας εργασίας. Παρά τις αυξημένες υποχρεώσεις του συνέβαλλε ποικιλοτρόπως στην ολοκλήρωση της εργασίας, ορίζοντας την κατεύθυνση που εκείνη ακολούθησε και παρέχοντας πολύτιμες συμβουλές όπου κρίθηκε αναγκαίο.

Θα ήθελα επίσης να εκφράσω την εκτίμηση μου στο πρόσωπο της Αναπληρώτριας Καθηγήτριας του Τμήματος Μάρας Νικολαΐδου για το αληθινό ενδιαφέρον και την απαιτητικότητα που επέδειξε ώστε η Πτυχιακή αυτή εργασία να εκπληρώσει το μέγιστο των δυνατοτήτων της, προτείνοντας ουσιαστικές διορθώσεις και εξασφαλίζοντας τις καλύτερες δυνατές συνθήκες εργασίας στο Πανεπιστήμιο.

Στο σημείο αυτό επιθυμώ να πω ένα μεγάλο ευχαριστώ στην διδάσκουσα βάσει του Π.Δ. 407/80 Ουρανία Χατζή, μέλος της τριμελούς επιτροπής της Πτυχιακής εργασίας, για την πολύτιμη βοήθεια που προσέφερε. Η υποστήριξη και διαθεσιμότητά της καθ' όλη την διάρκεια εκπόνησης της εργασίας αποτέλεσε σπουδαία βοήθεια, όπως και οι ουσιαστικές παρατηρήσεις και προτάσεις της ώστε να αντιμετωπισθούν τα προβλήματα που εμφανίσθηκαν.

Θα ήθελα ακόμα να ευχαριστήσω τον Λέκτορα Ηρακλή Βαρλάμη για την συνεισφορά του στην μετατροπή του web crawler σε πολυνηματικό και τις ενδιαφέρουσες ιδέες του, τον Λέκτορα

Κωνσταντίνο Τσερπέ για τις πολύτιμες συμβουλές του αναφορικά με την παρουσίαση της εργασίας, καθώς και τον υποψήφιο διδάκτορα Αντώνη Κόγια για τις πολύτιμες προγραμματιστικές παρατηρήσεις που μου έδωσε για βελτιστοποιήσεις του web crawler.

Επιπροσθέτως, θα ήθελα να ευχαριστήσω τους Βασίλη Δαλάκα, Γιώργο Συμιακάκη και Λορέτα Μήτση για την υπομονή τους κατά την διάρκεια της συνύπαρξής μας στο ίδιο γραφείο του Κέντρου Δικτύων αλλά και τους φίλους και συμφοιτητές μου Παναγιώτη Κατσιβέλη-Περάκη και Βαλεντίνο Χούδρα για την ιδιαίτερη υποστήριξη τους.

Τέλος, νιώθω την ανάγκη να ευχαριστήσω την οικογένεια μου για την αγάπη, κατανόηση και υποστήριξή τους, χωρίς τα οποία δεν θα ήταν δυνατή η ολοκλήρωση της παρούσας εργασίας.

Περίληψη

Οι διαδικτυακές υπηρεσίες αποτελούν την επικρατέστερη τεχνολογία για τη διαλειτουργικότητα μεταξύ ετερογενών συστημάτων, την κατασκευή διαδικτυακών εφαρμογών και γενικότερα την προσφορά λειτουργικότητας μέσω διαδικτύου. Στόχος της παρούσας πτυχιακής εργασίας είναι να συμβάλλει στην βελτίωση της ανακάλυψης και χρήσης διαδικτυακών υπηρεσιών μέσω της κατασκευής μιας μηχανής αναζήτησης εξειδικευμένης για περιγραφές διαδικτυακών υπηρεσιών.

Αρχικά, εξετάστηκαν οι υπάρχουσες μέθοδοι ανακάλυψης διαδικτυακών υπηρεσιών και διαπιστώθηκε πως δεν υπάρχει αποτελεσματικός μηχανισμός που να επιτρέπει την εύκολη ανακάλυψη τους. Πιο συγκεκριμένα, οι μηχανές αναζήτησης γενικού σκοπού παρουσιάζουν αδυναμίες στην επιτυχή ανάκληση και ακρίβεια των αποτελεσμάτων, ενώ οι προσπάθειες για μηχανές αναζήτησης στοχευμένες σε διαδικτυακές υπηρεσίες κρίνονται ανεπαρκείς, καθώς στην πλειοψηφία τους δεν παρέχουν υλοποίηση και δεν λαμβάνουν υπόψη τους τη σημασιολογία των διαδικτυακών υπηρεσιών.

Επίσης, στο πλαίσιο της θεωρητικής μελέτης, εξετάστηκαν τα πρότυπα περιγραφών διαδικτυακών υπηρεσιών WSDL και OWL-S και προσδιορίστηκαν τα στοιχεία τους εκείνα που έκαναν την αναζήτηση αποτελεσματική.

Όσον αφορά την υλοποίηση, στο πλαίσιο της παρούσας πτυχιακής εργασίας κατασκευάστηκε μια μηχανή αναζήτησης για διαδικτυακές υπηρεσίες που αποτελείται από έναν web crawler, τον μηχανισμό ευρετηριοποίησης και την γραφική της διεπαφή.

Ο web crawler αναπτύχθηκε σταδιακά και υπέστη μια σειρά βελτιστοποιήσεων, όπως για παράδειγμα η μετατροπή του σε πολυνηματικό, η χρήση του δίσκου για ενδιάμεση αποθήκευση, και η αποφυγή αρχείων κατά την αναζήτηση, ώστε να είναι δυνατή η αποδοτική εκτέλεσή του σε συμβατικά συστήματα με περιορισμένους πόρους. Επιπλέον, αναπτύχθηκαν και εφαρμόστηκαν ευριστικοί κανόνες οι οποίοι κατέστησαν την λειτουργία της συλλογής περιγραφών διαδικτυακών υπηρεσιών από τον Παγκόσμιο Ιστό αποδοτικότερη, οι οποίοι περιλαμβάνουν την τοπική αλλαγή του αλγόριθμου αναζήτησης και την ευριστική δημιουργία λίστας αποφυγής. Η λειτουργία του είχε σαν αποτέλεσμα τον εντοπισμό μέχρι στιγμής περίπου 1500 περιγραφών διαδικτυακών υπηρεσιών.

Ο μηχανισμός ευρετηριοποίησης και αναζήτησης των περιγραφών που χρησιμοποιήθηκε βασίστηκε στο ανεστραμμένο ευρετήριο. Για την εύρεση της εγγύτητας – ομοιότητας μεταξύ των περιγραφών διαδικτυακών υπηρεσιών που συγκεντρώθηκαν και της επερώτησης του χρήστη χρησιμοποιείται το διανυσματικό μοντέλο και η μετρική tf-idf. Το μοντέλο προσαρμόστηκε στην περίπτωση των διαδικτυακών υπηρεσιών με τη χρήση βαρών, έτσι ώστε να δίνει ορθότερα αποτελέσματα και να είναι δυνατή η εύρεση περιγραφών όχι μόνο με βάση απλές λέξεις κλειδιά αλλά με βάση επιπλέον χαρακτηριστικά που περιέχονται στις περιγραφές.

Η γραφική διεπαφή παρέχει τη δυνατότητα απλής αναζήτησης, μέσω λέξεων-κλειδιών, καθώς και τη δυνατότητα σύνθετων αναζητήσεων, στις οποίες οι χρήστες μπορούν να υποδείξουν και το τμήμα της περιγραφής μιας διαδικτυακής υπηρεσίας στο οποίο επιθυμούν να βρίσκονται οι λέξεις-κλειδιά. Η γραφική διεπαφή συνδέεται με τον μηχανισμό ευρετηριοποίησης και απάντησης επερωτήσεων μέσω της τεχνολογίας των διαδικτυακών υπηρεσιών.

Η κατασκευή της μηχανής αναζήτησης ολοκληρώθηκε με πειράματα και περιπτώσεις χρήσης, ώστε να υποστηριχθεί η ορθή λειτουργία της και να επιδειχθεί η χρησιμότητα και αποτελεσματικότητά της.

Abstract

The web service standard has become widely accepted, as it enables and supports interoperability between heterogenous systems and facilitates the building of web applications and the exposure of functionality through the Internet. The goal of the present BSc Thesis is the design and creation of a web service specific search engine, which will accommodate web service discovery and usability, thus contributing to the further establishment of the standard.

The review of available web service search and discovery methods revealed that current efforts are not effective enough. In particular, regular search engines were insufficient for web service search, returning results with low precision and recall, whereas, most efforts for a web service specific search engine were inadequate due to the lack of a working implementation. Furthermore, current web service specific search engines do not fully exploit the semantic information available in web service descriptions.

Additionally, the web service description standards WSDL and OWL-S were examined, in order to determine and locate those elements that are significant for search and produce better search results.

The conclusion of the theoretical part was followed by the search engine design and implementation. The web service specific search engine consists of a web crawler, an indexing mechanism and a web interface.

The development of the web crawler included a number of optimizations such as modifications to support multiple threads, use of the hard disk drive to extend the search frontier, and the exclusion of certain content type during search, which enable it to be executed on a conventional computer. Moreover, the web crawler incorporates heuristic rules, such as the local change of the search algorithm from breadth-first to depth-first and the heuristic creation of the exclusion list, which improve the process of locating web service descriptions. The web crawler has so far discovered about 1500 web service descriptions.

Regarding the indexing mechanism, the structure used to index the located web service descriptions is an inverted index. The method used to retrieve these web service descriptions is a combination of the tf-idf method and the vector space model. Moreover, the tf-idf method was extended using weights to optimize its efficiency and support for web service descriptions.

The web interface permits the interaction between the system and its users, and has been designed to offer two search methods. The default search method corresponds to the conventional keyword search accustomed to typical search engines. The advanced search offers extended capabilities such as enabling the user to search for a keyword on a specific part of the web service description. The connection between the web interface and the indexing mechanism is accomplished using a web service.

The system implementation was followed by a number of experiments and use cases to establish its functionality and performance.

Περιεχόμενα

ΠΡΟΛΟΓΟΣ	3
ΠΕΡΙΛΗΨΗ	5
ABSTRACT	7
ΠΕΡΙΕΧΟΜΕΝΑ	9
ΚΕΦΑΛΑΙΟ 1 ΕΙΣΑΓΩΓΗ.....	12
1.1. ΔΙΑΔΙΚΤΥΑΚΕΣ ΥΠΗΡΕΣΙΕΣ	13
1.2. ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΗΣ	16
1.2.1. Ιστορικά στοιχεία	16
1.2.2. Τρόπος λειτουργίας	17
1.2.3. Συλλογή.....	18
1.2.4. Ευρετηριοποίηση	19
1.2.5. Αναζήτηση.....	20
1.2.6. Προκλήσεις	20
1.3. ΣΤΟΧΟΣ	21
1.4. ΔΟΜΗ	21
ΚΕΦΑΛΑΙΟ 2 ΒΙΒΛΙΟΓΡΑΦΙΚΗ ΕΠΙΣΚΟΠΗΣΗ	23
2.1. ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΗΣ ΓΕΝΙΚΟΥ ΣΚΟΠΟΥ	23
2.1.1. Google	23
2.1.2. Bing	24
2.1.3. Yahoo!.....	25
2.1.4. Λοιπές	26
2.1.5. Χρήση μηχανών αναζήτησης γενικού σκοπού για την αναζήτηση διαδικτυακών υπηρεσιών.....	26
2.2. ΕΞΕΙΔΙΚΕΥΜΕΝΕΣ ΜΗΧΑΝΕΣ ΑΝΑΖΗΤΗΣΗΣ ΓΙΑ ΔΙΑΔΙΚΤΥΑΚΕΣ ΥΠΗΡΕΣΙΕΣ.....	28
2.2.1. Seekda.....	28
2.2.2. Woogler.....	30
2.2.3. WSExpress (QoS-aware Search Engine for Web Services).....	31
2.2.4. VUSE (VitaLab Universal WS Search Engine)	33
2.2.5. SWSE (Semantic Web Search Engine)	33
2.2.6. OSSE (Ontology-Based Prototype Service Search Engine).....	34
2.2.7. Σύνοψη.....	34
ΚΕΦΑΛΑΙΟ 3 ΠΡΟΤΥΠΑ ΠΕΡΙΓΡΑΦΗΣ ΔΙΑΔΙΚΤΥΑΚΩΝ ΥΠΗΡΕΣΙΩΝ	36
3.1. WSDL	36
3.1.1. Περιγραφή	36
3.1.2. Παραδείγματα	41
3.1.2.1. reservationList.wsdl.....	41
3.1.2.2. GreatH.wsdl	42
3.1.3. Αναζήτηση.....	44
3.2. SAWSDL	45
3.2.1. Περιγραφή	46
3.2.2. Παράδειγμα	48
3.2.3. Αναζήτηση.....	49
3.3. OWL-S	49

3.3.1.	Περιγραφή	49
3.3.1.1.	Service Profile	51
3.3.1.2.	Service Model	54
3.3.1.3.	Service Grounding	55
3.3.2.	Παράδειγμα	56
3.3.2.1.	ISBNtoBookAuthor.owl	57
3.3.2.2.	books.owl	57
3.3.2.3.	portal.owl	58
3.3.3.	Αναζήτηση.....	59
ΚΕΦΑΛΑΙΟ 4 ΣΧΕΔΙΑΣΜΟΣ ΜΗΧΑΝΗΣ ΑΝΑΖΗΤΗΣΗΣ		61
4.1.	ΠΕΡΙΓΡΑΦΗ ΣΥΣΤΗΜΑΤΟΣ.....	61
4.1.1.	Στόχος	61
4.1.2.	Αρχιτεκτονική Συστήματος	63
4.2.	WEB CRAWLER	65
4.2.1.	Πολιτικές.....	67
4.2.1.1.	Επιλογή (Selection).....	67
4.2.1.2.	Ευγένεια (Politeness)	68
4.2.1.3.	Επανεπίσκεψη (Revisit)	69
4.2.1.4.	Παραλληλοποίηση (Parallelization)	70
4.2.2.	Ρυθμίσεις (Configuration).....	70
4.3.	PARSING	71
4.4.	INDEXING	73
4.5.	WEB INTERFACE	75
ΚΕΦΑΛΑΙΟ 5 ΥΛΟΠΟΙΗΣΗ ΜΗΧΑΝΗΣ ΑΝΑΖΗΤΗΣΗΣ		77
5.1.	WEB CRAWLER	77
5.1.1.	Βασική λειτουργικότητα	77
5.1.2.	Πρόσθετα Χαρακτηριστικά	78
5.1.2.1.	Προσδιορισμός περιγραφών διαδικτυακών υπηρεσιών	79
5.1.2.2.	Αποφυγή αρχείων	80
5.1.2.3.	Πολυνηματική υλοποίηση	81
5.1.2.4.	Διαχείριση και ελαχιστοποίηση χρήσης μνήμης	82
5.1.2.5.	Διαχωρισμός μετώπου αναζήτησης σε αρχεία	83
5.1.2.6.	Συμπίεση	84
5.1.2.7.	Περιορισμός κλειστού συνόλου	85
5.1.2.8.	Αποθήκευση κατάστασης.....	85
5.1.2.9.	Ευριστικός κανόνας για τα domain και λίστα αποφυγής	86
5.1.2.10.	Ευριστικός κανόνας για εύρεση περιγραφών WSDL από περιγραφές OWL-S.....	86
5.1.2.11.	Τοπική αλλαγή αλγορίθμου αναζήτησης (Breadth-to-depth heuristic)	86
5.1.2.12.	Προγραμματιστικές Βελτιστοποιήσεις	87
5.1.2.13.	Ρυθμίσεις	88
5.1.3.	Υποστηριζόμενη Λειτουργία	89
5.2.	PARSING	98
5.2.1.	Βασική λειτουργικότητα	98
5.2.2.	Πρόσθετα Χαρακτηριστικά	98
5.2.2.1.	Στοιχεία αναζήτησης	99
5.2.2.2.	Επιπλοκές περιγραφών	99
5.2.2.3.	Αρχεία εξόδου	99
5.3.	INDEX.....	99
5.3.1.	Βασική λειτουργικότητα	100
5.3.2.	Πρόσθετα Χαρακτηριστικά	100

5.3.2.1.	Ρυθμίσεις	100
5.3.2.2.	Parsing	102
5.3.2.3.	Επεξεργασία δεδομένων	102
5.3.2.4.	Αφαίρεση αριθμητικών ψηφίων και σημείων στίξης	102
5.3.2.5.	Αφαίρεση camelCase και splitting.....	102
5.3.2.6.	Case normalization	103
5.3.2.7.	Αφαίρεση stop words.....	103
5.3.2.8.	Εξαγωγή ρίζας όρων.....	105
5.3.2.9.	Ευρετήριο	105
5.3.2.10.	TF-IDF	106
5.3.2.11.	Επιπλέον βάρη	107
5.3.2.12.	Εγγύτητα επερώτησης – εγγράφου.....	107
5.3.2.13.	Απλή αναζήτηση.....	108
5.3.2.14.	Σύνθετη αναζήτηση	108
5.3.2.15.	Επικοινωνία Index – Web Interface.....	109
5.4.	WEB INTERFACE	109
5.4.1.	Βασική λειτουργικότητα	109
5.4.2.	Πρόσθετα Χαρακτηριστικά	110
5.4.2.1.	Απλή αναζήτηση.....	110
5.4.2.2.	Σύνθετη αναζήτηση	110
5.4.2.3.	Άλλοι σύνδεσμοι	110
5.4.2.4.	Εμφάνιση.....	110
5.5.	ΤΕΧΝΟΛΟΓΙΕΣ, ΓΛΩΣΣΕΣ ΚΑΙ ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ	111
ΚΕΦΑΛΑΙΟ 6 ΈΛΕΓΧΟΣ ΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑΣ, ΑΠΟΔΟΣΗ ΚΑΙ ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ		114
6.1.	ΠΕΡΙΒΑΛΛΟΝ ΚΑΙ ΔΙΑΔΙΚΑΣΙΑ ΕΚΤΕΛΕΣΗΣ ΜΗΧΑΝΗΣ ΑΝΑΖΗΤΗΣΗΣ.....	114
6.2.	ΔΙΑΔΙΚΑΣΙΑ ΠΕΙΡΑΜΑΤΙΣΜΟΥ - ΑΠΟΔΟΣΗ.....	115
6.3.	ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ	118
6.3.1.	Απλή αναζήτηση	118
6.3.2.	Σύνθετη αναζήτηση	121
ΚΕΦΑΛΑΙΟ 7 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΚΑΤΕΥΘΥΝΣΕΙΣ		128
ΚΕΦΑΛΑΙΟ 7 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΕΣ ΚΑΤΕΥΘΥΝΣΕΙΣ		128
7.1.	ΣΥΜΠΕΡΑΣΜΑΤΑ	128
7.2.	ΜΕΛΛΟΝΤΙΚΕΣ ΚΑΤΕΥΘΥΝΣΕΙΣ	131
ΒΙΒΛΙΟΓΡΑΦΙΑ		133

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Τα τελευταία χρόνια το Διαδίκτυο (Internet) εξελίσσεται και αλλάζει σημαντικά. Μεταβαίνει από μία συλλογή κειμένων και πληροφορίας σε ένα μέσο πολύ περισσότερο αλληλεπιδραστικό, επίκεντρο του οποίου αποτελούν οι ίδιοι οι χρήστες [23]. Η έλευση του Σημασιολογικού Διαδικτύου (Semantic Web) έχει συντελέσει ιδιαίτερα προς αυτή την κατεύθυνση [10].

Ο χρήστης, πέρα από την δυνατότητα να λαμβάνει παθητικά πληροφορίες, καθίσταται πλέον ικανός να μετάσχει πιο ενεργητικά, να δημιουργήσει πληροφορία, όπως για παράδειγμα γίνεται στα κοινωνικά δίκτυα (social networks) και τα ιστολόγια (blogs), αλλά και να αλληλεπιδράσει με το Διαδίκτυο σε ένα υψηλό επίπεδο, λαμβάνοντας υπηρεσίες στις οποίες μέχρι τώρα είχε πρόσβαση μόνο με την φυσική παρουσία του στους παρόχους των υπηρεσιών αυτών, π.χ. καταστήματα, κρατικές υπηρεσίες, οργανισμούς.

Ήδη ένα μεγάλο μέρος των υπηρεσιών μέσω διαδικτύου υλοποιείται χρησιμοποιώντας το πρότυπο των διαδικτυακών υπηρεσιών (web services) [2]. Το πρότυπο αυτό έχει στόχο την τυποποίηση και διευκόλυνση της περιγραφής, του εντοπισμού και της επικοινωνίας ανάμεσα σε συστήματα που παρέχουν την λειτουργικότητά τους στο Διαδίκτυο με τη μορφή υπηρεσίας [14]. Για παράδειγμα, δύο υπηρεσίες που παρέχονται στο Διαδίκτυο είναι οι υπηρεσίες ηλεκτρονικής τραπεζικής (e-banking), όπου ο χρήστης μπορεί να δει τα υπόλοιπα των λογαριασμών του, αλλά και να εκτελέσει μεταφορές ποσών και πληρωμές σε πραγματικό χρόνο, καθώς και η αγορά αεροπορικών εισιτηρίων προς οποιοδήποτε προορισμό, χωρίς τη απαραίτητη μεσολάβηση κάποιου ταξιδιωτικού πρακτορείου. Τέτοιου είδους υπηρεσίες υλοποιούνται χρησιμοποιώντας το πρότυπο των διαδικτυακών υπηρεσιών, έτσι ώστε να προάγεται η τυποποιημένη χρήση και η διαλειτουργικότητά τους.

Ο τεράστιος όγκος των πληροφοριών που παρέχονται στο Διαδίκτυο δημιούργησε την ανάγκη για μηχανές αναζήτησης (search engines) όπως το Google, το Yahoo! κλπ., οι οποίες διευκολύνουν την περιήγηση στον ιστό και τον εντοπισμό της κατάλληλης πληροφορίας σε εύλογο χρονικό διάστημα και χωρίς ιδιαίτερη προσπάθεια από τον χρήστη [41].

Κατά αντιστοιχία, ο συνεχώς αυξανόμενος αριθμός διαδικτυακών υπηρεσιών δημιουργεί την ανάγκη για αποδοτικό και ορθό εντοπισμό της λειτουργικότητας-υπηρεσίας που επιζητεί ο χρήστης κάθε φορά. Οι υπάρχουσες μηχανές αναζήτησης γενικού σκοπού αποδεικνύονται

ανεπαρκείς σε αυτή την ανάγκη. Για αυτό το λόγο, δημιουργήθηκαν εξειδικευμένες μηχανές αναζήτησης, στοχευμένες στις διαδικτυακές υπηρεσίες [61].

Κατά τη διάρκεια των τελευταίων χρόνων έχουν γίνει προσπάθειες για την ανάπτυξη τέτοιων μηχανών αναζήτησης για διαδικτυακές υπηρεσίες. Ωστόσο, επειδή η αναζήτηση διαδικτυακών υπηρεσιών είναι ένας σχετικά νέος τομέας, υπάρχουν ακόμα σημαντικά περιθώρια βελτίωσης και η έρευνα είναι ενεργή.

Η εξάπλωση και η βελτίωση μηχανών αναζήτησης για διαδικτυακές υπηρεσίες με την πάροδο του χρόνου, αναμένεται να συμβάλλει σημαντικά στη διευκόλυνση των χρηστών και την εξάπλωση και διείσδυση των υπηρεσιών μέσω Διαδικτύου στην κοινωνία [32].

1.1. Διαδικτυακές Υπηρεσίες

Οι διαδικτυακές υπηρεσίες (web services) σήμερα αποτελούν δομικά συστατικά στοιχεία του Παγκόσμιου Ιστού (World Wide Web), καθώς αυτός τείνει να μετατραπεί από μια απλή συλλογή κειμένων σε ένα ολοκληρωμένο περιβάλλον, σημαντικό τμήμα του οποίου σχετίζεται με την προσφορά υπηρεσιών [33]. Για την επιτυχημένη αλληλεπίδραση των πληροφοριακών συστημάτων στον Παγκόσμιο Ιστό είναι αναγκαία η επικοινωνία και η συνεργασία ανάμεσα σε ετερογενή συστήματα που βασίζονται σε διαφορετικές τεχνολογίες. Οι διαδικτυακές υπηρεσίες στοχεύουν στην απλοποίηση τέτοιου είδους αλληλεπιδράσεων, ορίζοντας έναν πρότυπο μηχανισμό για την περιγραφή, τον εντοπισμό και την επικοινωνία μεταξύ πληροφοριακών συστημάτων που εκτίθενται και λειτουργούν στο Διαδίκτυο (Internet). Με τη χρήση της τεχνολογίας αυτής, είναι εφικτός ο μετασχηματισμός και η έκθεση οποιασδήποτε εφαρμογής ως διαδικτυακής, στην οποία μπορούν να έχουν πρόσβαση άλλα συστήματα χρησιμοποιώντας ανοικτά και σαφώς ορισμένα πρότυπα που έχουν συσταθεί από το World Wide Web Consortium [81]. Σύμφωνα με το μοντέλο των διαδικτυακών υπηρεσιών, ο Παγκόσμιος Ιστός λειτουργεί ως ενδιάμεσο λογισμικό για την ολοκλήρωση ετερογενών εφαρμογών και περιβαλλόντων.

Υπάρχουν τρεις τρόποι σύμφωνα με τους οποίους οι διαδικτυακές υπηρεσίες είναι δυνατό να χρησιμοποιηθούν [80]:

- *Remote procedure calls*

Οι διαδικτυακές υπηρεσίες RPC ουσιαστικά αποτελούν διεπαφές κλήσεων συναρτήσεων, στις οποίες καθοριστικό ρόλο διαδραματίζουν οι λειτουργίες των περιγραφών των υπηρεσιών.

- *Service-oriented architecture*

Οι διαδικτυακές υπηρεσίες μπορούν επίσης να χρησιμοποιηθούν στο πλαίσιο μιας αρχιτεκτονικής προσανατολισμένης στις υπηρεσίες (service-oriented architecture). Σε αυτή την περίπτωση βασικό στοιχείο της επικοινωνίας είναι τα μηνύματα και όχι οι λειτουργίες των υπηρεσιών.

- *Representational state transfer (REST)*

Τέλος, μπορούν να χρησιμοποιηθούν χρησιμοποιώντας την αρχιτεκτονική REST σύμφωνα με την οποία η διεπαφή περιορίζεται στην χρήση κάποιων καθιερωμένων πρωτοκόλλων και λειτουργιών. Στην περίπτωση αυτή οι υπηρεσίες εστιάζουν σε stateful πόρους αντί για τα μηνύματα ή τις λειτουργίες.

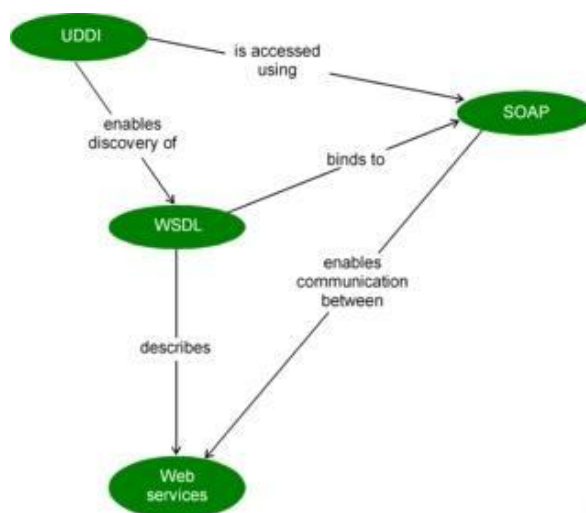
Η πρόσβαση σε κάποια διαδικτυακή υπηρεσία επιτυγχάνεται μέσω ενός Ενιαίου Προσδιοριστή Πόρων (Universal Resource Identifier - URI), ενώ η ίδια η υπηρεσία ορίζεται και περιγράφεται από γλώσσες βασισμένες στην XML, όπως η Web Service Description Language (WSDL) [86]. Η WSDL περιγράφει τις διεπαφές της διαδικτυακής υπηρεσίας, δηλαδή τις πληροφορίες οι οποίες θα πρέπει να είναι ορατές και διαθέσιμες ώστε να πραγματοποιηθεί η επικοινωνία μαζί της, αλλά περιορίζεται μόνο στη συντακτική τους περιγραφή. Η ίδια η διαδικτυακή υπηρεσία αντιμετωπίζεται ως μια απομακρυσμένη υλοποίηση αυτών των διεπαφών, που είναι ανεξάρτητη από πλατφόρμα υλοποίησης.

Οι διαδικτυακές υπηρεσίες μπορούν να δημοσιευθούν και να ανακαλυφθούν μέσω μητρώων UDDI (Universal Description, Discovery and Integration) [75]. Τα μητρώα UDDI παρέχουν μια πρότυπη πλατφόρμα που επιτρέπει τον εντοπισμό διαδικτυακών υπηρεσιών με γρήγορο, εύκολο και δυναμικό τρόπο. Τα μητρώα UDDI περιέχουν εγγραφές των διαθέσιμων διαδικτυακών υπηρεσιών και παρέχουν τεχνικές πληροφορίες για τον τρόπο πρόσβασης και χρήσης των υπηρεσιών, όπως επίσης και κατηγοριοποίηση των διαδικτυακών υπηρεσιών.

Χρήστες, συστήματα λογισμικού ή ευφυείς πράκτορες (intelligent agents) που εντοπίζουν μια διαδικτυακή υπηρεσία μέσω ενός μητρώου UDDI μπορούν να αλληλεπιδράσουν με αυτήν χρησιμοποιώντας το πρωτόκολλο SOAP (Simple Object Access Protocol) [69]. Τα μηνύματα

SOAP μεταφέρονται μέσω πρωτοκόλλων μεταφοράς (transfer protocols) στο Διαδίκτυο όπως το HTTP (Hyper Text Transfer Protocol) [37]. Το SOAP βασίζεται στην XML και παρέχει το βασικό πλαίσιο για τη σύνταξη μηνυμάτων που χρησιμοποιούνται για την ανακάλυψη και την επικοινωνία με τις διαδικτυακές υπηρεσίες.

Οι αλληλεπιδράσεις μεταξύ των προαναφερθέντων προτύπων συνοψίζονται στην Εικόνα 1.



Εικόνα 1. Αλληλεπιδράσεις μεταξύ των προτύπων WSDL, UDDI και SOAP [70].

Το τρέχον πρότυπο περιγραφής διαδικτυακών υπηρεσιών WSDL λειτουργεί καθαρά στο συντακτικό επίπεδο και επομένως σε αρκετές περιπτώσεις δεν είναι σε θέση να αποτυπώσει τις απαιτήσεις και τις δυνατότητες μιας διαδικτυακής υπηρεσίας, για παράδειγμα τις εισόδους και τις εξόδους της, με τρόπο κατανοητό τόσο από τον άνθρωπο όσο και από τον υπολογιστή. Η προσθήκη σημασιολογικών (semantic) πληροφοριών στην περιγραφή των διαδικτυακών υπηρεσιών έχει ως αποτελέσματα βελτιωμένες δυνατότητες ανακάλυψής τους, και τη διευκόλυνση της αυτοματοποίησης διαδικασιών σχετικών με τη διαχείριση των διαδικτυακών υπηρεσιών, όπως για παράδειγμα τη σύνθεση, την εκτέλεση και τη συνεργασία. Επιπλέον, μπορεί να διευκολύνει τη μετατροπή υπαρχόντων εφαρμογών σε διαδικτυακές υπηρεσίες, με απώτερο στόχο την ολοκλήρωση επιχειρησιακών διαδικασιών (business process integration) [57].

Για την κάλυψη της ανάγκης προσθήκης σημασιολογικών πληροφοριών προτάθηκαν νέες γλώσσες περιγραφής διαδικτυακών υπηρεσιών, όπως η SAWSDL (Semantic Annotations for

Web Service Description Language) [65] και η WSDL-S (Web Service Description Language - Semantics) [87], που στην πλειοψηφία τους είναι συμβατές και λειτουργούν συμπληρωματικά με τα τρέχοντα πρότυπα για συντακτική περιγραφή των υπηρεσιών. Πέραν από τις σημασιολογικές πληροφορίες που εισάγονται με αυτές τις γλώσσες, η σημασιολογική επέκταση των διαδικτυακών υπηρεσιών, πρέπει να υποστηριχθεί από οντολογίες (ontologies). Οι οντολογίες εμπλουτίζουν τις σημασιολογικές πληροφορίες απεικονίζοντας τις συσχετίσεις μεταξύ των εννοιών που χρησιμοποιούνται για να αποδώσουν τη σημασιολογία των διαδικτυακών υπηρεσιών. Η OWL-S [53], γνωστή ως DAML-S στις πρώτες εκδόσεις της, είναι μια οντολογία για διαδικτυακές υπηρεσίες βασισμένη στην OWL (Web Ontology Language) [52], η οποία παρέχει ένα σύνολο γλωσσικών δομών επισήμανσης (markup language constructs) για τη σημασιολογική περιγραφή των ιδιοτήτων και δυνατοτήτων των διαδικτυακών υπηρεσιών. Η OWL-S έχει επικρατήσει ως πρότυπο για τη σημασιολογική περιγραφή διαδικτυακών υπηρεσιών.

1.2. Μηχανές Αναζήτησης

Ο Παγκόσμιος Ιστός (World Wide Web) [9] θα μπορούσε να περιγραφεί αφαιρετικά σαν ένας τεράστιος όγκος πληροφοριών που καθημερινά αλλάζει και επεκτείνεται. Η πληροφορία όμως δεν έχει κάποια ιδιαίτερη χρησιμότητα όταν απλά παράγεται, χρειάζεται να είναι προσβάσιμη από τους χρήστες. Οι μηχανές αναζήτησης είναι αυτές που δίνουν αξία στην πληροφορία και καθιστούν τον Παγκόσμιο Ιστό τόσο χρήσιμο και ευρέως χρησιμοποιούμενο σήμερα, παρέχοντας στον χρήστη την δυνατότητα να περιηγηθεί σύμφωνα με τις ανάγκες και τις προτιμήσεις του.

1.2.1. Ιστορικά στοιχεία

Η ιστορία των μηχανών αναζήτησης είναι συνδεδεμένη με την ιστορία του Διαδικτύου. Πριν εμφανισθεί ο Παγκόσμιος Ιστός, για την ανταλλαγή και τον διαμοιρασμό δεδομένων στο Διαδίκτυο χρησιμοποιούνταν το πρωτόκολλο μεταφοράς αρχείων (File Transfer Protocol) [27]. Η ανάγκη για αναζήτηση ήταν ακόμη και τότε εμφανής. Οι πρώτες προσπάθειες αφορούσαν καταλόγους των διαθέσιμων διακομιστών αλλά σύντομα εμφανίσθηκε η μηχανή αναζήτησης που θεωρείτε ο πρόγονος των μεταγενέστερων.

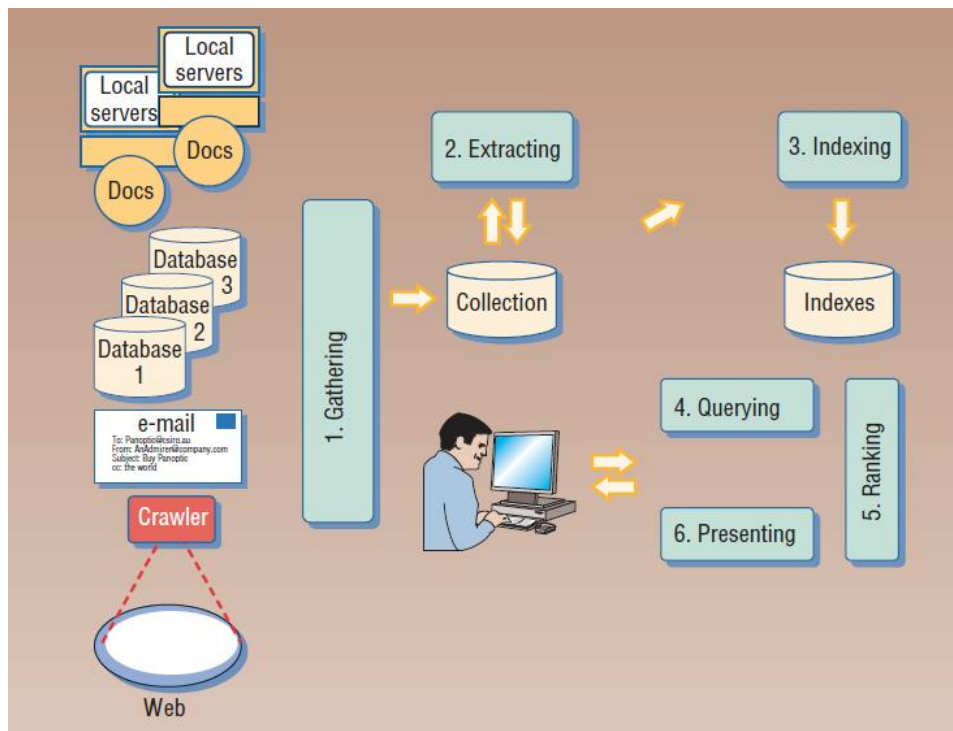
Ήταν το 1990 οπότε δημιουργήθηκε η μηχανή αναζήτησης Archie [36]. Η λειτουργία της βασιζόταν στην αποθήκευση των λιστών με τα αρχεία που υπήρχαν στους δημόσιους διακομιστές FTP, δημιουργώντας έτσι μια βάση δεδομένων με τα ονόματα των διαθέσιμων αρχείων στην οποία μπορούσαν να πραγματοποιηθούν αναζητήσεις. Την μηχανή αναζήτησης Archie ακολούθησαν οι Veronica και Jughead οι οποίες λειτουργούσαν με τον ίδιο τρόπο εκμεταλλευόμενες όμως το νέο τότε πρωτόκολλο Gopher [74].

Η πρώτη μηχανή αναζήτησης που ακολουθούσε την δομή που χρησιμοποιούν οι σημερινές λεγόταν JumpStation. Κατά τα σημερινά πρότυπα χρησιμοποιούσε έναν web crawler που έβρισκε τις σελίδες τις οποίες στην συνέχεια τοποθετούσε σε ένα ευρετήριο. Τελικά, έδινε την δυνατότητα στους χρήστες να πραγματοποιήσουν επερωτήσεις χρησιμοποιώντας μια φόρμα. Το μειονέκτημα της ήταν πως ευρετηριοποιούσε μόνο τους τίτλους των σελίδων. Για να φτάσουμε στο σημερινό πρότυπο όπου όλα τα δεδομένα των ιστοσελίδων ευρετηριοποιούνται, το βήμα έκανε η μηχανή αναζήτησης WebCrawler [83] που εμφανίστηκε το 1994 και χρησιμοποιούσε όλα τα δεδομένα των ιστοσελίδων. Τα χρόνια που ακολούθησαν εμφανίσθηκαν πολλές νέες μηχανές αναζήτησης όπως οι Lycos [45], Inktomi [38], AltaVista [3], MSN Search [50] κλπ.

Η επόμενη μεγάλη αλλαγή οφειλόταν σε μια καινοτομία της μηχανής αναζήτησης Google [31] που εμφανίστηκε το 1998. Για την βελτίωση των αποτελεσμάτων της αναζήτησης, η Google χρησιμοποιούσε τον αλγόριθμο PageRank [67] ο οποίος έδινε ιδιαίτερη βαρύτητα στους συνδέσμους κάθε σελίδας. Η χρήση του βασιζόταν στο σκεπτικό πως οι καλές σελίδες δείχνονται από πολλές άλλες με συνδέσμους. Χάρη σε αυτή και άλλες καινοτομίες η μηχανή αναζήτησης Google είναι σήμερα η πλέον χρησιμοποιούμενη [20] και η σελίδα με τις περισσότερες επισκέψεις παγκοσμίως [29].

1.2.2. Τρόπος λειτουργίας

Όπως αναφέρθηκε νωρίτερα, οι περισσότερες μηχανές αναζήτησης σήμερα λειτουργούν σε τρία επίπεδα. Η πρώτη απαραίτητη ενέργεια είναι η συλλογή των δεδομένων των ιστοσελίδων. Ακολουθείται από την τοποθέτηση των δεδομένων αυτών σε ένα ευρετήριο, το οποίο τελικά αποκρίνεται στις αναζητήσεις των χρηστών. Η καθιερωμένη αρχιτεκτονική για μια μηχανή αναζήτησης φαίνεται στην Εικόνα 2 [34].



Εικόνα 2. Η αρχιτεκτονική μιας μηχανής αναζήτησης.

1.2.3. Συλλογή

Η συλλογή των ιστοσελίδων που τελικά θα ευρετηριοποιηθούν πραγματοποιείται χρησιμοποιώντας προγράμματα που διασχίζουν τον Παγκόσμιο Ιστό με αυτοματοποιημένο και μεθοδικό τρόπο τα οποία καλούνται web crawlers.

Ο web crawler [77] επιτυγχάνει αυτή την λειτουργία διατηρώντας μιας ουρά με τις ιστοσελίδες που πρόκειται να επισκεφθεί και χρησιμοποιώντας μια μέθοδο που του επιτρέπει να αποφεύγει ιστοσελίδες που έχει ήδη επισκεφθεί. Ξεκινά την λειτουργία του από κάποιες ιστοσελίδες σπόρους ενώ στην συνέχεια ανατροφοδοτείται από την ουρά των προς επίσκεψη σελίδων. Για κάθε σελίδα, αφού αποκτήσει πρόσβαση στο περιεχόμενό της, το αποθηκεύει και στην συνέχεια εντοπίζει όλους τους συνδέσμους που υπάρχουν σε αυτή τοποθετώντας τους στην ουρά με τις προς επίσκεψη ιστοσελίδες. Η λειτουργία του συνεχίζεται μέχρι η ουρά να αδειάσει.

Στην πράξη ο αλγόριθμος λειτουργίας ενός web crawler είναι αρκετά πιο σύνθετος για διάφορους λόγους. Δεδομένου πως υπάρχουν διάφοροι περιορισμοί όπως το διαθέσιμο εύρος ζώνης, το υλικό, ο όγκος του Παγκόσμιου Ιστού κλπ. είναι απαραίτητο ο web crawler να

λειτουργεί με αποδοτικό τρόπο που θα εξασφαλίζει και την ποιότητα της προσφερόμενης υπηρεσίας. [25]

Για τον λόγο αυτό είναι θεμιτό να λειτουργεί με βάση κάποιες ξεκάθαρες πολιτικές που αφορούν [16]:

- την επιλεκτικότητα (selection), που καθορίζει ποιές ιστοσελίδες θα κατεβάζει και ποιές θα αποφεύγει. Οι τελευταίες μπορούν να είναι διπλότυπα, spam, παγίδες ειδικά για crawlers ή απλά σελίδες χωρίς χρήσιμο περιεχόμενο,
- την επανεπίσκεψη (re-visit), που ορίζει πότε θα επανεξετάζονται ιστοσελίδες για τυχόν αλλαγές στο περιεχόμενο,
- την ευγένεια (politeness) που διευκρινίζει τον τρόπο σύμφωνα με τον οποίο αποφεύγεται η αθέμιτη υπερφόρτωση των διακομιστών των επισκεπτόμενων ιστοσελίδων, και
- την παραλληλοποίηση (parallelization), που ορίζει πως θα συντονίζεται η λειτουργία ανάμεσα σε καταναμημένους web crawlers.

1.2.4. Ευρετηριοποίηση

Τα δεδομένα που συνέλλεξε ο web crawler είναι αυτά που θα χρησιμοποιηθούν για να αποκριθεί η μηχανή αναζήτηση στις αναζητήσεις των χρηστών. Για την αποθήκευση τους χρησιμοποιούνται ευρετήρια [63], στόχος των οποίων είναι η βελτιστοποίηση της απόδοσης και της ταχύτητας με την οποία ανακτώνται τα σχετικά με την επερώτηση του χρήστη δεδομένα.

Ο συνηθέστερος τύπος ευρετηρίου που χρησιμοποιείται στις μηχανές αναζήτησης είναι το ανεστραμμένο ευρετήριο [40], το οποίο συμβάλλει σημαντικά στην επίτευξη του στόχου, αφού διευκολύνει τον γρήγορο εντοπισμό των εγγράφων που περιέχουν τους όρους της επερώτησης.

Βέβαια, τα δεδομένα δεν τοποθετούνται αυτούσια από τις ιστοσελίδες στο ευρετήριο. Πριν την ευρετηριοποίηση περνούν διάφορα στάδια επεξεργασίας ώστε να μετατραπούν σε μορφή τέτοια που διευκολύνει την ευρετηριοποίηση, επιταχύνοντας την διαδικασία και μειώνοντας τον κατακερματισμό του ευρετηρίου.

Η αποδοτική λειτουργία ενός ευρετηρίου αποτελεί πρόκληση καθώς υπάρχουν πολλοί παράγοντες που είναι ανάγκη να ληφθούν υπόψη κατά τον σχεδιασμό ενός τέτοιου συστήματος. Μερικοί από αυτούς είναι:

- το μέγεθος του ευρετηρίου και οι τεχνικές αποθήκευσης που χρησιμοποιούνται,
- ο χρόνος απόκρισης του ευρετηρίου,
- η συντήρηση και η αξιοπιστία του [18][7], και
- η αναβάθμιση-εμπλουτισμός του με νέα δεδομένα.

1.2.5. Αναζήτηση

Η τελική φάση της λειτουργίας μιας μηχανής αναζήτησης η είναι αλληλεπίδραση με τον χρήστη. Η αλληλεπίδραση αυτή πραγματοποιείται διαμέσου μιας γραφικής διεπαφής στην οποία ο χρήστης έχει την δυνατότητα να εισάγει τα αντικείμενα της αναζήτησης του με την μορφή ερωτήσεων.

Αφού ο χρήστης εισάγει μια ερώτηση, εκείνη συσχετίζεται με τα περιεχόμενα του ευρετηρίου και τελικά επιλέγονται και παρουσιάζονται στον χρήστη τα αποτελέσματα της αναζήτησης (hits). Τα αποτελέσματα αυτά σχετίζονται με το περιεχόμενο της ερώτησης και τον αλγόριθμο που χρησιμοποιεί κάθε μηχανή αναζήτησης για τον εντοπισμό των καλύτερων αποτελεσμάτων και την ταξινόμηση τους. Οι αλγόριθμοι αυτοί αποτελούν την ουσιαστική διαφορά ανάμεσα στις διάφορες μηχανές αναζήτησης, γεγονός που τους κάνει να φυλλάσσονται αυστηρά σαν εταιρικά μυστικά που είναι.

1.2.6. Προκλήσεις

Όπως προκύπτει από τα επιμέρους τμήματα της, η φύση του έργου που επιτελεί μια μηχανή αναζήτησης καθιστά την ανάπτυξή της ιδιαίτερα επίπονη. Για την επίτευξη της ευχαίρειας κλιμάκωσης και της ανοχής σε σφάλματα που απαιτούνται, οι σύγχρονες ευρέως χρησιμοποιούμενες μηχανές αναζήτησης χρησιμοποιούν γεωγραφικά κατανομημένα κέντρα δεδομένων (data centers) [78].

Στα κέντρα αυτά, υπάρχουν υπολογιστικές συστάδες (clusters) ή ξεχωριστοί διακομιστές που χρησιμοποιούνται αποκλειστικά για την διεκπαιρέωση εξειδικευμένων εργασιών όπως το crawling, η ευρετηριοποίηση, ο χειρισμός των ερωτήσεων, η εύρεση των αποτελεσμάτων κλπ [34].

Για να γίνουν κατανοητά τα μεγέθη, σήμερα, τα δεδομένα του Παγκόσμιου Ιστού που υπόκεινται crawling και ευρετηριοποίηση είναι της τάξεως των 400 terabytes, τοποθετώντας έτσι τεράστιο φόρτον στους διακομιστές και το δίκτυο [34].

1.3. Στόχος

Ο στόχος της παρούσας πτυχιακής εργασίας είναι η κατασκευή μιας μηχανής αναζήτησης εξειδικευμένης για περιγραφές διαδικτυακών υπηρεσιών. Στο πλαίσιο της κατασκευής αυτής μελετάται το θεωρητικό υπόβαθρο για την αναζήτηση διαδικτυακών υπηρεσιών καθώς και οι ερευνητικές προσπάθειες που έχουν πραγματοποιηθεί, ενώ υλοποιούνται όλα τα τμήματα μιας σύγχρονης μηχανής αναζήτησης, δηλαδή ένας web crawler, το ευρετήριο και η γραφική διεπαφή της μηχανής αναζήτησης. Η ολοκλήρωση της προσπάθειας της παρούσας εργασίας ακολουθείται από την παροχή της συγκεκριμένης υπηρεσίας στο Διαδίκτυο χρησιμοποιώντας υποδομές του Χαροκοπείου Πανεπιστημίου.

Το βασικό στοιχείο που καθοδηγεί τον στόχο αυτό είναι η πίστη πως το πρότυπο των διαδικτυακών υπηρεσιών, με τα πλεονεκτήματα που το συνοδεύουν μπορεί να αποτελέσει ένα πολύ ισχυρό εργαλείο για την κατασκευή διαδικτυακών εφαρμογών. Διαπιστώνοντας πως οι δομές των προτυποποιημένων περιγραφών διαδικτυακών υπηρεσιών επιδέχονται αναζήτηση και οι υπάρχουσες μέθοδοι αναζήτησης τέτοιων υπηρεσιών δεν είναι αποτελεσματικές, θεωρήθηκε πως η συγκεκριμένη προσπάθεια θα μπορούσε να προχωρήσει το πρόβλημα της ανακάλυψης υπηρεσιών ένα βήμα παραπέρα ώστε να συμβάλλει στην εδραίωση του προτύπου.

1.4. Δομή

Η παρούσα πτυχιακή εργασία είναι δομημένη ως εξής:

Στο Κεφάλαιο 1 παρουσιάζονται εισαγωγικά στοιχεία και ορισμοί σχετικά με τις διαδικτυακές υπηρεσίες και τις μηχανές αναζήτησης. Επίσης παρουσιάζεται ο στόχος της εργασίας και τα βασικά στοιχεία του συστήματος που αναπτύχθηκε.

Στο Κεφάλαιο 2 αρχικά εξετάζονται οι μηχανές αναζήτησης γενικού σκοπού ως προς την αναζήτηση διαδικτυακών υπηρεσιών και στην συνέχεια η έρευνα και οι προσπάθειες που έχουν πραγματοποιηθεί για την δημιουργία εξειδικευμένων μηχανών αναζήτησης για διαδικτυακές

υπηρεσίες. Με βάση τα στοιχεία αυτά εξάγονται συμπεράσματα για μελλοντική ενασχόληση με τον τομέα της αναζήτησης διαδικτυακών υπηρεσιών.

Στο Κεφάλαιο 3 περιγράφονται αναλυτικά τα πλέον επικρατή πρότυπα διαδικτυακών υπηρεσιών και προσδιορίζονται τα στοιχεία τους που περιέχουν χρήσιμη πληροφορία για αναζητήσεις.

Στο Κεφάλαιο 4 πραγματοποιείται ο σχεδιασμός του συστήματος. Παρουσιάζεται η αρχιτεκτονική της μηχανής αναζήτησης για διαδικτυακές υπηρεσίες, τα συστατικά στοιχεία της, οι βασικές λειτουργικές απαιτήσεις του καθενός από αυτά, καθώς και οι προκλήσεις που πρέπει να αντιμετωπιστούν για την επιτυχή λειτουργία του συστήματος.

Στο Κεφάλαιο 5 περιγράφεται η υλοποίηση της μηχανής αναζήτησης. Για το κάθε υποσύστημα της παρουσιάζεται η βασική λειτουργικότητά του, τα ιδιαίτερα χαρακτηριστικά που αναπτύχθηκαν κατά την υλοποίησή του, και οι τεχνολογίες, εργαλεία και περιβάλλοντα ανάπτυξης που χρησιμοποιήθηκαν. Επιπλέον, για κάθε υποσύστημα παρουσιάζεται και περιγράφεται το διάγραμμα λειτουργίας του.

Στο Κεφάλαιο 6 παρουσιάζονται τα αποτελέσματα περιπτώσεων χρήσης της μηχανής αναζήτησης για διαδικτυακές υπηρεσίες ώστε να διαπιστωθεί η αν ικανοποιεί τις προδιαγραφές που τέθηκαν κατά τον σχεδιασμό της.

Στο Κεφάλαιο 7 συνοψίζεται η πτυχιακή εργασία που πραγματοποιήθηκε. Παρουσιάζονται τα συμπεράσματα που εξήχθησαν και οι μελλοντικές κατευθύνσεις που θα μπορούσε να ακολουθήσει.

Βιβλιογραφική Επισκόπηση

Σκοπός του παρόντος Κεφαλαίου είναι η συνοπτική παρουσίαση των επικρατέστερων μηχανών αναζήτησης γενικού σκοπού, αφού αυτές μπορούν να χρησιμοποιηθούν για αναζήτηση διαδικτυακών υπηρεσιών, ενώ επιπλέον αποτελούν την βάση για όλες τις εξειδικευμένες μηχανές αναζήτησης, καθώς και η περιγραφή της ερευνητικής δραστηριότητας σχετικής με μηχανές αναζήτησης στοχευμένες στις διαδικτυακές υπηρεσίες, έτσι ώστε να αποτυπωθεί η παρούσα κατάσταση και πιθανές μελλοντικές κατευθύνσεις στον τομέα αυτό.

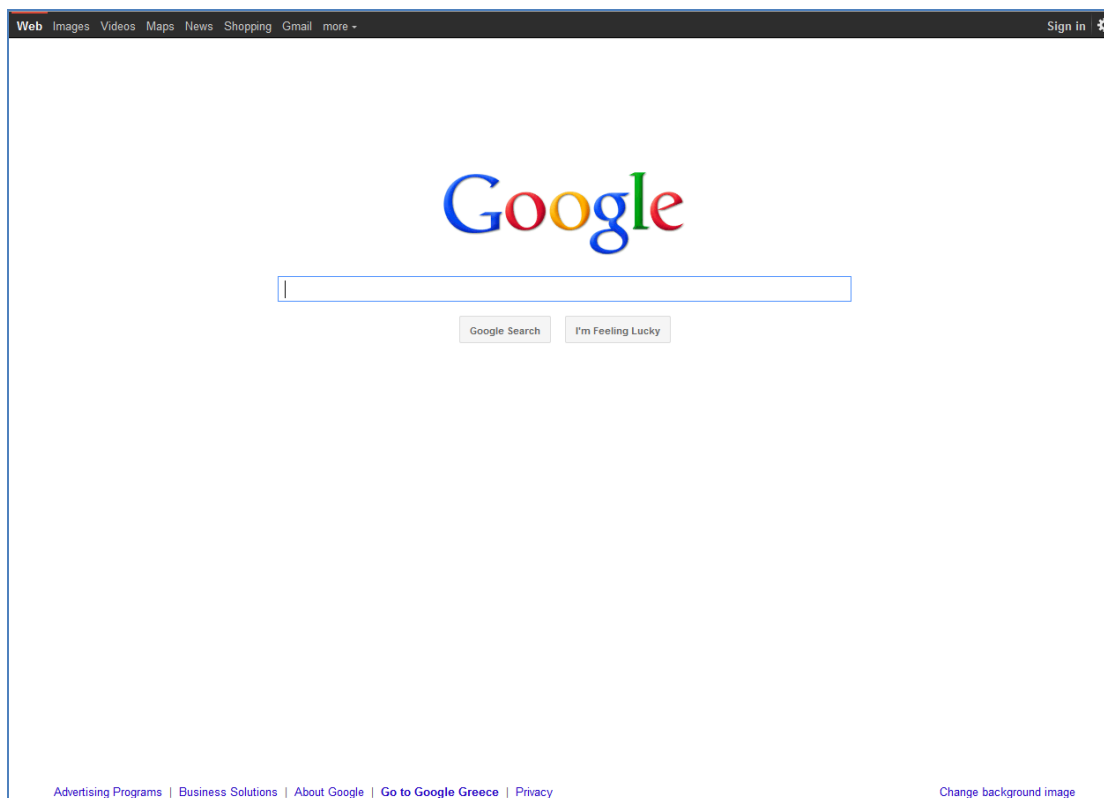
2.1. Μηχανές Αναζήτησης Γενικού Σκοπού

Η αναζήτηση οποιασδήποτε πληροφορίας στον Παγκόσμιο Ιστό, συμπεριλαμβανομένου και των διαδικτυακών υπηρεσιών, είναι εφικτή χρησιμοποιώντας μηχανές αναζήτησης γενικού σκοπού, οι επικρατέστερες των οποίων παρουσιάζονται συνοπτικά παρακάτω.

2.1.1. Google

Το Google [31] είναι αυτή τη στιγμή η πιο συχνά χρησιμοποιούμενη μηχανή αναζήτησης στον κόσμο [1]. Η προεπιλεγμένη μορφή αναζήτησης στην συγκεκριμένη μηχανή πραγματοποιείται χρησιμοποιώντας λέξεις-κλειδιά (keywords), ενώ παρέχει πολλούς ακόμη τρόπους αναζήτησης όπως η αναζήτηση με την χρήση εικόνων (image search), η φωνητική αναζήτηση (voice search), η χρήση λογικών τελεστών στις ερωτήσεις κ.α. Η επιτυχία του Google οφείλεται σε μεγάλο βαθμό στην χρήση του αλγορίθμου PageRank [67] για την βαθμονόμηση των αποτελεσμάτων που επιστρέφει στους χρήστες. Πέρα από την συχνότητα εμφάνισης των όρων της ερώτησης σε μια ιστοσελίδα, ο συγκεκριμένος αλγόριθμος δίνει ιδιαίτερη σημασία στην σημαντικότητα και την συνάφεια των υπερσυνδέσμων που δείχνουν σε αυτή. Η βαθμονόμηση αυτή σήμερα πραγματοποιείται χρησιμοποιώντας περισσότερα από 200 διαφορετικά κριτήρια ώστε να εξασφαλίζεται η ποιότητα των αποτελεσμάτων [30]. Για την ευρετηριοποίηση των ιστοσελίδων, το Google χρησιμοποιεί προγράμματα συλλογής και ανάκτησης ιστοσελίδων από το Διαδίκτυο (crawlers), τα οποία συλλέγουν σελίδες επί 24ώρου βάσεως. Ένα ακόμη στοιχείο που βοήθησε το Google να καθιερωθεί ήταν η πρωτοποριακή απλότητα της ιστοσελίδας του την εποχή που

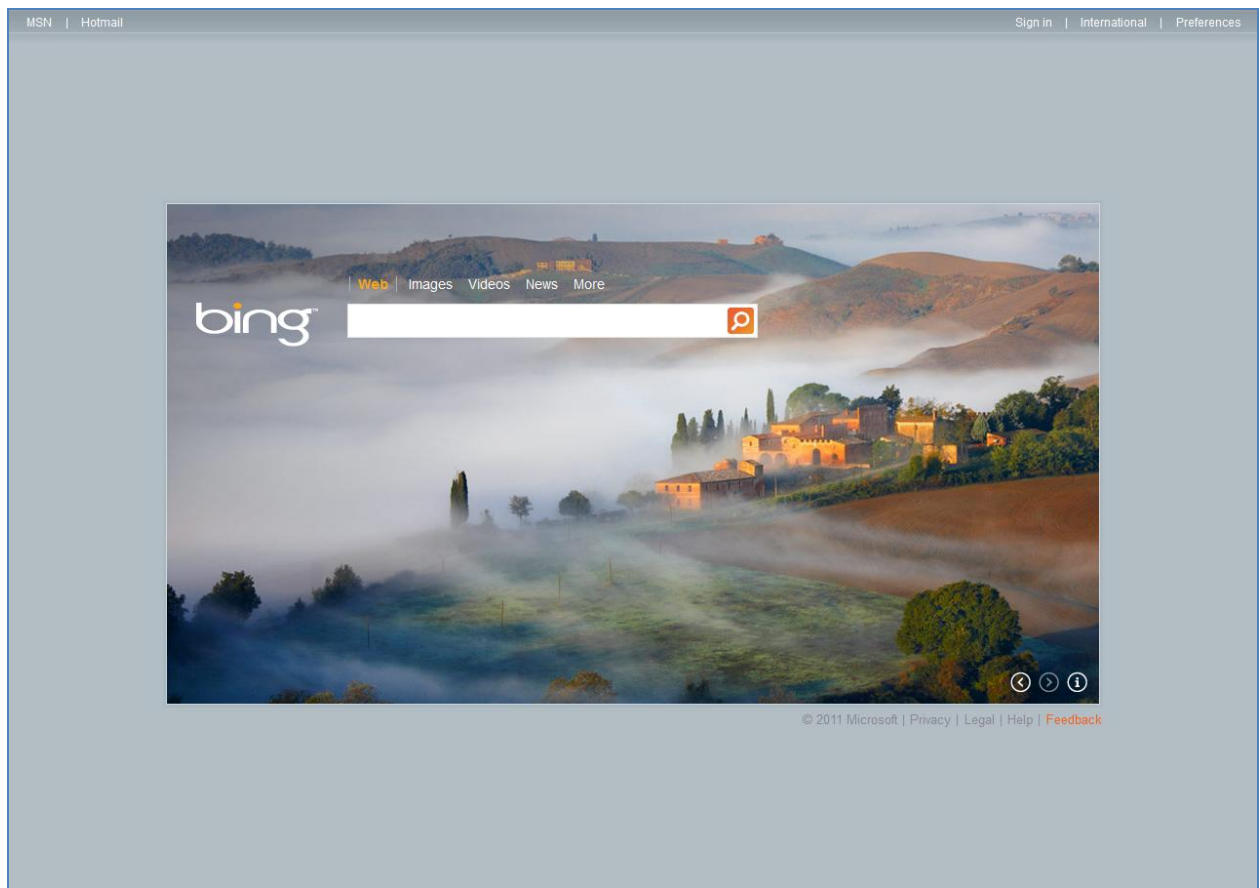
ξεκίνησε. Ο τρόπος που έχει διαμορφωθεί η ιστοσελίδα σήμερα φαίνεται στην Εικόνα 3 που ακολουθεί.



Εικόνα 3. Ο διαδικτυακός τόπος της μηχανής αναζήτησης της Google.

2.1.2. Bing

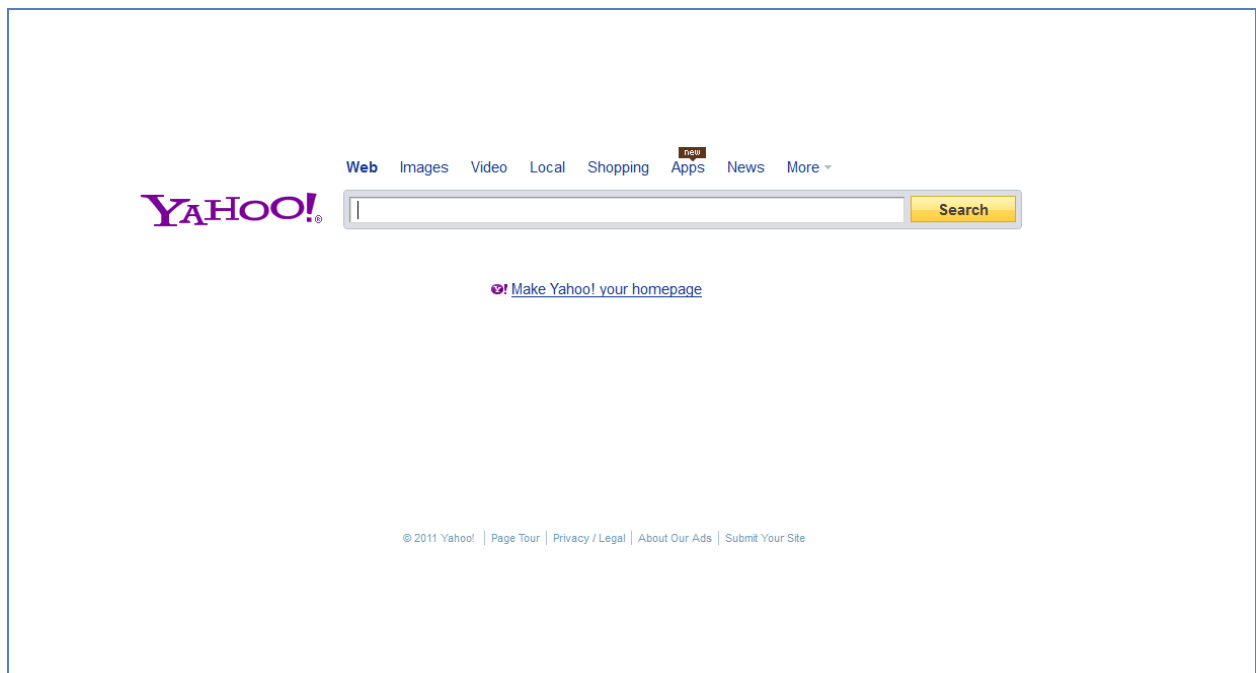
Σημαντικό μερίδιο της αγοράς κατέχει και η μηχανή αναζήτησης Bing [12], η οποία αποτελεί την τελευταία προσπάθεια της Microsoft στον χώρο της αναζήτησης μετά το MSN Search, το Windows Live Search και το Live Search. Ο τρόπος λειτουργίας της – όπως και των περισσότερων μηχανών αναζήτησης – είναι παρεμφερής με εκείνον της Google. Η βασική τους διαφορά έγκειται στον αλγόριθμο που χρησιμοποιείται για την αξιολόγηση και βαθμονόμηση των ιστοσελίδων. Οι υπηρεσίες αναζήτησης που παρέχει καλύπτουν διαφορετικά αντικείμενα όπως η αναζήτηση εικόνων, βίντεο κλπ. Πλέον, με το μερίδιο αγοράς που του αναλογεί να αυξάνεται αργά αλλά σταθερά αποτελεί τον σημαντικότερο ανταγωνιστή της μηχανής αναζήτησης της Google. Ο διαδικτυακός τόπος της συγκεκριμένης μηχανής αναζήτησης φαίνεται παρακάτω στην Εικόνα 4.



Εικόνα 4. Ο διαδικτυακός τόπος της μηχανής αναζήτησης της Bing.

2.1.3. Yahoo!

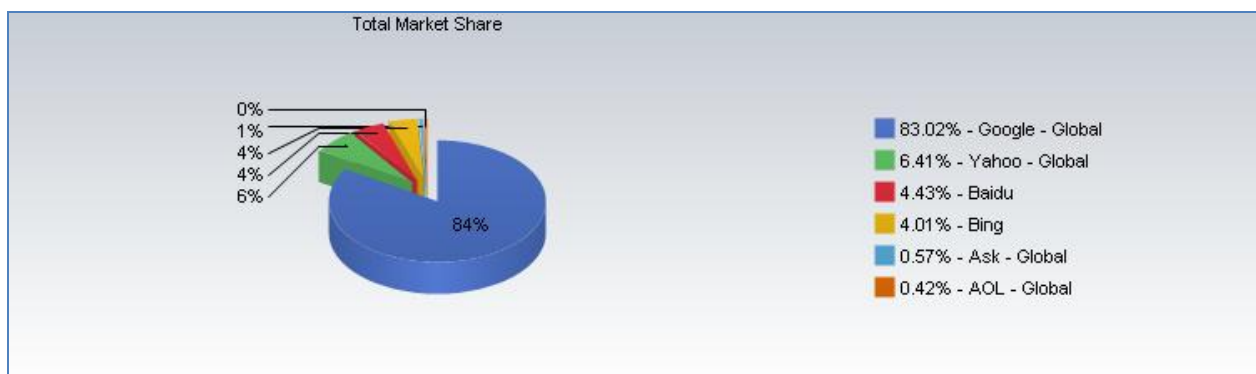
Η πρώτη απόπειρα της Yahoo! [90] στην αναζήτηση αποτελούσε έναν κατάλογο ιστοσελίδων οργανωμένων με ιεραρχικό τρόπο. Η αλλαγή στην μορφή μιας συμβατικής μηχανής αναζήτησης πραγματοποιήθηκε στα τέλη της δεκαετίας του 90. Ακόμα και τότε όμως η Yahoo! δεν χρησιμοποιούσε δικό της ευρετήριο για τα αποτελέσματα, συνεργαζόμενη αρχικά με την εταιρία Inktomi και στην συνέχεια με την Google. Ήταν μόλις το 2003 οπότε η μηχανή αναζήτησης της Yahoo χρησιμοποίησε τον δικό της web crawler και το δικό της ευρετήριο. Από το 2009 τα αποτελέσματα της μηχανής αναζήτησης παρέχει το Bing. Ακολουθεί στην Εικόνα 5 η ιστοσελίδα του Yahoo! Search.



Εικόνα 5. Ο διαδικτυακός τόπος της μηχανής αναζήτησης της Yahoo!.

2.1.4. Λοιπές

Άλλες μηχανές αναζήτησης με σημαντικά μικρότερη χρήση είναι οι Baidu [8], Yandex [91], Blekko [13], DuckDuckGo [24], Lycos [118], AOL [4] και Ask [6]. Το μερίδιο αγοράς κάθε μηχανής αναζήτησης φαίνεται αναλυτικά στην Εικόνα 6 [21].



Εικόνα 6. Παγκόσμιο μερίδιο αγοράς ανά μηχανή αναζήτησης.

2.1.5. Χρήση μηχανών αναζήτησης γενικού σκοπού για την αναζήτηση διαδικτυακών υπηρεσιών

Η αναζήτηση διαδικτυακών υπηρεσιών μέσω μηχανών αναζήτησης γενικού σκοπού είναι εφικτή με χρήση των κατάλληλων λέξεων-κλειδιών. Η αναζήτηση αυτή επιτελείται πάνω σε ένα τεράστιο όγκο δεδομένων, για αυτό το λόγο συνοδεύεται από μια σειρά μειονεκτημάτων που την καθιστούν αναποτελεσματική:

- Όντας μη εξειδικευμένες, οι μηχανές αναζήτησης γενικού σκοπού επιστρέφουν στον χρήστη πολύ μεγάλο αριθμό αποτελεσμάτων, εκ των οποίων η πλειοψηφία δεν αφορά περιγραφές διαδικτυακών υπηρεσιών, ακόμα και αν προστεθούν οι σχετικοί όροι σαν λέξεις-κλειδιά στην αναζήτηση. Ενδεικτικά αναφέρεται ότι μόλις το 12% των επιστρεφόμενων αποτελεσμάτων συνοδεύονται από WSDL αρχεία με τυποποιημένες περιγραφές διαδικτυακών υπηρεσιών [44]. Ο χρήστης δεν είναι πάντα σε θέση να γνωρίζει ακριβώς τι θέλει εκ των προτέρων ώστε να περιορίσει αυτό το φαινόμενο, οπότε τελικά καλείται να αφιερώσει πολύ χρόνο εξερευνώντας τα επιστρεφόμενα αποτελέσματα, χωρίς εγγύηση ότι θα εντοπίσει αυτό που έψαχνε.
- Οι τυπικές μηχανές αναζήτησης αντιμετωπίζουν τις ιστοσελίδες λεκτικά, δηλαδή όπως προκύπτει από την λεκτική ανάλυση των όρων οι οποίοι περιέχονται σε αυτές. Η συγκεκριμένη προσέγγιση όμως δεν είναι η καταλληλότερη για διαδικτυακές υπηρεσίες, καθώς τα δομικά στοιχεία ενός αρχείου όπως είναι οι επισημάνσεις (tags) για μια περιγραφή WSDL, εμφανίζονται πολλές φορές μέσα στο ίδιο έγγραφο, και επομένως αποκτούν μεγαλύτερη βαρύτητα απ' ό τι θα έπρεπε, επηρεάζοντας αρνητικά τόσο την ακρίβεια όσο και την ορθότητα των επιστρεφόμενων αποτελεσμάτων.
- Ο χρήστης δεν έχει την δυνατότητα να παραμετροποιήσει επαρκώς την αναζήτησή του όταν πρόκειται για διαδικτυακές υπηρεσίες. Συγκεκριμένα, δεν μπορεί να υποδείξει στη μηχανή αναζήτησης το τμήμα της περιγραφής της υπηρεσίας στο οποίο επιθυμεί να βρίσκονται συγκεκριμένα στοιχεία. Για παράδειγμα, έστω ότι ο χρήστης αναζητεί μια διαδικτυακή υπηρεσία που δέχεται σαν είσοδο ένα τηλέφωνο και επιστρέφει την διεύθυνση του κατόχου του. Μια μηχανή αναζήτησης γενικού σκοπού θα επέστρεφε σαν αποτέλεσμα, μεταξύ άλλων, διαδικτυακές υπηρεσίες όπου οι όροι «τηλέφωνο» και «διεύθυνση» θα αναφέρονταν απλά στην τεκμηρίωση ή θα ήταν αντίστροφα, δηλαδή θα είχαν την διεύθυνση σαν είσοδο και το τηλέφωνο σαν έξοδο κ.ο.κ., καθιστώντας την αναζήτηση αναποτελεσματική. Επίσης, δεν είναι δυνατός ο προσδιορισμός επιπρόσθετων χαρακτηριστικών που καθορίζουν την ποιότητα υπηρεσίας που προσφέρεται, αφού οι συγκεκριμένες πληροφορίες δεν είναι δυνατόν να

αναπαρασταθούν μόνο μέσω απλών λέξεων-κλειδιών που αναπαριστούν χαρακτηριστικά ποιότητας, αλλά απαιτούνται ζεύγη χαρακτηριστικού και τιμής.

Ως «χρήσιμες» περιγραφές υπηρεσιών ορίζονται οι περιγραφές εκείνες που ανταποκρίνονται στην ανάγκη του χρήστη, όπως αυτή εκφράστηκε μέσω μιας επερώτησης που τέθηκε στην μηχανή αναζήτησης. Τα δύο πρώτα σημεία επηρεάζουν την *ανάκληση (recall)* της μηχανής αναζήτησης, δηλαδή τον λόγο του αριθμού των χρήσιμων περιγραφών υπηρεσιών που επιστρέφονται τελικά προς τον συνολικό αριθμό των χρήσιμων υπηρεσιών που υπάρχουν και θα έπρεπε να επιστρέφονται στην ιδανική περίπτωση. Το τρίτο σημείο επηρεάζει την *ακρίβεια (precision)* της μηχανής αναζήτησης, που αντιστοιχεί στον λόγο του αριθμού των χρήσιμων περιγραφών υπηρεσιών που επιστράφηκαν προς τον συνολικό αριθμό εγγράφων που παρουσιάστηκαν σαν αποτελέσματα της συγκεκριμένης αναζήτησης [7][48].

2.2. Εξειδικευμένες Μηχανές Αναζήτησης για Διαδικτυακές Υπηρεσίες

Η αναζήτηση διαδικτυακών υπηρεσιών, όπως έχει παρουσιαστεί μέχρι στιγμής στη βιβλιογραφία και σε συστήματα που βρίσκονται σε λειτουργία, πραγματοποιείται είτε χρησιμοποιώντας μητρώα (registries / repositories) είτε με λέξεις κλειδιά [56]. Η παρούσα ενότητα περιγράφει τις μηχανές αναζήτησης διαδικτυακών υπηρεσιών που έχουν βρεθεί στη βιβλιογραφία.

2.2.1. Seekda

Η Seekda [64] είναι μια εταιρία που συνδυάζει τις επιχειρηματικές προτάσεις της με τις τελευταίες τεχνολογικές εξελίξεις στον τομέα των διαδικτυακών υπηρεσιών. Στο πλαίσιο αυτής της προσπάθειας διατηρεί μια μηχανή αναζήτησης για διαδικτυακές υπηρεσίες. Η δημιουργία της μηχανής προηγήθηκε της ίδρυσης της εταιρίας και αποτελεί έργο ενός εκ των ιδρυτών της. Μέχρι σήμερα έχει ανακαλύψει 28535 διαδικτυακές υπηρεσίες προερχόμενες από 7690 παρόχους. Σκοπός της εταιρίας είναι η μηχανή να στηρίζεται στην αυτοματοποίηση των διαδικασιών και στην εξόρυξη σημασιολογικής πληροφορίας από υπάρχουσες πληροφορίες.

Τα παραπάνω επιτυγχάνονται με μια σειρά από βήματα. Αρχικά, πραγματοποιείται στοχευμένη αναζήτηση (crawling) στον παγκόσμιο ιστό με σκοπό την εύρεση περιγραφών WSDL

διαδικτυακών υπηρεσιών χρησιμοποιώντας τον Heritrix Web Crawler [35]. Η στοχευμένη αναζήτηση επιτυγχάνεται περιορίζοντας τον χώρο αναζήτησης του crawler και παρέχοντας σαν σημεία εκκίνησης της αναζήτησης (seeds) επιλεγμένες σελίδες. Στην συγκεκριμένη περίπτωση οι σελίδες αυτές ήταν είτε γνωστές σελίδες που παρείχαν ή περιείχαν υπερσυνδέσμους (links) σε διαδικτυακές υπηρεσίες, είτε σελίδες που βρέθηκαν με την βοήθεια της μηχανής αναζήτησης γενικού σκοπού Alexa.

Οι περιγραφές που συγκεντρώνονται στη συνέχεια περνούν ένα στάδιο ανάλυσης. Ιδιαίτερη έμφαση δίνεται στο σημείο που αναφέρονται τα τερματικά (endpoints) της υπηρεσίας, δηλαδή οι διαδικτυακές διευθύνσεις στις οποίες μια συγκεκριμένη υπηρεσία είναι διαθέσιμη, αφού όσες δεν δείχνουν σε κάποια έγκυρη διεύθυνση αφαιρούνται. Στην συνέχεια, οι περιγραφές των υπηρεσιών συσχετίζονται με επιπλέον σημασιολογική πληροφορία και αφού υποστούν επεξεργασία ώστε να βρεθούν σε μορφή που να βελτιστοποιεί τις λεκτικές αναζητήσεις, αποθηκεύονται. Η λεκτική αυτή επεξεργασία περιλαμβάνει στάδια όπως είναι ο διαχωρισμός ενωμένων λέξεων, η αντιστοίχιση κάθε λέξης με έναν πρωτογενή όρο και ο προσδιορισμός ενός βάρους που καθορίζεται από τη θέση της στην περιγραφή για καθεμία.

Κατά την αναζήτηση με χρήση λέξεων-κλειδίων (keywords), η μηχανή αναζήτησης διαδικτυακών υπηρεσιών Seekda προσφέρει στον χρήστη την δυνατότητα ορισμού διαφόρων κριτηρίων αναζήτησης. Τα κριτήρια μπορούν να είναι ο πάροχος της διαδικτυακής υπηρεσίας, η χώρα στην οποία φιλοξενείται και διάφορα tags, καθώς και συνδυασμοί αυτών με την χρήση λογικών τελεστών. Για παράδειγμα μια υπηρεσία με βάση τις λέξεις «fax» ή «sms» που δεν βρίσκεται όμως στις Ηνωμένες Πολιτείες, θα μπορούσε να αναζητηθεί με την εξής επερώτηση:

“fax OR sms -country:US”.

Ας σημειωθεί ότι προβλέπεται και η περίπτωση λανθασμένης εισαγωγής λέξεων λόγω ορθογραφίας προτείνοντας εναλλακτικές επιλογές, όπως γίνεται και στις τυπικές μηχανές αναζήτησης.

Κατά την ιεράρχηση των αποτελεσμάτων που επιστρέφονται στον χρήστη, πέρα από το βάρος, εξετάζονται ο χρόνος απόκρισης της υπηρεσίας και η φερεγγυότητα της υπηρεσίας, με την έννοια ότι ελέγχεται αν κάθε τερματικό που παρέχει την διαδικτυακή υπηρεσία ακολουθεί τα πρωτόκολλα που ισχυρίζεται ότι υποστηρίζει.

Λαμβάνεται επίσης υπόψη ο αριθμός των ιστοσελίδων που περιέχουν συνδέσμους προς κάθε διαδικτυακή υπηρεσία, καθώς υπηρεσίες με πολλούς συνδέσμους θεωρείται ότι είναι πιο συχνά χρησιμοποιούμενες άρα και καλύτερες.

Σε κάθε περίπτωση, ο χρήστης μπορεί να ρυθμίσει τον τρόπο παρουσίασης των αποτελεσμάτων με βάση παράγοντες όπως η σχετικότητα, η ύπαρξη τεκμηρίωσης, η διαθεσιμότητα της υπηρεσίας την τελευταία εβδομάδα ή τον τελευταίο χρόνο, η χώρα στην οποία φιλοξενείται και τα έτη που αυτή είναι διαθέσιμη.

Η αναζήτηση μπορεί να πραγματοποιηθεί και με διαφορετικό, μη αυτοματοποιημένο, τρόπο, καθώς στον χρήστη δίνεται η δυνατότητα να περιηγηθεί στις υπηρεσίες με την χρήση ενός tag cloud, μέσω καταλόγου με την μορφή θεματικών κατηγοριών.

Τελικά, παρακολουθείται ο τρόπος αλληλεπίδρασης των χρηστών με τη μηχανή αναζήτησης με σκοπό την βελτίωσή της.

2.2.2. Woogle

Το Woogle [22] είναι μια ακόμη μηχανή αναζήτησης για διαδικτυακές υπηρεσίες. Ο τρόπος λειτουργίας του διαφέρει σημαντικά από αυτόν μιας τυπικής μηχανής αναζήτησης, αφού αποκτά τις διαδικτυακές υπηρεσίες από μητρώα (UDDI registries), δεν ψάχνει δηλαδή στον Παγκόσμιο Ιστό για υπηρεσίες κάνοντας crawling. Στην συνέχεια, εξάγει πληροφορία από τις WSDL περιγραφές των διαδικτυακών υπηρεσιών, τις συνθέτει ώστε να δημιουργήσει έναν κατάλογο και τελικά παρουσιάζει στον χρήστη μια γραφική διεπαφή (interface), η οποία του παρέχει την δυνατότητα να αναζητήσει υπηρεσίες και στην οποία εκτίθεται η σημασιολογική σχέση ανάμεσά τους.

Στην πρώτη υλοποίηση του το 2003, το Woogle επέτρεπε στους χρήστες του να αναζητήσουν διαδικτυακές υπηρεσίες είτε με χρήση λέξεων-κλειδιών είτε με βάση την είσοδο και την έξοδο των υπηρεσιών. Υπήρχε επίσης η δυνατότητα περιήγησης στον θεματικό κατάλογο των διαδικτυακών υπηρεσιών ανά κατηγορία. Παρέχονταν ακόμα στον χρήστη πληροφορίες για την κατάσταση (status) της υπηρεσίας, αν δηλαδή λειτουργούσε ή όχι, και του επιτρεπόταν η δοκιμή της υπηρεσίας μέσω της ίδιας ιστοσελίδας, χωρίς να εγκαταλείψει τον ιστοχώρο της μηχανής αναζήτησης.

Οι δυνατότητες του Woogle επεκτάθηκαν δύο φορές το 2004. Την πρώτη φορά προστέθηκε η δυνατότητα αναζήτησης υπηρεσιών με κριτήριο την ομοιότητα. Η υπηρεσία αυτή ήταν εφικτή με την χρήση ενός αλγορίθμου ο οποίος εκμεταλλευόμενος τη δομή των διαδικτυακών υπηρεσιών χρησιμοποιούσε ένα μηχανισμό συσταδοποίησης (clustering mechanism) που ομαδοποιούσε τα ονόματα των παραμέτρων σε σημασιολογικές έννοιες. Ο χρήστης είχε έτσι την δυνατότητα παρέχοντας μια περιγραφή λειτουργικότητας να λάβει υπηρεσίες με παρόμοια λειτουργικότητα, είσοδο και έξοδο, αλλά και υπηρεσίες ικανές για σύνθεση, με σκοπό την παραγωγή πιο προηγμένης λειτουργικότητας.

Στην τρίτη εκδοχή του, το Woogle υποστήριζε τρεις νέους τρόπους αναζήτησης διαδικτυακών υπηρεσιών:

- Αναζήτηση με κριτήρια την είσοδο, την έξοδο και την περιγραφή της λειτουργικότητας των λειτουργιών διαδικτυακών υπηρεσιών.
- Αναζήτηση διαδικτυακών υπηρεσιών των οποίων συνδυασμένες οι λειτουργίες ικανοποιούν το αίτημα του χρήστη.
- Αναζήτηση διαδικτυακών υπηρεσιών που συνδυαζόμενες προσφέρουν παρεμφερή λειτουργικότητα με μια λειτουργία που έχει προσδιορίσει ο χρήστης.

Ας σημειωθεί πως η συγκεκριμένη μηχανή αναζήτησης δεν φαίνεται να υπάρχει πλέον διαθέσιμη, ενώ η ανάπτυξή της έχει εγκαταλειφθεί, πράγμα που ίσως οφείλεται στην καμπή που παρουσιάζει η χρήση των μητρώων UDDI τα τελευταία χρόνια.

2.2.3. WSExpress (QoS-aware Search Engine for Web Services)

Το WSExpress [92] είναι μια μηχανή αναζήτησης που κατατάσσει τις διαδικτυακές υπηρεσίες λαμβάνοντας υπόψη τόσο τις λειτουργικές απαιτήσεις που καθορίζονται από τα ερωτήματα (queries) των χρηστών, όσο και τα μη λειτουργικά χαρακτηριστικά των διαδικτυακών υπηρεσιών που έχουν να κάνουν με την ποιότητα υπηρεσίας που προσφέρουν.

Η υλοποίησή της πραγματοποιήθηκε με χρήση της γλώσσας προγραμματισμού Java. Οι διαδικτυακές υπηρεσίες που προσφέρονται εντοπίστηκαν μέσω αναζήτησης (crawling) στον Παγκόσμιο Ιστό, ενώ για να προσδιοριστούν οι μη λειτουργικές επιδόσεις των διαδικτυακών

υπηρεσιών χρησιμοποιήθηκαν 339 κατανεμημένοι υπολογιστές του Planet-lab [58] από 30 χώρες που παρακολουθούσαν τη χρήση τους.

Μια σειρά από πειράματα που συνολικά χρησιμοποίησαν 3738 διαδικτυακές υπηρεσίες (15811 λειτουργίες) από 69 χώρες πραγματοποιήθηκαν για να διαπιστωθεί η αποδοτικότητα του WSExpress.

Η μηχανή αναζήτησης λειτουργεί σε τρία στάδια: πρώτα πραγματοποιείται μη λειτουργική αξιολόγηση, ακολουθεί η λειτουργική αξιολόγηση και τελικά γίνεται κατάταξη των διαδικτυακών υπηρεσιών με βάση τα συμπεράσματα από τα δύο προηγούμενα στάδια.

Κατά το στάδιο της μη λειτουργικής αξιολόγησης – θα μπορούσε να εκληφθεί και ως αξιολόγηση με βάση την προσφερόμενη ποιότητα υπηρεσίας – αποκτώνται από την μηχανή αναζήτησης τιμές που αφορούν συγκεκριμένα κριτήρια μη λειτουργικής αξιολόγησης όπως είναι ο χρόνος απόκρισης, η ρυθμαπόδοση (throughput), η διαθεσιμότητα και η τιμή κάθε διαθέσιμης διαδικτυακής υπηρεσίας. Στη συνέχεια, ανάλογα με τους περιορισμούς και τις προτιμήσεις που έχουν εκφραστεί από τον χρήστη στο ερώτημά του, υπολογίζεται ένας δείκτης ποιότητας υπηρεσίας για κάθε διαδικτυακή υπηρεσία. Η παραπάνω περιγραφή της διαδικασίας αξιολόγησης των μη λειτουργικών χαρακτηριστικών γίνεται εφικτή χρησιμοποιώντας διανύσματα. Ειδικότερα, κάθε διαδικτυακή υπηρεσία αναπαρίσταται με ένα διάνυσμα το οποίο έχει μια μεταβλητή για κάθε κριτήριο. Μια συνάρτηση χρησιμότητας (utility function) αναλαμβάνει την αξιολόγηση της ποιότητας υπηρεσίας, αντιστοιχίζοντας μια πραγματική τιμή σε κάθε διαδικτυακή υπηρεσία. Ακολουθούν δύο βήματα ώστε η πραγματική τιμή που προκύπτει να εκφράζει τις προτιμήσεις του χρήστη. Τα κριτήρια ποιότητας υπηρεσίας αρχικά κανονικοποιούνται, ενώ τελικά ο αριθμός που αντιστοιχίζεται σε κάθε υπηρεσία προκύπτει έπειτα από σύγκριση των παραμέτρων της με τις απαιτήσεις του χρήστη. Στην συνέχεια, προστίθενται βάρη στα κριτήρια ανάλογα με τις προτεραιότητες που έχουν εκφραστεί από τον χρήστη.

Η λειτουργική αξιολόγηση πραγματοποιείται σε δύο φάσεις. Αρχικά, γίνεται προεπεξεργασία στις WSDL περιγραφές που συνοδεύουν τις διαδικτυακές υπηρεσίες. Αφού ολοκληρωθεί η επεξεργασία, η μηχανή αναζήτησης αξιολογεί τα λειτουργικά χαρακτηριστικά κάθε διαδικτυακής υπηρεσίας συγκρίνοντας τη λειτουργικότητα που έχει προσδιοριστεί από τον χρήστη στο ερώτημά του και τις λειτουργίες που προσφέρει κάθε διαδικτυακή υπηρεσία. Για τον προσδιορισμό της λειτουργίας μιας διαδικτυακής υπηρεσίας ακολουθείται αντίστοιχη διαδικασία με την αξιολόγηση των μη λειτουργικών χαρακτηριστικών, χρησιμοποιούνται δηλαδή και πάλι

διανύσματα. Ένα διάνυσμα αναπαριστά τις λειτουργικές απαιτήσεις του χρήστη και ένα δεύτερο διάνυσμα αναπαριστά τις λειτουργίες της διαδικτυακής υπηρεσίας. Τα διανύσματα έχουν τρεις μεταβλητές που αναπαριστούν τις λέξεις κλειδιά, τις εισόδους και τις εξόδους της διαδικτυακής υπηρεσίας. Πρώτα πραγματοποιείται προεπεξεργασία στην WSDL περιγραφή ώστε να προσδιοριστούν με τον καλύτερο δυνατό τρόπο οι λειτουργίες των διαδικτυακών υπηρεσιών και κάποια χαρακτηριστικά τους όπως η είσοδος και η έξοδος. Στην συνέχεια, χρησιμοποιώντας το μέτρο TF-IDF (Term Frequency - Inverse Document Frequency) καθορίζεται η ομοιότητα (εγγύτητα) της κάθε υπηρεσίας με την επιθυμία του χρήστη [62]. Τελικά, τα αποτελέσματα των δύο προηγούμενων σταδίων της λειτουργικής αξιολόγησης συνδυάζονται. Ο συνδυασμός τους αποτελεί το κριτήριο σύμφωνα με το οποίο οι υπηρεσίες ταξινομούνται ως προς τις λειτουργικές απαιτήσεις.

Η κατάταξη με την οποία εμφανίζονται τελικά οι διαδικτυακές υπηρεσίες στον χρήστη προκύπτει από μια συνάρτηση η οποία λαμβάνει υπόψη της την θέση της υπηρεσίας ως προς τις λειτουργικές και μη απαιτήσεις.

Τα παραπάνω χρησιμοποιούνται για να προσφέρουν βελτιστοποιημένη αναζήτηση ως προς τις επιμέρους απαιτήσεις των χρηστών – αφού λαμβάνονται υπόψη και παράγοντες ποιότητας υπηρεσίας – με κριτήρια τις λέξεις κλειδιά, τις λειτουργικές απαιτήσεις (είσοδο / έξοδο) και τη λειτουργική ομοιότητα των υπηρεσιών.

2.2.4. VUSE (VitaLab Universal WS Search Engine)

Το VUSE [59], όπως και το Woogle, δεν περιηγείται στον Παγκόσμιο Ιστό για να βρει υπηρεσίες, αλλά επικεντρώνεται στα μητρώα, κρατώντας τις περιγραφές που βρίσκει σε ένα αποθετήριο (repository). Η ευρετηρίαση (indexing) στο αποθετήριο πραγματοποιείται με τη χρήση ενός αλγορίθμου διανυσματικού χώρου (vector space). Κάθε έγγραφο αποτελεί ένα διάνυσμα στον χώρο αυτό, ενώ κάθε όρος του εγγράφου αποτελεί μια διαφορετική μεταβλητή του διανύσματος. Για την εύρεση της ομοιότητας της κάθε υπηρεσίας με την επερώτηση του χρήστη χρησιμοποιείται παραλλαγή του μέτρου TF-IDF. Σημειώνεται ότι η συγκεκριμένη μηχανή αναζήτησης δεν φαίνεται να είναι διαθέσιμη πλέον.

2.2.5. SWSE (Semantic Web Search Engine)

Σκοπός της μηχανής αναζήτησης SWSE [47] είναι να συνδυάσει την τεχνολογία που χρησιμοποιείται από τις σύγχρονες μηχανές αναζήτησης με την αξιοποίηση της σημασιολογικής πληροφορίας, ώστε να πετύχει την καλύτερη δυνατή ακρίβεια και ανάκληση. Ωστόσο, προς το παρόν δεν παρέχεται ούτε υλοποίηση ούτε αποτελέσματα πειραμάτων που να περιγράφουν την λειτουργικότητα και την αποτελεσματικότητά της.

2.2.6. OSSE (Ontology-Based Prototype Service Search Engine)

Η προσέγγιση της μηχανής αναζήτησης OSSE [19] βασίζεται στην αντίληψη ότι η αναζήτηση βάσει οντολογιών είναι αποτελεσματική. Προτείνει έναν αλγόριθμο ο οποίος δημιουργεί οντολογίες που περιγράφουν διαδικτυακές υπηρεσίες. Ωστόσο, προς το παρόν δεν παρέχεται ούτε η υλοποίηση, ούτε αποτελέσματα πειραμάτων που να στηρίζουν αυτή την προσέγγιση.

2.2.7. Σύνοψη

Το πρόβλημα της αναζήτησης διαδικτυακών υπηρεσιών που αντιμετωπίζεται από τις μηχανές αναζήτησης που περιγράφηκαν παραπάνω, προκύπτει από δύο βασικούς παράγοντες:

- τον συνεχώς αυξανόμενο αριθμό τους, που καθιστά τον εντοπισμό της κατάλληλης υπηρεσίας σε σχέση με τις ανάγκες των χρηστών δυσχερή, ακόμα και για πεπειραμένους χρήστες
- την ανεπάρκεια των υπάρχοντων διαδεδομένων μηχανών αναζήτησης γενικού σκοπού ως προς τον αποδοτικό και ακριβή εντοπισμό διαδικτυακών υπηρεσιών με μικρή παρέμβαση από τον χρήστη

Σαν λύση στο εν λόγω πρόβλημα στην βιβλιογραφία έχει προταθεί η ανάπτυξη εξειδικευμένων μηχανών αναζήτησης, οι οποίες στοχεύουν μόνο σε διαδικτυακές υπηρεσίες. Για την ανάπτυξή τους συναντώνται κυρίως δύο προσεγγίσεις: αυτές που χρησιμοποιούν κλασικούς αλγόριθμους ευρετηρίασης, και αποκτούν τις περιγραφές των διαδικτυακών υπηρεσιών μέσω αναζήτησης στον Παγκόσμιο Ιστό, και αυτές που χρησιμοποιούν μητρώα. Σχεδόν όλες οι μηχανές χρησιμοποιούν αναζήτηση μέσω λέξεων-κλειδιών, αρκετές από αυτές παρέχουν επιπλέον εξειδικευμένη λειτουργικότητα που αφορά συγκεκριμένα χαρακτηριστικά των προς αναζήτηση

υπηρεσιών, ενώ επίσης προσφέρεται και η δυνατότητα εντοπισμού μέσω περιήγησης σε θεματικούς καταλόγους.

Αν και οι προσπάθειες που παρουσιάζονται παραπάνω είναι αξιόλογες, διαφαίνεται πως υπάρχουν ακόμα σημαντικά περιθώρια βελτίωσης και μεγάλες ερευνητικές δυνατότητες στην περιοχή, που θα επικεντρώνονται κυρίως σε πιο αποδοτικούς τρόπους συγκέντρωσης των περιγραφών των διαδικτυακών υπηρεσιών από τον Παγκόσμιο Ιστό, καθώς και σε καλύτερους αλγόριθμους ευρετηρίασης, αναζήτησης και κατάταξης των αποτελεσμάτων. Επιπλέον, σημαντική βελτίωση θα αποτελέσει η εισαγωγή σημασιολογικών χαρακτηριστικών στην αναζήτηση, βασισμένων σε σημασιολογικές περιγραφές των διαδικτυακών υπηρεσιών. Τα παραπάνω θέματα αποτελούν το κύριο αντικείμενο της παρούσας πτυχιακής εργασίας.

ΚΕΦΑΛΑΙΟ 3

Πρότυπα Περιγραφής Διαδικτυακών Υπηρεσιών

Καθώς ο βασικός σκοπός της παρούσας πτυχιακής εργασίας αφορά την κατασκευή μιας μηχανής αναζήτησης εξειδικευμένης για περιγραφές διαδικτυακών υπηρεσιών, κρίνεται σκόπιμο, πριν την ανάλυση του συστήματος και την παρουσίαση της αρχιτεκτονικής και των λειτουργικών απαιτήσεων του, να παρουσιαστούν τα πλέον επικρατή πρότυπα περιγραφής διαδικτυακών υπηρεσιών. Τα αρχεία τα οποία η μηχανή αναζήτησης εντοπίζει, ευρετηριάζει και επιστρέφει στους χρήστες σαν αποτελέσματα συμμορφώνονται με αυτά τα πρότυπα.

Εκτός από την περιγραφή των προτύπων, το παρόν Κεφάλαιο έχει ως σκοπό την παρουσίαση και κατανόηση των συστατικών στοιχείων τους, από την οποία προκύπτει ο καθορισμός των στοιχείων εκείνων στον ορισμό του προτύπου τα οποία είναι τα σημαντικότερα για την περιγραφή μιας διαδικτυακής υπηρεσίας, περιέχουν δηλαδή την πιο χρήσιμη πληροφορία. Αυτά είναι και τα στοιχεία που θα χρησιμοποιηθούν στο πλαίσιο της παρούσας εργασίας.

3.1.WSDL

3.1.1. Περιγραφή

Η WSDL (Web Service Description Language) [q1] – γλώσσα περιγραφής διαδικτυακών υπηρεσιών, όπως δηλώνει και το όνομά της, παρέχει έναν πρότυπο τρόπο για την περιγραφή των διαδικτυακών υπηρεσιών, ενώ αποτελεί το πλέον διαδεδομένο πρότυπο περιγραφής.

Με την προτυποποίηση των πρωτοκόλλων επικοινωνίας και της μορφής των μηνυμάτων στην διαδικτυακή κοινότητα καθίσταται δυνατή αλλά και απαραίτητη η περιγραφή των επικοινωνιών με δομημένο τρόπο. Η WSDL αντιμετωπίζει αυτή την ανάγκη καθορίζοντας ένα συντακτικό για την περιγραφή των διαδικτυακών υπηρεσιών σαν συλλογές από τερματικά επικοινωνίας, ικανά να ανταλλάσσουν μηνύματα. Η περιγραφή των διαδικτυακών υπηρεσιών με αυτόν τον τρόπο από την WSDL παρέχει τεκμηρίωση για τα κατανεμημένα συστήματα. Επίσης, αποτελεί βασικό συστατικό για τη ρύθμιση των λεπτομερειών εκείνων που διευκολύνουν την αυτοματοποίηση εφαρμογών επικοινωνίας. Το συντακτικό που παρέχει η WSDL βασίζεται στην XML (Extensible Markup Language) [26].

Σε αντίθεση με την αρχική έκδοση, WSDL 1.1, η δεύτερη έκδοση της WSDL έγινε πρότυπο του World Wide Web Consortium (W3C) τον Ιούνιο του 2007. Η αρχική ονομασία της δεύτερης έκδοσης ήταν WSDL 1.2, αλλά μετονομάστηκε σε WSDL 2.0 [84] αφού παρουσίαζε σημαντικές διαφορές από την WSDL 1.1.

Σε σχέση με την WSDL 1.1, η WSDL 2.0 έχει τις διάφορες προσθήκες και αλλαγές, όπως την υποστήριξη περαιτέρω σημασιολογικής πληροφορίας στην γλώσσα περιγραφής, την αφαίρεση κάποιων δομών και την μετονομασία άλλων [79]. Η WSDL 2.0 επίσης παρέχει την δυνατότητα διαχωρισμού της αφαιρετικής (abstract) περιγραφής της λειτουργικότητας μιας διαδικτυακής υπηρεσίας από συγκεκριμένες (concrete) λεπτομέρειες που αφορούν τον τρόπο που προσφέρεται αυτή η λειτουργικότητα. Ο διαχωρισμός αυτός καθιστά ευδιάκριτα τα διάφορα τμήματα μιας υπηρεσίας και της περιγραφής της, προάγοντας έτσι την κατανόηση αλλά και την επαναχρησιμοποίησή της. Η αφηρημένη και η συγκεκριμένη περιγραφή μιας υπηρεσίας συνυπάρχουν στο ίδιο αρχείο WSDL.

Στο αφαιρετικό επίπεδο, η WSDL περιγράφει μια διαδικτυακή υπηρεσία σαν μια διεπαφή (interface) ανάμεσα σε έναν «καταναλωτή» (*consumer*), ο οποίος μπορεί να είναι χρήστης, πληροφοριακό σύστημα, ευφυής πράκτορας (intelligent agent) ή ακόμα και κάποια άλλη διαδικτυακή υπηρεσία, και σε κάποια βασική λειτουργικότητα που προσφέρεται με την μορφή υπηρεσίας. Η διεπαφή αυτή ορίζεται σαν ένα σύνολο από *λειτουργίες* (*operations*), που αντιπροσωπεύουν μια απλή αλληλεπίδραση ανάμεσα στον καταναλωτή και την υπηρεσία. Κάθε τέτοια λειτουργία έχει καθορισμένο τύπο μηνυμάτων (messages) που μπορεί να στείλει και να λάβει, καθώς και συγκεκριμένη σειρά με την οποία θα πραγματοποιηθεί η ανταλλαγή των μηνυμάτων αυτών.

Ο τρόπος με τον οποίο θα γίνει η ανταλλαγή των παραπάνω μηνυμάτων εμπίπτει στο συγκεκριμένο (concrete) επίπεδο και καθορίζεται από ένα *binding*. Για κάθε διεπαφή που ορίζεται στο αφαιρετικό επίπεδο, είναι απαραίτητη η ύπαρξη ενός αντίστοιχου binding το οποίο ορίζει λεπτομέρειες όπως είναι το περιεχόμενο και η μορφοποίηση των μηνυμάτων και το πρωτόκολλο επικοινωνίας που χρησιμοποιείται για την μετάδοσή τους. Αυτά περιλαμβάνουν τα πρωτόκολλα SOAP (Simple Object Access Protocol) και HTTP (Hypertext Transfer Protocol). Το SOAP μπορεί να χρησιμοποιεί το πρωτόκολλο HTTP σε χαμηλότερο επίπεδο, για τη μεταφορά του, οπότε και πρέπει να προσδιορίζεται αν κάθε λειτουργία θα χρησιμοποιεί μέθοδο HTTP GET ή POST. Επίσης, είναι δυνατό να χρησιμοποιηθεί μόνο το πρωτόκολλο HTTP για το

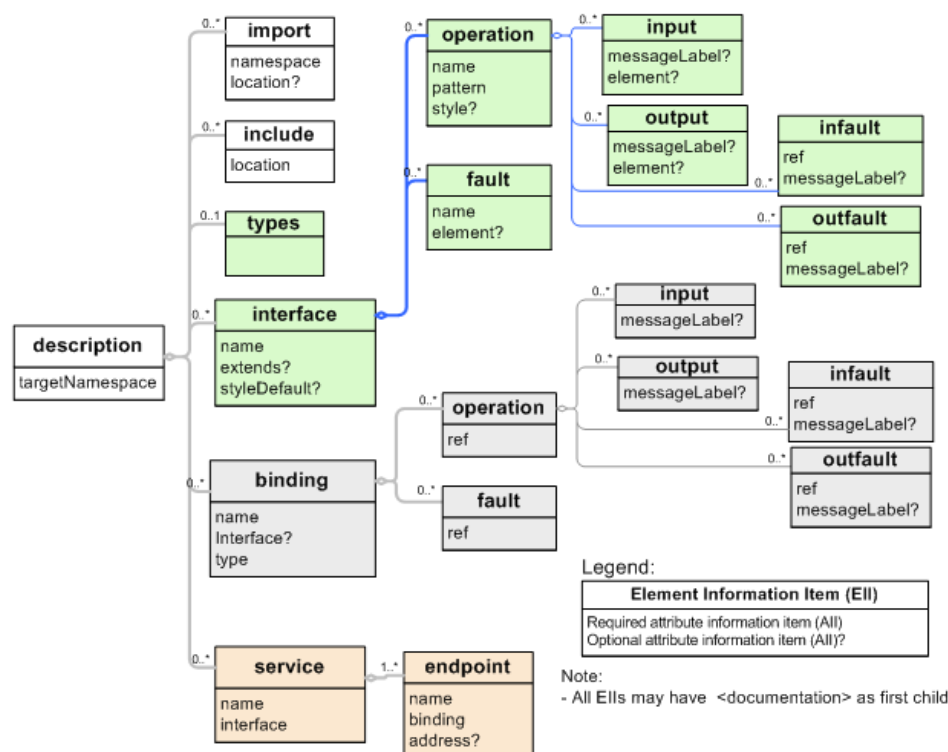
binding, οπότε η αποστολή και λήψη των μηνυμάτων πραγματοποιείται χωρίς αυτά να κωδικοποιηθούν σε SOAP.

Τα *τερματικά (endpoints)* μέσω των οποίων είναι διαθέσιμες όλες οι λειτουργίες μιας διαδικτυακής υπηρεσίας ορίζονται στην WSDL περιγραφή της εντός ενός στοιχείου *υπηρεσίας (service)*. Κάθε τερματικό πρέπει να είναι συσχετισμένο με ένα συγκεκριμένο binding που καθορίζει τον τρόπο πρόσβασης στην υπηρεσία για το συγκεκριμένο τερματικό.

Η δομή μιας WSDL 2.0 περιγραφής μπορεί να αναπαρασταθεί με διάφορους τρόπους όπως το XML Infoset (XML Information Set) [88] σε επίπεδο συντακτικού, το XML Schema [12], και σε υψηλότερο επίπεδο το Component Model.

Στο συντακτικό επίπεδο, μια WSDL περιγραφή συμβαδίζει με τους κανόνες που ορίζει το XML Infoset. Συγκεκριμένα, αποτελείται από ένα στοιχείο *περιγραφής (description)*, το οποίο βρίσκεται στο ιεραρχικά ανώτερο επίπεδο και εμπεριέχει όλα τα υπόλοιπα στοιχεία, ακολουθώντας τους κανόνες τεκμηρίωσης του προτύπου.

Το διάγραμμα που φαίνεται στην Εικόνα 7 παρουσιάζει τη συντακτική δομή ενός WSDL 2.0 εγγράφου χρησιμοποιώντας το XML Infoset.



Εικόνα 7. Η συντακτική δομή ενός WSDL 2.0 εγγράφου χρησιμοποιώντας το XML Infoset [82].

Το πρότυπο παρέχει επίσης ένα XML Schema. Το σχήμα αυτό, σε συνδυασμό με την τεκμηρίωση, θέτουν τους περιορισμούς σύμφωνα με τους οποίους ένα έγγραφο λογίζεται σαν έγκυρη WSDL 2.0 περιγραφή. Ο συνδυασμός αυτός μπορεί να γίνει κατανοητός αν εξεταστεί ο τρόπος που παρουσιάζονται τα στοιχεία σε μια περιγραφή, καθώς το σχήμα δεν καθορίζει μια συγκεκριμένη σειρά παρουσίασης των στοιχείων, κάτι όμως που γίνεται από την τεκμηρίωση.

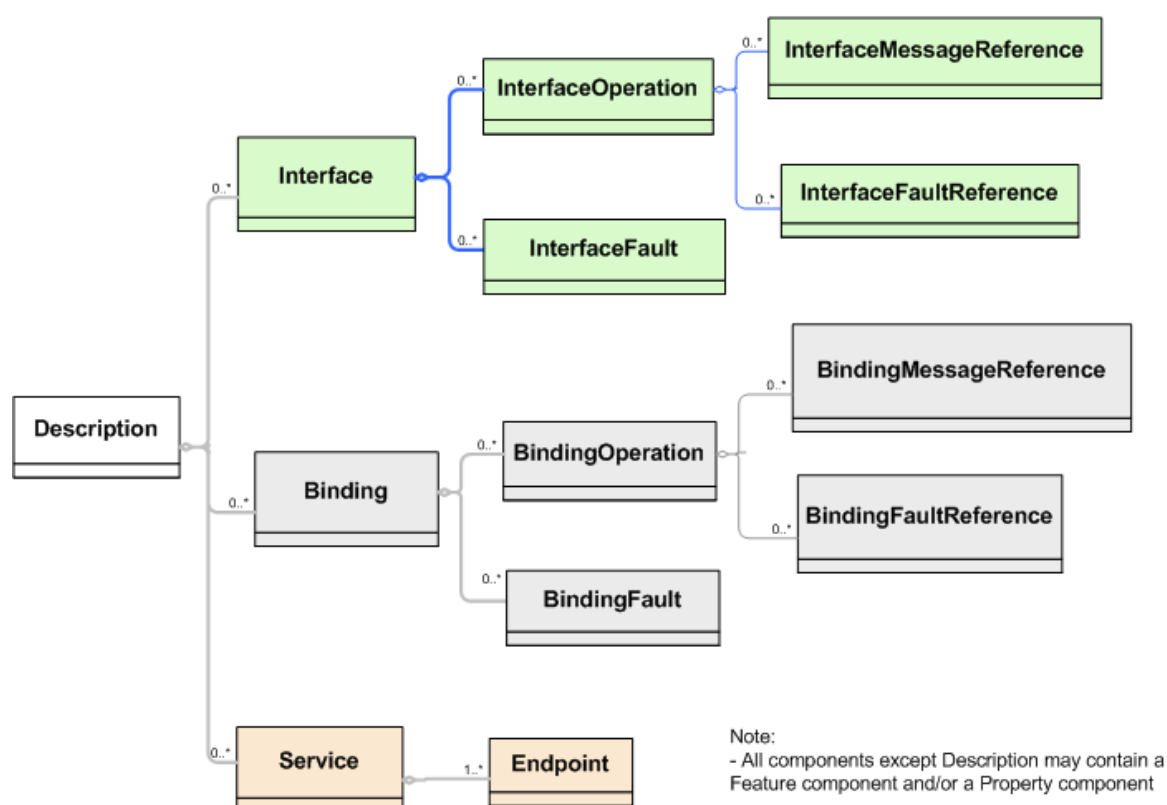
Το στοιχείο `description` δρα σαν container για όλα τα υπόλοιπα στοιχεία μιας WSDL περιγραφής. Σε αυτό άμεσα δηλώνονται μόνο τα namespaces. Τα στοιχεία απόγονοι του στοιχείου `description` και η σειρά με την οποία εμφανίζονται σε μια περιγραφή είναι τα εξής:

- Πρώτο βρίσκεται ένα στοιχείο *τεκμηρίωσης (documentation)*. Η ύπαρξή του είναι προαιρετική. Το κείμενο της τεκμηρίωσης απευθύνεται σε ανθρώπους και παρέχει διάφορες πληροφορίες για την υπηρεσία. Συνήθως περιλαμβάνει ανακατεύθυνση σε εξωτερική πηγή τεκμηρίωσης.
- Ακολουθούν μηδέν ή περισσότερα στοιχεία από τα παρακάτω σε οποιαδήποτε σειρά.
 - *include*: καθιστά μια WSDL 2.0 περιγραφή ικανή να περιλαμβάνει στοιχεία (components) που έχουν δηλωθεί σε άλλες περιγραφές σαν να έχουν δηλωθεί στην ίδια, χωρίς να τα εισάγει στην περιγραφή.
 - *import*: λειτουργεί σαν μια αναφορά σε στοιχεία που έχουν δηλωθεί σε άλλη περιγραφή, εισάγοντας τις περιγραφές τους.
- Ένα προαιρετικό στοιχείο *types*, το οποίο καθορίζει τους τύπους δεδομένων των μηνυμάτων που ανταλλάσσει η διαδικτυακή υπηρεσία
- Μηδέν ή περισσότερα στοιχεία από τα παρακάτω σε οποιαδήποτε σειρά.
 - *interface*: περιγράφει με αφαιρετικό τρόπο την λειτουργικότητα που προσφέρει μια διαδικτυακή υπηρεσία. Το κάθε interface μπορεί να περιέχει ένα ή περισσότερα στοιχεία *operation*, το καθένα εκ των οποίων αφορά μια απλή αλληλεπίδραση ανάμεσα στον χρήστη και την υπηρεσία. Για το κάθε στοιχείο *operation* ορίζεται ένα στοιχείο *input* και ένα στοιχείο *output*, τα οποία καθορίζουν αντίστοιχα τα μηνύματα εισόδου και εξόδου του *operation*. Σε κάθε *operation* αντιστοιχεί και ένα στοιχείο *fault*, το οποίο περιγράφει καταστάσεις λάθους που είναι δυνατόν να συμβούν κατά την εκτέλεση της υπηρεσίας, και

περιέχει τα στοιχεία *infaul*t και *outfaul*t που αφορούν σφάλματα μηνυμάτων εισόδου και εξόδου αντίστοιχα.

- *binding*: καθορίζει τις λεπτομέρειες (πρωτόκολλο επικοινωνίας, μορφοποίηση μηνύματος) για κάθε λειτουργία και σφάλμα σε μια διεπαφή οπότε υπάρχει αντιστοιχία ανάμεσα στα στοιχεία του και μιας διεπαφής
- *service*: προσδιορίζει σε ποιά τερματικά μπορεί να βρεθεί κάθε διεπαφή, και περιέχει ένα ή περισσότερα στοιχεία *endpoint*, τα οποία καθορίζουν τη διεύθυνση στην οποία μπορεί να βρεθεί μια υπηρεσία

Για τον ακριβή ορισμό μίας WSDL 2.0 περιγραφής όμως δεν αρκούν μόνο τα παραπάνω στοιχεία, αφού η WSDL 2.0 θέτει σημασιολογικούς περιορισμούς σε υψηλότερο επίπεδο από τους δομικούς και συντακτικούς που καλύπτουν τα XML Infoset και Schema. Για αυτόν τον λόγο, ήταν αναγκαία η προσθήκη του *component model* (Εικόνα 8), μίας αφαιρετικής περιγραφής σε υψηλότερο επίπεδο, η οποία είναι σε θέση να καταγράψει την σημασία μιας WSDL 2.0 περιγραφής. [82]



Εικόνα 8. Το component model μιας WSDL 2.0 περιγραφής. [82]

3.1.2. Παραδείγματα

3.1.2.1. reservationList.wsdl

Το παρακάτω WSDL 2.0 έγγραφο παρουσιάζει την περιγραφή μιας διαδικτυακής υπηρεσίας που επιστρέφει τις κρατήσεις ενός ξενοδοχείου. Η συγκεκριμένη διαδικτυακή υπηρεσία παρέχει μια λειτουργία (operation), η οποία δίνει τη δυνατότητα επιστροφής όλων των κρατήσεων αλλά και μέρος αυτών ανάλογα με κάποιο κριτήριο αναζήτησης. Τα κριτήρια αναζήτησης μπορούν να είναι ο αριθμός της κράτησης, η ημερομηνία εισόδου και η ημερομηνία εξόδου [82].

Σκοπός του συγκεκριμένου παραδείγματος, που παρέχεται στο primer του προτύπου WSDL 2.0, είναι να επιδείξει τα αφαιρετικά (abstract) και τα συγκεκριμένα (concrete) τμήματα μιας τέτοιας περιγραφής. Η απεικόνιση έχει γίνει με χρήση σχολίων, καθώς και με την παρακάτω χρωματική κωδικοποίηση:

Abstract Type



Abstract Interface



Concrete Binding with SOAP



Concrete Service



```
<?xml version="1.0" encoding="utf-8" ?>
<description
  xmlns="http://www.w3.org/ns/wsdl"
  targetNamespace="http://greath.example.com/2004/services/reservationList"
  xmlns:tns="http://greath.example.com/2004/services/reservationList"
  xmlns:details="http://greath.example.com/2004/schemas/reservationDetails"
  xmlns:list="http://greath.example.com/2004/schemas/reservationList"
  xmlns:soap="http://www.w3.org/ns/wsdl/soap"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">

  <documentation>
    This document describes the GreatH Reservation List Web
    services. Use this service to retrieve lists of reservations
    based on a variety of search criteria.
  </documentation>

  <!-- Abstract Type -->
  <types>
    <xs:import
      namespace="http://greath.example.com/2004/schemas/reservationDetails"
      schemaLocation="reservationDetails.xsd" />
    <xs:import
      namespace="http://greath.example.com/2004/schemas/reservationList"
      schemaLocation="reservationList.xsd" />
  </types>

  <!-- Abstract Interface -->
  <interface name="reservationListInterface">

    <operation name="retrieve"
      pattern="http://www.w3.org/ns/wsdl/in-out">
      <input messageLabel="In" element="#none" />
      <output messageLabel="Out" element="list:reservationList" />
    </operation>
```

```

        <operation name="retrieveByConfirmationNumber"
            pattern="http://www.w3.org/ns/wsdl/in-out">
            <input messageLabel="In"
                element="details:confirmationNumber" />
            <output messageLabel="Out" element="list:reservationList" />
        </operation>

        <operation name="retrieveByCheckInDate"
            pattern="http://www.w3.org/ns/wsdl/in-out">
            <input messageLabel="In" element="details:checkInDate" />
            <output messageLabel="Out" element="list:reservationList" />
        </operation>

        <operation name="retrieveByCheckOutDate"
            pattern="http://www.w3.org/ns/wsdl/in-out">
            <input messageLabel="In" element="details:checkOutDate" />
            <output messageLabel="Out" element="list:reservationList" />
        </operation>

    </interface>

    <!-- Concrete Binding with SOAP -->
    <binding name="reservationListSOAPBinding"
        interface="tns:reservationListInterface"
        type="http://www.w3.org/ns/wsdl/soap"
        wssoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/">

        <operation ref="tns:retrieve"
            wssoap:mep="http://www.w3.org/2003/05/soap/mep/request-
            response" />

        <operation ref="tns:retrieveByConfirmationNumber"
            wssoap:mep="http://www.w3.org/2003/05/soap/mep/request-
            response" />

        <operation ref="tns:retrieveByCheckInDate"
            wssoap:mep="http://www.w3.org/2003/05/soap/mep/request-
            response" />

        <operation ref="tns:retrieveByCheckOutDate"
            wssoap:mep="http://www.w3.org/2003/05/soap/mep/request-
            response" />

    </binding>

    <!-- Concrete Service-->
    <service name="reservationListService"
        interface="tns:reservationListInterface">

        <endpoint name="reservationListEndpoint"
            binding="tns:reservationListSOAPBinding"
            address="http://greath.example.com/2004/reservationList" />

    </service>

</description>

```

3.1.2.2. GreatH.wsdl

Το παρακάτω έγγραφο WSDL 2.0, που επίσης αποτελεί μέρος του primer του προτύπου WSDL 2.0, περιγράφει μια διαδικτυακή υπηρεσία κρατήσεων ενός υποθετικού ξενοδοχείου που ονομάζεται Hotel GreatH. Η συγκεκριμένη διαδικτυακή υπηρεσία έχει μία μόνο λειτουργία (operation), η οποία λαμβάνει σαν είσοδο την ημερομηνία εισόδου (check-in date), την ημερομηνία εξόδου (check-out date) και το είδος του δωματίου που θέλει να κλείσει ο χρήστης. Στην συνέχεια ελέγχει αν υπάρχει διαθέσιμο δωμάτιο σύμφωνα με τα στοιχεία, και αν βρει, επιστρέφει έναν αριθμό που αντιστοιχεί στην ημερήσια χρέωσή του. Αν δεν υπάρχει διαθέσιμο

δωμάτιο επιστρέφει 0, ενώ αν δεν δόθηκαν σωστά δεδομένα εισόδου επιστρέφει μήνυμα σφάλματος [82].

Σκοπός του συγκεκριμένου παραδείγματος είναι να επιδείξει τα βασικά δομικά στοιχεία της WSDL 2.0 περιγραφής που αφορούν στη λειτουργικότητα που προσφέρει η υπηρεσία. Η απεικόνιση έχει γίνει με χρήση σχολίων, καθώς και με την παρακάτω χρωματική κωδικοποίηση:

Documentation, **Interface**, **Binding**, **Service**.

```
<?xml version="1.0" encoding="utf-8" ?>
<description
  xmlns="http://www.w3.org/ns/wsd1"
  targetNamespace= "http://greath.example.com/2004/wsd1/resSvc"
  xmlns:tns= "http://greath.example.com/2004/wsd1/resSvc"
  xmlns:ghns = "http://greath.example.com/2004/schemas/resSvc"
  xmlns:wssoap= "http://www.w3.org/ns/wsd1/soap"
  xmlns:soap="http://www.w3.org/2003/05/soap-envelope"
  xmlns:wsd1x= "http://www.w3.org/ns/wsd1-extensions">

  <!-- Documentation -->
  <documentation>
    This document describes the GreatH Web service. Additional
    application-level requirements for use of this service --
    beyond what WSDL 2.0 is able to describe -- are available
    at http://greath.example.com/2004/reservation-documentation.html
  </documentation>

  <types>
    <xs:schema
      xmlns:xs="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://greath.example.com/2004/schemas/resSvc"
      xmlns="http://greath.example.com/2004/schemas/resSvc">

      <xs:element name="checkAvailability" type="tCheckAvailability"/>
      <xs:complexType name="tCheckAvailability">
        <xs:sequence>
          <xs:element name="checkInDate" type="xs:date"/>
          <xs:element name="checkOutDate" type="xs:date"/>
          <xs:element name="roomType" type="xs:string"/>
        </xs:sequence>
      </xs:complexType>

      <xs:element name="checkAvailabilityResponse" type="xs:double"/>

      <xs:element name="invalidDataError" type="xs:string"/>

    </xs:schema>
  </types>

  <!-- Interface -->
  <interface name = "reservationInterface" >

    <fault name = "invalidDataFault"
      element = "ghns:invalidDataError"/>

    <operation name="opCheckAvailability"
      pattern="http://www.w3.org/ns/wsd1/in-out"
      style="http://www.w3.org/ns/wsd1/style/iri"
      wsdlx:safe = "true">
      <input messageLabel="In"
        element="ghns:checkAvailability" />
      <output messageLabel="Out"
        element="ghns:checkAvailabilityResponse" />
      <outfault ref="tns:invalidDataFault" messageLabel="Out"/>
    </operation>
  </interface>

  <!-- Binding -->
  <binding name="reservationSOAPBinding"
    interface="tns:reservationInterface"
```

```

        type="http://www.w3.org/ns/wsdl/soap"
        wsoap:protocol="http://www.w3.org/2003/05/soap/bindings/HTTP/">

        <fault ref="tns:invalidDataFault"
            wsoap:code="soap:Sender"/>

        <operation ref="tns:opCheckAvailability"
            wsoap:mep="http://www.w3.org/2003/05/soap/mep/soap-response"/>
    </binding>

<!-- Service -->
<service name="reservationService"
    interface="tns:reservationInterface">

    <endpoint name="reservationEndpoint"
        binding="tns:reservationSOAPBinding"
        address ="http://greath.example.com/2004/reservation"/>

</service>

</description>

```

3.1.3. Αναζήτηση

Η αναλυτική μελέτη του προτύπου WSDL που προηγήθηκε σκοπό είχε την εξοικείωση με τα συστατικά στοιχεία μιας περιγραφής WSDL. Καθένα από αυτά τα στοιχεία συμβάλλει με έναν ξεχωριστό τρόπο που περιγράφηκε παραπάνω στην πληρότητα του προτύπου. Ωστόσο, για την αναζήτηση περιγραφών διαδικτυακών υπηρεσιών που αποτελεί τον σκοπό της μηχανής αναζήτησης που κατασκευάζεται, μόνο ένα μέρος από αυτά τα συστατικά στοιχεία προσφέρουν χρήσιμη πληροφορία - πληροφορία που με κάποιο τρόπο περιγράφει την λειτουργία που προσφέρει η υπηρεσία.

Πριν προσδιορισθούν τα συγκεκριμένα σημεία μιας περιγραφής αξίζει να σημειωθεί πως δύο πολύ σημαντικά στοιχεία που εξυπηρετούν αυτό τον σκοπό βρίσκονται εκτός της περιγραφής. Τα δύο αυτά στοιχεία είναι:

- Η διεύθυνση όπου εντοπίσθηκε η περιγραφή διαδικτυακής υπηρεσίας.

Τα URL αυτά θα αποτελούν το ουσιαστικό κομμάτι των αποτελεσμάτων που επιστρέφονται στον χρήστη, δίνοντας του την δυνατότητα να ανακατευθυνθεί σε μια περιγραφή κατά τρόπο αντίστοιχο με την ανακατεύθυνση σε σχετικές με την επρώτηση ιστοσελίδες στις συμβατικές μηχανές αναζήτησης.

- Το όνομα του αρχείου της περιγραφής.

Το όνομα συνήθως συμβαδίζει με το όνομα της περιγραφής που θα δούμε παρακάτω. Ωστόσο για να διασφαλιστούν και οι περιπτώσεις όπου αυτό δεν συμβαίνει κρατάμε και

το όνομα του αρχείου το οποίο είθισται να είναι περιγραφικό της λειτουργικότητας που προσφέρεται.

Τα δύο παραπάνω εξωτερικά στοιχεία συμβάλλουν κυρίως στην αναγνώριση και διευθυνσιοδότηση μιας περιγραφής. Αντίθετα, τα στοιχεία εκείνα που αποτελούν τμήματα της περιγραφής προσφέρουν χρήσιμη πληροφορία που καθιστά δυνατό τον προσδιορισμό της παρεχόμενης λειτουργικότητας. Τα στοιχεία αυτά είναι:

- Τα κάθε είδους ονόματα (names) στην περιγραφή ανεξάρτητα από τα tags στα οποία εμφανίζονται.

Τα ονόματα εξ ορισμού είναι περιγραφικά, ώστε να είναι σαφής ο σκοπός τους. Η πληροφορία κάθε ονόματος είναι χρήσιμη αφού το σύνολο όλων των ονομάτων δίνουν μια σαφή εικόνα του τύπου της υπηρεσίας που προσφέρεται. Περιλαμβάνουν το όνομα της υπηρεσίας και τα ονόματα των λειτουργιών της μαζί με άλλα λιγότερο προφανή ονόματα.

- Η τεκμηρίωση (documentation) της περιγραφής και των λειτουργιών της.

Η τεκμηρίωση αποτελεί προαιρετικό στοιχείο ενός WSDL 2.0 εγγράφου. Παρόλο που βρίσκεται μέσα σε ένα αρχείο WSDL 2.0, πρόκειται για κείμενο που απευθύνεται σε ανθρώπους (human readable text). Παρέχει διάφορες πληροφορίες για την υπηρεσία και συνήθως παραθέτει κάποιο link για εξωτερική αναλυτικότερη τεκμηρίωση. Κατά τρόπο αντίστοιχο με την τεκμηρίωση της περιγραφής, οι τεκμηριώσεις των λειτουργιών εξυπηρετούν τον ίδιο σκοπό συγκεκριμένα για τις λειτουργίες. Το συγκεκριμένο τμήμα θα μπορούσε να χρησιμοποιηθεί για την παροχή πληροφορίας στα αποτελέσματα που επιστρέφει η μηχανή αναζήτησης όπως συμβαίνει με τα snippets στις συμβατικές μηχανές αναζήτησης.

- Οι είσοδοι (inputs) και έξοδοι (outputs) των λειτουργιών.

Οι λειτουργίες αποτελούν τις ενέργειες που μπορεί να πραγματοποιήσει η υπηρεσία. Οι είσοδοι και οι έξοδοι των λειτουργιών παρέχουν επιπλέον πληροφορία, ενώ μπορούν να χρησιμοποιηθούν και σαν κριτήρια σύνθετης αναζήτησης.

3.2. SAWSDL

3.2.1. Περιγραφή

Όπως διαπιστώθηκε, οι WSDL 2.0 περιγραφές των διαδικτυακών υπηρεσιών επικεντρώνονται στο συντακτικό επίπεδο και επομένως σε αρκετές περιπτώσεις δεν είναι σε θέση να αποτυπώσουν τις απαιτήσεις και τις δυνατότητες μιας διαδικτυακής υπηρεσίας με τρόπο κατανοητό τόσο από τον άνθρωπο όσο και από τον υπολογιστή.

Η SAWSDL (Semantic Annotations for Web Services Description Language) [65] αποτελεί επέκταση της WSDL σύμφωνα με τους κανόνες επεκτασιμότητας που ορίζουν τα πρότυπα των WSDL 1.0, WSDL 2.0 και XML Schema. Σκοπός της είναι η προσθήκη σημασιολογικής πληροφορίας σε WSDL περιγραφές, διατηρώντας πλήρη διαλειτουργικότητα με τα χρησιμοποιούμενα πρότυπα. Η σημασιολογική πληροφορία μπορεί να ενσωματωθεί με αυτόν τον τρόπο σε υπάρχοντα WSDL αρχεία. Εγκρίθηκε σαν πρότυπο από το W3C τον Αύγουστο του 2007.

Η σημασιολογική πληροφορία αποτυπώνεται με σημασιολογικές επισημάνσεις (semantic annotations), οι οποίες είναι XML χαρακτηριστικά (attributes) που προστίθενται στην WSDL περιγραφή ή σε κάποιο σχετιζόμενο XML Schema, ως επισημάνσεις στο αντίστοιχο XML στοιχείο που περιγράφουν. Η χρήση τους επιτρέπει στην WSDL περιγραφή να αναφερθεί σε έννοιες (concepts) που ανήκουν σε σημασιολογικά μοντέλα.

Οι σημασιολογικές επισημάνσεις είναι δύο ειδών

- αναγνωριστικά σε έννοιες που αναφέρονται σε μια οντολογία (ontology)
- αναγνωριστικά αντιστοιχίσεων (mappings) από WSDL στοιχεία σε έννοιες και αντίστροφα.

Σημειώνεται ότι οι σημασιολογικές επισημάνσεις είναι ανεξάρτητες από την γλώσσα περιγραφής κάθε οντολογίας (ontology expression language) αλλά και από την γλώσσα αντιστοίχισης (mapping language).

Οι σημασιολογικές επισημάνσεις μπορούν να χρησιμοποιηθούν για την παροχή επιπλέον πληροφοριών κατηγοριοποίησης, ώστε να δημοσιευθεί μια διαδικτυακή υπηρεσία σε έναν κατάλογο (registry). Επίσης, συμβάλλουν στην δυναμική ανακάλυψη (discovery) των διαδικτυακών υπηρεσιών, και στην αυτοματοποίηση διαδικασιών σχετικών με τη διαχείριση των διαδικτυακών υπηρεσιών, όπως για παράδειγμα τη σύνθεση (composition), την εκτέλεση (execution) και τη συνεργασία (orchestration) τους [33][65]. Τέλος, μπορούν να διευκολύνουν

τη μετατροπή υπάρχοντων εφαρμογών σε διαδικτυακές υπηρεσίες, αφού εμπλουτίζουν την εκφραστικότητα των WSDL περιγραφών, με απώτερο στόχο την ολοκλήρωση επιχειρησιακών διαδικασιών (business process integration) [57].

Μια περιγραφή σε SAWSDL έχει όλα τα στοιχεία της WSDL που προαναφέρθηκαν, καθώς και κάποια επιπλέον χαρακτηριστικά επέκτασης. Ακολουθεί η περιγραφή των χαρακτηριστικών επέκτασης που καθορίζονται από την SAWSDL:

Το χαρακτηριστικό επέκτασης `modelReference` (αναφορά μοντέλου) αποτελείται από ένα ή περισσότερα URI, τα οποία αποτελούν αναγνωριστικά εννοιών / δείχνουν σε έννοιες σε σημασιολογικά μοντέλα. Σκοπός της αντιστοίχισης αυτής είναι η παροχή σημασιολογικής πληροφορίας για το στοιχείο της WSDL περιγραφής ή του XML Schema που είναι αντικείμενο της επισήμανσης. Χρησιμοποιείται για να τοποθετήσει σημασιολογικές επισημάνσεις σε ορισμούς τύπων, δηλώσεις στοιχείων και χαρακτηριστικών ενός XML Schema αλλά και σε διεπαφές, λειτουργίες και σφάλματα ενός WSDL αρχείου.

Τα χαρακτηριστικά επέκτασης που ονομάζονται `liftingSchemaMapping` και `loweringSchemaMapping` προστίθενται στις δηλώσεις στοιχείων και τύπων ενός XML Schema για την εξομάλυνση της αντιστοίχισης ανάμεσα στα στοιχεία της XML και του σημασιολογικού μοντέλου, όταν παρουσιάζονται διαφορές.

Οι αναφορές μοντέλων μπορούν να χρησιμοποιηθούν με επιτυχία για να προσδιορίσουν αν μια υπηρεσία ανταποκρίνεται στις απαιτήσεις ενός πελάτη με βάση την σημασιολογία. Ωστόσο, δεν εγγυώνται ότι δεν θα υπάρχουν διαφορές ανάμεσα στο σημασιολογικό μοντέλο και στην δομή των εισόδων και εξόδων. Αν υπάρχουν τέτοιου είδους διαφορές, είναι αναγκαία η χρήση των `liftingSchemaMapping` και `loweringSchemaMapping`. Για παράδειγμα έστω πως ένας πελάτης έχει δώσει το όνομα του και το επώνυμο του σαν δεδομένα ενώ η διαδικτυακή υπηρεσία απαιτεί ενιαίο το ονοματεπώνυμο. Ένα `lowering schema mapping` θα έπαιρνε τα σημασιολογικά δεδομένα του πελάτη / χρήστη και θα τα μετέτρεπε σε XML, ενώ ταυτόχρονα θα πραγματοποιούσε την συνένωση ώστε να παραχθεί ενιαίο το ονοματεπώνυμο.

Η αντιστοιχία περισσότερων της μίας σημασιολογικών επισημάνσεων σε ένα WSDL στοιχείο επιτρέπεται από την SAWSDL, είτε αυτά είναι σχήματα αντιστοίχισης είτε αναφορές μοντέλου. Πολλαπλά σχήματα αντιστοίχισης ερμηνεύονται σαν εναλλακτικές επιλογές, ενώ όλες οι αναφορές μοντέλου ισχύουν, χωρίς όμως να δημιουργείται λογική σχέση μεταξύ τους.

Η προσθήκη σημασιολογικών επισημάνσεων σε μια περιγραφή WSDL μπορεί να πραγματοποιηθεί και εξωτερικά με την χρήση προεπεξεργασίας XSLT (Extensible Stylesheet Language Transformations) [89] ή χρησιμοποιώντας RDF (Resource Description Framework) [85].

3.2.2. Παράδειγμα

Το παράδειγμα που ακολουθεί χρησιμοποιείται στο πρότυπο των σημασιολογικών επισημάνσεων και αφορά μια διαδικτυακή υπηρεσία παραγγελιών. Η υπηρεσία έχει μια λειτουργία (operation) η οποία λαμβάνει σαν είσοδο τον αριθμό λογαριασμού του χρήστη και μια λίστα από κωδικούς (Universal Product Codes) που αντιστοιχούν στα αντικείμενα προς παραγγελία αλλά και την ποσότητα του κάθε αντικειμένου. Σαν έξοδος επιστρέφεται η κατάσταση της παραγγελίας, η οποία μπορεί να έχει γίνει αποδεκτή, να μην έχει γίνει αποδεκτή ή να βρίσκεται σε αναμονή [65].

Σκοπός του παραδείγματος είναι η επίδειξη των σημασιολογικών επισημάνσεων που έχουν προστεθεί στην περιγραφή της διαδικτυακής υπηρεσίας. Οι σημασιολογικές επισημάνσεις αυτές παρουσιάζονται με έντονη γραφή.

```
<wsdl:description
  targetNamespace="http://www.w3.org/2002/ws/sawSDL/spec/wsdl/order#"
  xmlns="http://www.w3.org/2002/ws/sawSDL/spec/wsdl/order#"
  xmlns:wsdl="http://www.w3.org/ns/wsdl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sawSDL="http://www.w3.org/ns/sawSDL">

  <wsdl:types>
    <xs:schema targetNamespace="http://www.w3.org/2002/ws/sawSDL/spec/wsdl/order#"
      elementFormDefault="qualified">
      <xs:element name="OrderRequest"

sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purchaseorder#OrderRequest"

sawSDL:loweringSchemaMapping="http://www.w3.org/2002/ws/sawSDL/spec/mapping/RDFOnt2Request.xml">

        <xs:complexType>
          <xs:sequence>
            <xs:element name="customerNo" type="xs:integer" />
            <xs:element name="orderItem" type="item" minOccurs="1" maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:complexType name="item">
        <xs:all>
          <xs:element name="UPC" type="xs:string" />
        </xs:all>
        <xs:attribute name="quantity" type="xs:integer" />
      </xs:complexType>
      <xs:element name="OrderResponse" type="confirmation" />
      <xs:simpleType name="confirmation"
```



```

sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purchaseorder#OrderConfirmation">

    <xs:restriction base="xs:string">
        <xs:enumeration value="Confirmed" />
        <xs:enumeration value="Pending" />
        <xs:enumeration value="Rejected" />
    </xs:restriction>
</xs:simpleType>
</xs:schema>
</wsdl:types>

<wsdl:interface name="Order"
    sawSDL:modelReference="http://example.org/categorization/products/electronics">
    <wsdl:operation name="order" pattern="http://www.w3.org/ns/wsdl/in-out"

sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purchaseorder#RequestPurchaseOrder">

    <wsdl:input element="OrderRequest" />
    <wsdl:output element="OrderResponse" />
</wsdl:operation>
</wsdl:interface>
</wsdl:description>

```

3.2.3. Αναζήτηση

Για την αναζήτηση περιγραφών SAWSDL πέρα από τα στοιχεία που έχει ήδη επισημανθεί πως χρειάζονται από τις WSDL 2.0 περιγραφές, είναι απαραίτητες οι αναφορές μοντέλων. Αυτό συμβαίνει γιατί προσδιορίζουν ουσιαστικά τη σημασιολογία των στοιχείων προσδίδοντας έτσι επιπλέον πληροφορία σε αυτά, σε αντίθεση με τα σχήματα αντιστοίχισης που δεν παρέχουν τόσο σημασιολογική πληροφορία, όσο χρησιμοποιούνται για να διευκολύνουν λειτουργίες πάνω στα δεδομένα.

3.3. OWL-S

3.3.1. Περιγραφή

Με τη διάδοση του σημασιολογικού ιστού εμφανίζονται νέοι τρόποι πρόσβασης και διαχείρισης του περιεχομένου αλλά και των υπηρεσιών του διαδικτύου. Οι χρήστες και οι πράκτορες λογισμικού (software agents), επωφελούμενοι από τις νέες δυνατότητες που προσφέρονται, είναι σε θέση να ανακαλύπτουν, να καλούν, να συνθέτουν και να παρακολουθούν τις διαδικτυακές υπηρεσίες, και μάλιστα, όταν είναι δυνατό, με αυτοματοποιημένο τρόπο.

Πέρα από τις σημασιολογικές πληροφορίες που εισάγονται με το προαναφερθέν πρότυπο (WSDL 2.0), η σημασιολογική επέκταση των διαδικτυακών υπηρεσιών πρέπει να υποστηριχθεί από οντολογίες (ontologies). Οι οντολογίες εμπλουτίζουν τις σημασιολογικές πληροφορίες

απεικονίζοντας τις συσχετίσεις μεταξύ των εννοιών που χρησιμοποιούνται για να αποδώσουν τη σημασιολογία των διαδικτυακών υπηρεσιών [66].

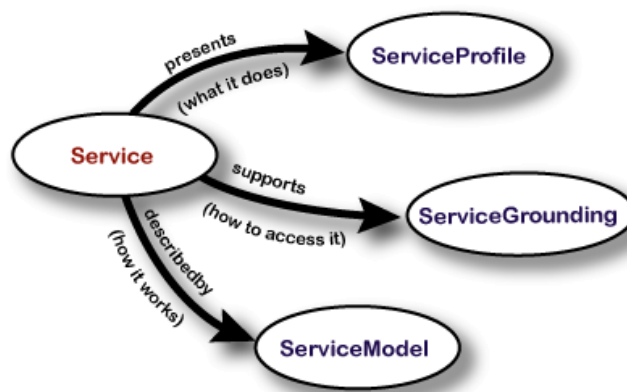
Η OWL-S (Semantic Web Ontology Language) [53], γνωστή ως DAML-S στις πρώτες εκδόσεις της, είναι μια οντολογία για διαδικτυακές υπηρεσίες βασισμένη στην OWL (Web Ontology Language) [52], η οποία παρέχει ένα σύνολο γλωσσικών δομών επισήμανσης (markup language constructs) για τη σημασιολογική περιγραφή των ιδιοτήτων και δυνατοτήτων των διαδικτυακών υπηρεσιών [53].

Οι σημασιολογικές πληροφορίες που εισάγονται με την OWL-S καθιστούν δυνατές ή διευκολύνουν σημαντικά λειτουργίες όπως:

- Αυτοματοποιημένη ανακάλυψη υπηρεσιών διαδικτύου (Automatic Web Service Discovery). Δηλαδή αναζήτηση και εύρεση υπηρεσιών διαδικτύου που προσφέρουν συγκεκριμένες υπηρεσίες με βάση κριτήρια καθορισμένα από τον χρήστη.
- Αυτοματοποιημένη κλήση υπηρεσιών διαδικτύου (Automatic Web Service Invocation). Κλήση μιας υπηρεσίας από κάποιο πρόγραμμα ή πράκτορα, με είσοδο μόνο μια περιγραφή που δηλώνει το τι πρέπει να κάνει η υπηρεσία.
- Αυτοματοποιημένη σύνθεση και συνεργασία υπηρεσιών διαδικτύου (Automatic Web service composition and interoperation), η οποία περιλαμβάνει αυτοματοποιημένη επιλογή, σύνθεση και συνεργασία υπηρεσιών διαδικτύου για την εκτέλεση μιας σύνθετης λειτουργίας, με είσοδο μια υψηλού επιπέδου περιγραφή του επιθυμητού αποτελέσματος.

Κάθε διαδικτυακή υπηρεσία αντιπροσωπεύεται σαν OWL-S περιγραφή από ένα μοναδικό στιγμιότυπο (instance) μιας κλάσης Service, η οποία έχει τις ιδιότητες (properties) *presents*, *describedBy*, και *supports*. Κάθε στιγμιότυπο της κλάσης Service, όσον αφορά τις προαναφερθείσες ιδιότητες, αποτελείται από τρεις επιμέρους περιγραφές: ServiceProfile, ServiceModel, ServiceGrounding.

Η δομή μίας οντολογίας που αφορά κάποια υπηρεσία, όπως φαίνεται στην Εικόνα 9, αποτελείται από τα τρία παραπάνω τμήματα (ServiceProfile, ServiceModel, ServiceGrounding), τα οποία καλύπτουν διαφορετικές ανάγκες περιγραφής της υπηρεσίας και παρέχουν την αντίστοιχη πληροφορία.



Εικόνα 9. Δομή της οντολογίας μιας διαδικτυακής υπηρεσίας.

Γενικά, το ServiceProfile παρέχει τις πληροφορίες που είναι απαραίτητες ώστε να διευκολύνεται η ανακάλυψη της διαδικτυακής υπηρεσίας από κάποιον πράκτορα, ενώ τα ServiceModel και ServiceGrounding παρέχουν πληροφορία που καθιστά τον πράκτορα ικανό να χρησιμοποιήσει την διαδικτυακή υπηρεσία, αφού την ανακαλύψει [53].

Εφόσον η περιγραφή της OWL-S στην παρούσα εργασία έχει σκοπό να προσδιορίσει και να τονίσει τα σημεία που θα διευκόλυναν την αναζήτηση μιας περιγραφής OWL-S, αναλυτικά θα εξεταστεί μόνο το ServiceProfile.

3.3.1.1. Service Profile

Το Service Profile επεξηγεί τι προσφέρει μια διαδικτυακή υπηρεσία κατά τρόπο που αποσαφηνίζει αν είναι κατάλληλη για την εξυπηρέτηση μια αίτησης. Περιλαμβάνει περιγραφές του αποτελέσματος της διαδικτυακής υπηρεσίας, των περιορισμών ως προς την καταλληλότητά της, της ποιότητας της υπηρεσίας που προσφέρεται και των απαιτήσεων που ο χρήστης πρέπει να πληροί για να χρησιμοποιήσει επιτυχώς την διαδικτυακή υπηρεσία.

Η OWL-S παρέχει έναν τρόπο αναπαράστασης του Service Profile με την χρήση μιας κλάσης *Profile*. Η συγκεκριμένη κλάση περιγράφει πληροφορία σχετική με τον οργανισμό που προσφέρει την υπηρεσία, τη λειτουργικότητα που προσφέρει η υπηρεσία και κάποια επιπλέον χαρακτηριστικά της.

Το Profile μιας διαδικτυακής υπηρεσίας χρησιμοποιείται για την δημοσίευσή της σε μητρώα (registries). Αν ο χρήστης κρίνει ότι η συγκεκριμένη διαδικτυακή υπηρεσία ικανοποιεί τις απαιτήσεις του, τότε χρησιμοποιεί το Process Model για να ελέγξει την αλληλεπίδραση με

αυτήν. Εφόσον τα Profile και Process Model αποτελούν ουσιαστικά διαφορετικές αναπαραστάσεις της ίδιας διαδικτυακής υπηρεσίας, χαρακτηριστικά όπως η είσοδος, η έξοδος, οι προϋποθέσεις και τα αποτελέσματά (effects) τους πρέπει να ταυτίζονται. Η OWL-S δεν είναι δεσμευτική πάνω σε αυτό, αλλά σε αντίθετη περίπτωση η υπηρεσία κάποια στιγμή κατά την εκτέλεση της θα παρουσιάσει σφάλματα.

Τα χαρακτηριστικά του Profile μπορούν να χωριστούν σε τέσσερις κατηγορίες:

Η πρώτη περιγράφει τον τρόπο που συνδέεται η κλάση Service Profile με την κλάση Service και την κλάση Process Model. Υπάρχει διμερής (two-way) σχέση ανάμεσά τους, που εκφράζεται από τις ιδιότητες *presents* και *presentedBy*:

- *presents*: περιγράφει την σχέση ανάμεσα σε ένα στιγμιότυπο μιας διαδικτυακής υπηρεσίας και σε ένα στιγμιότυπο του profile, και δηλώνει ότι η συγκεκριμένη υπηρεσία περιγράφεται από το εν λόγω.
- *presentedBy*: είναι το αντίστροφο της ιδιότητας *presents* και δηλώνει ότι το συγκεκριμένο profile περιγράφει μια υπηρεσία.

Η δεύτερη κατηγορία παρέχει πληροφορίες που απευθύνονται κυρίως στους χρήστες (human-readable) και δεν προορίζεται για επεξεργασία με αυτοματοποιημένο τρόπο. Συγκεκριμένα, παρέχει τα παρακάτω στοιχεία:

- *serviceName*: το όνομα της υπηρεσίας που προσφέρεται. Μπορεί να χρησιμοποιηθεί σαν αναγνωριστικό της υπηρεσίας. Το κάθε profile μπορεί να έχει το πολύ ένα στοιχείο *serviceName*.
- *textDescription*: παρέχει μια σύντομη περιγραφή της υπηρεσίας. Αναφέρει τι προσφέρει, τι χρειάζεται για να δουλέψει, και πρόσθετες πληροφορίες που ο πάροχος της υπηρεσίας πιθανόν να επιθυμεί να γνωστοποιήσει για αυτή στους μελλοντικούς καταναλωτές. Το κάθε profile μπορεί να έχει το πολύ ένα στοιχείο *textDescription*.
- *contactInformation*: παρέχει έναν μηχανισμό αναφοράς στα άτομα που είναι υπεύθυνα για την υπηρεσία. Το κάθε profile μπορεί να έχει ένα ή και παραπάνω στοιχεία *contactInformation*.

Στην τρίτη κατηγορία περιγράφεται η λειτουργική αναπαράσταση της υπηρεσίας με βάση τις εισόδους της (*inputs*), τις εξόδους της (*outputs*), τις προϋποθέσεις (*preconditions*) και τα

αποτελέσματα της (*effects*). Ένα βασικό κομμάτι του Profile είναι ο καθορισμός της λειτουργικότητας που παρέχει η υπηρεσία αλλά και των συνθηκών που πρέπει να ικανοποιούνται για επιτυχές αποτέλεσμα, ενώ επιπροσθέτως καθορίζει τι αποτελέσματα δίνει η χρήση της υπηρεσίας. Η κλάση OWL-S Profile αναπαριστά δύο πτυχές της λειτουργικότητας της υπηρεσίας: την μετατροπή πληροφορίας (*information transformation*) που αναπαριστάται από την είσοδο και την έξοδο αλλά και την αλλαγή της κατάστασης (*state change*) που παράγεται από την εκτέλεση της υπηρεσίας και αναπαριστάται από τις προϋποθέσεις και τα αποτελέσματα. Για παράδειγμα, έστω πως έχουμε μια υπηρεσία για την πώληση βιβλίων. Σαν είσοδο ζητά τον αριθμό της πιστωτικής κάρτας και την ημερομηνία λήξης, με την προϋπόθεση όμως ότι η πιστωτική κάρτα υπάρχει και μάλιστα δεν είναι υπερχρεωμένη. Η έξοδος της υπηρεσίας είναι η απόδειξη που πιστοποιεί την επιτυχή ολοκλήρωση της συναλλαγής και το αποτέλεσμα είναι η αλλαγή της κυριότητας και η φυσική παράδοση του βιβλίου από την αποθήκη του πωλητή στην διεύθυνση του αγοραστή. Για την περιγραφή των εισόδων, εξόδων, προϋποθέσεων και αποτελεσμάτων (*Inputs, Outputs, Preconditions, Effects - IOPEs*), η οντολογία Profile ορίζει τις παρακάτω ιδιότητες:

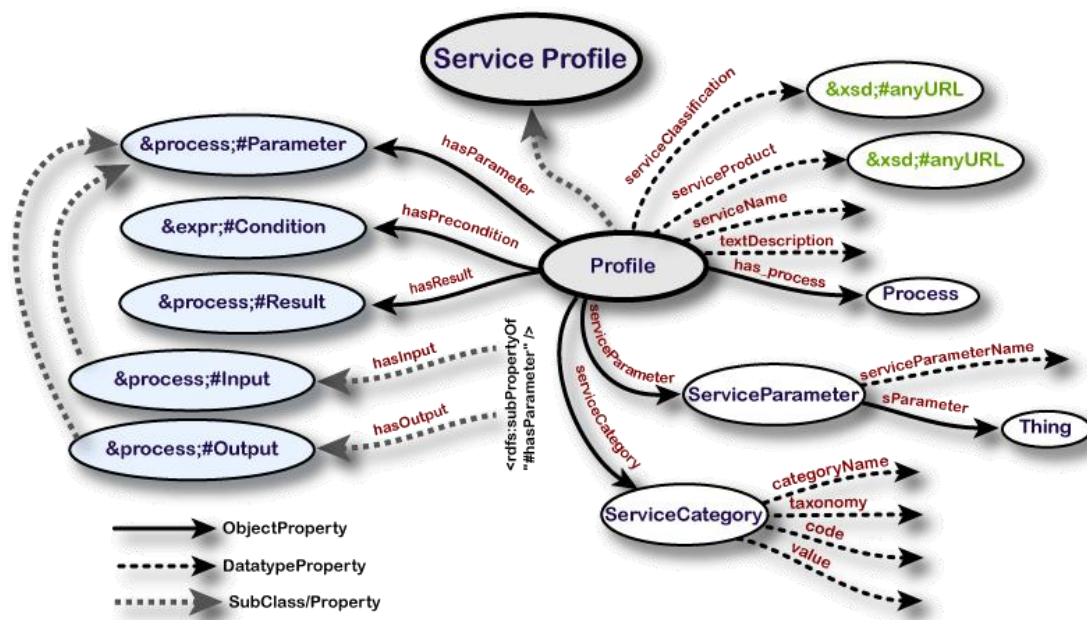
- *hasParameter*: δείχνει στα στιγμιότυπα των παραμέτρων, όπως αυτά έχουν οριστεί από την οντολογία Process.
- *hasInput*: δείχνει στις οντότητες των εισόδων όπως αυτές έχουν οριστεί από την οντολογία του Process.
- *hasOutput*: δείχνει (*ranges*) στις οντότητες των εξόδων όπως αυτές έχουν οριστεί από την οντολογία του Process.
- *hasPrecondition*: προσδιορίζει μια προϋπόθεση της υπηρεσίας και δείχνει (*ranges*) στην οντότητα της αντίστοιχης προϋπόθεσης που έχει οριστεί σύμφωνα με την οντολογία του Process.
- *hasResult*: καθορίζει τα αποτελέσματα της υπηρεσίας όπως αυτά ορίζονται από την κλάση *result* στην οντολογία του Process. Καθορίζει κάτω από ποιές συνθήκες παράγεται η έξοδος και τι αλλαγές στο domain παράγονται κατά την εκτέλεση της διαδικτυακής υπηρεσίας.

Στην τέταρτη και τελευταία κατηγορία περιγράφονται τα χαρακτηριστικά (*attributes*) του Profile, καθώς εκτός από τη λειτουργικότητα των υπηρεσιών, που περιγράφεται πλήρως από τα χαρακτηριστικά των προηγούμενων κατηγοριών, υπάρχουν και μη λειτουργικά χαρακτηριστικά

που επιθυμούν να γνωρίζουν οι πιθανοί χρήστες της διαδικτυακής υπηρεσίας. Αυτά περιλαμβάνουν τις εγγυήσεις ποιότητας που προσφέρουν οι διαδικτυακές υπηρεσίες, πιθανές κατηγοριοποιήσεις (classifications) της αλλά και πρόσθετες παραμέτρους. Οι ιδιότητες που ορίζονται για την περιγραφή των χαρακτηριστικών είναι:

- *serviceParameter*: επεκτάσιμη λίστα από ιδιότητες που μπορεί να συνοδεύει την περιγραφή ενός Profile. Η τιμή της είναι μια οντότητα της κλάσης ServiceParameter.
- *serviceCategory*: αναφέρεται σε μια εγγραφή (entry) κάποιας οντολογίας ή ταξονομίας (taxonomy) υπηρεσιών. Η τιμή της είναι ένα στιγμιότυπο της κλάσης ServiceCategory.

Η ιεραρχία των υποκλάσεων και των ιδιοτήτων της κλάσης Profile φαίνεται στην Εικόνα 10.



Εικόνα 10. Οι κλάσεις και οι ιδιότητες της κλάσης Profile.

3.3.1.2. Service Model

Ο τρόπος αλληλεπίδρασης με μια διαδικτυακή υπηρεσία ακολουθεί το πρότυπο κλήσης μιας (απομακρυσμένης) διεργασίας (process). Για αυτόν τον λόγο, η OWL-S 1.1 καθορίζει μια υποκλάση του ServiceModel, την Process, η οποία εξυπηρετεί την αλληλεπίδραση με την υπηρεσία.

Σαν ατομική διεργασία (atomic process) ορίζεται μια διεργασία η οποία λαμβάνει ένα (πιθανότατα σύνθετο) μήνυμα σαν είσοδο και επιστρέφει ένα μήνυμα σαν έξοδο. Αντίθετα, σαν

σύνθετη διεργασία (composite process) ορίζεται μια διεργασία με μεγαλύτερη λειτουργική πολυπλοκότητα, αφού διατηρεί και κάποιο είδος κατάστασης.

Μια διεργασία μπορεί να εξυπηρετεί δύο σκοπούς:

- να παράγει και να επιστρέφει νέα πληροφορία, βασισμένη στις πληροφορίες που της δόθηκαν και την κατάσταση του κόσμου, όπως την αντιλαμβάνεται τη δεδομένη χρονική στιγμή διαδικασία η οποία περιγράφεται από τις εισόδους και τις εξόδους της διεργασίας.
- να αλλάζει την κατάσταση του κόσμου, διαδικασία η οποία περιγράφεται από τα αποτελέσματα και τις προϋποθέσεις της διεργασίας.

Μια διεργασία μπορεί να έχει οποιονδήποτε αριθμό από εισόδους, εξόδους, προϋποθέσεις και αποτελέσματα, δεν είναι όμως απαραίτητο να δέχεται είσοδο.

Στο Process Model παρέχεται το σχήμα της περιγραφής για τις εισόδους, τις εξόδους, τις προϋποθέσεις και τα αποτελέσματα, ενώ το Profile της περιγραφής μιας διαδικτυακής υπηρεσίας αναφέρεται στις οντότητες αυτές.

3.3.1.3. Service Grounding

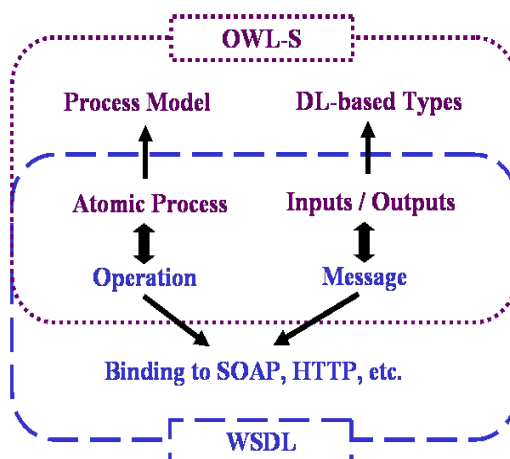
Οι πληροφορίες υλοποίησης (service grounding) καθορίζουν τις λεπτομέρειες που αφορούν στον τρόπο με τον οποίο ένας πράκτορας μπορεί να χρησιμοποιήσει μια υπηρεσία. Τυπικά, καθορίζουν ένα πρωτόκολλο επικοινωνίας, τους τύπους των μηνυμάτων και άλλες λεπτομέρειες όπως είναι οι αριθμοί θυρών (port numbers) που χρησιμοποιούνται για την επικοινωνία με μια υπηρεσία. Πρέπει ακόμα να καθορίζουν, για κάθε σημασιολογικό τύπο εισόδου και εξόδου που ορίζεται στο ServiceModel, έναν μονοσήμαντο τρόπο ανταλλαγής στοιχείων δεδομένων αυτού του τύπου με την υπηρεσία, δηλαδή τις τεχνικές σειριοποίησης (serialization) που χρησιμοποιούνται.

Η βασική λειτουργία του grounding είναι να ορίσει τον τρόπο με τον οποίο οι αφηρημένες (abstract) είσοδοι και έξοδοι μπορούν να παρασταθούν σαν συγκεκριμένα (concrete) μηνύματα που μεταφέρουν τις εισόδους και τις εξόδους σε κάποια κατάλληλη μορφοποίηση.

Για την υλοποίηση του grounding της OWL-S προτείνεται συνήθως η χρήση της WSDL. Μάλιστα, υπάρχει αντιστοιχία ανάμεσα στο binding της WSDL και στο grounding της OWL-S. Με αυτό τον τρόπο το όφελος είναι διπλό αφού χρησιμοποιείται το Process Model και η

εκφραστικότητα των κλάσεων της OWL-S, ενώ ταυτόχρονα χρησιμοποιείται ένα πρότυπο το οποίο είναι γενικά αποδεκτό και δοκιμασμένο όπως η WSDL.

Στην Εικόνα 11 φαίνεται η συσχέτιση μεταξύ μιας περιγραφής WSDL και της αντίστοιχης περιγραφής OWL-S.



Εικόνα 11. Συσχέτιση μεταξύ μιας περιγραφής WSDL και της αντίστοιχης περιγραφής OWL-S.

3.3.2. Παράδειγμα

Ακολουθεί παράδειγμα ενός Service Profile μιας διαδικτυακής υπηρεσίας σε OWL-S. Περιέχει το όνομα και την περιγραφή της υπηρεσίας, καθώς και πληροφορίες για τις παραμέτρους της. Η συγκεκριμένη διαδικτυακή υπηρεσία δέχεται σαν παράμετρο εισόδου τον αριθμό ISBN ενός βιβλίου, εντοπίζει το βιβλίο και επιστρέφει σαν παραμέτρους εξόδου τον τίτλο και τον συγγραφέα του. Η διαδικτυακή υπηρεσία ανήκει στο σύνολο δοκιμών (test set) OWLS-TC [54], το οποίο περιέχει και τις οντολογίες books.owl και portal.owl, στις οποίες ανήκουν οι έννοιες Book, Author και ISBN που χρησιμοποιούνται για να καθορίσουν τη σημασιολογία των παραμέτρων εισόδου και εξόδου. Τμήματα των οντολογιών books.owl και portal.owl που χρησιμοποιεί το παράδειγμα παρουσιάζονται παρακάτω. Στην οντολογία books.owl φαίνονται οι κλάσεις / έννοιες Author και Book, ενώ, στην οντολογία portal.owl ανήκει η έννοια ISBN. Η απεικόνιση έχει γίνει με χρήση της παρακάτω χρωματικής κωδικοποίησης:

Όνομα	■	Είσοδοι	■
Περιγραφή	■	Έξοδοι	■

3.3.2.1. ISBNtoBookAuthor.owl

```
<?xml version="1.0" encoding="WINDOWS-1252"?>
<rdf:RDF xmlns:owl = "http://www.w3.org/2002/07/owl#"
xmlns:rdfs = "http://www.w3.org/2000/01/rdf-schema#"
xmlns:rdf = "http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:service = "http://www.daml.org/services/owl-s/1.1/Service.owl#"
xmlns:process = "http://www.daml.org/services/owl-s/1.1/Process.owl#"
xmlns:profile = "http://www.daml.org/services/owl-s/1.1/Profile.owl#"
xmlns:grounding = "http://www.daml.org/services/owl-s/1.1/Grounding.owl#"
xml:base = "http://127.0.0.1/services/1.1/isbn_bookauthor_service.owl">
<owl:Ontology rdf:about="">
<owl:imports rdf:resource="http://127.0.0.1/ontology/Service.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/Process.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/Profile.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/Grounding.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/books.owl" />
<owl:imports rdf:resource="http://127.0.0.1/ontology/portal.owl" />
</owl:Ontology>
<service:Service rdf:ID="ISBN_BOOKAUTHOR_SERVICE">
<service:presents rdf:resource="#ISBN_BOOKAUTHOR_PROFILE"/>
<service:describedBy rdf:resource="#ISBN_BOOKAUTHOR_PROCESS_MODEL"/>
<service:supports rdf:resource="#ISBN_BOOKAUTHOR_GROUNDING"/>
</service:Service>
<profile:Profile rdf:ID="ISBN_BOOKAUTHOR_PROFILE">
<service:isPresentedBy rdf:resource="#ISBN_BOOKAUTHOR_SERVICE"/>
<profile:serviceName xml:lang="en"> BookProviderService </profile:serviceName>
<profile:textDescription xml:lang="en">
    This service provides a book and its author, on the given ISBN.
</profile:textDescription>
<profile:hasInput rdf:resource="#_ISBN"/>
<profile:hasOutput rdf:resource="#_BOOK"/>
<profile:hasOutput rdf:resource="#_AUTHOR"/>
<profile:has_process rdf:resource="ISBN_BOOKAUTHOR_PROCESS" /></profile:Profile>
<process:ProcessModel rdf:ID="ISBN_BOOKAUTHOR_PROCESS_MODEL">
<service:describes rdf:resource="#ISBN_BOOKAUTHOR_SERVICE"/>
<process:hasProcess rdf:resource="#ISBN_BOOKAUTHOR_PROCESS"/>
</process:ProcessModel>
<process:AtomicProcess rdf:ID="ISBN_BOOKAUTHOR_PROCESS">
<process:hasInput rdf:resource="#_ISBN"/>
<process:hasOutput rdf:resource="#_BOOK"/>
<process:hasOutput rdf:resource="#_AUTHOR"/>
</process:AtomicProcess>
<process:Input rdf:ID="_ISBN">
<process:parameterType rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://127.0.0.1/ontology/portal.owl#ISBN</process:parameterType>
<rdfs:label></rdfs:label>
</process:Input>
<process:Output rdf:ID="_BOOK">
<process:parameterType rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://127.0.0.1/ontology/portal.owl#Book</process:parameterType>
<rdfs:label></rdfs:label>
</process:Output>
<process:Output rdf:ID="_AUTHOR">
<process:parameterType rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">
    http://127.0.0.1/ontology/books.owl#Author</process:parameterType>
<rdfs:label></rdfs:label>
</process:Output>
<grounding:Wsd1Grounding rdf:ID="ISBN_BOOKAUTHOR_GROUNDING">
<service:supportedBy rdf:resource="#ISBN_BOOKAUTHOR_SERVICE"/>
</grounding:Wsd1Grounding>
</rdf:RDF>
```

3.3.2.2. books.owl

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [
<!ENTITY books.owl "http://127.0.0.1/ontology/books.owl">
<!ENTITY owl "http://www.w3.org/2002/07/owl#">
```

```

<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<!ENTITY simplified_sumo.owl "http://127.0.0.1/ontology/simplified_sumo.owl">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
]>
<rdf:RDF xml:base="&books.owl;"
xmlns:owl="&owl;"
xmlns:rdf="&rdf;"
xmlns:rdfs="&rdfs;">

<!-- Ontology Information -->
<owl:Ontology rdf:about=""
rdfs:label="Book Ontology"
owl:versionInfo="1.0">
<rdfs:comment>An ontology containing information about books</rdfs:comment>
<owl:imports>
<owl:Ontology rdf:about="&simplified_sumo.owl;" />
</owl:imports>
</owl:Ontology>

<!-- Classes -->
.
.
.
<owl:Class rdf:about="#Author">
<rdfs:subClassOf rdf:resource="#Person"/>
</owl:Class>
.
<owl:Class rdf:about="#Book">
<rdfs:subClassOf rdf:resource="#Monograph"/>
<rdfs:subClassOf>
<owl:Restriction rdf:nodeID="b10">
<owl:allValuesFrom rdf:resource="#Author"/>
<owl:onProperty rdf:resource="#writtenBy"/>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:allValuesFrom rdf:resource="#Book-Type"/>
<owl:onProperty rdf:resource="#hasType"/>
</owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
<owl:Restriction>
<owl:allValuesFrom rdf:resource="#Title"/>
<owl:onProperty rdf:resource="#isTitled"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>
.
.
.
<owl:Class rdf:about="&simplified_sumo.owl;#CorpuscularObject"/>
<owl:Class rdf:about="&simplified_sumo.owl;#Object"/>

<!-- Object Properties -->
<owl:ObjectProperty rdf:about="#contains"/>
<owl:ObjectProperty rdf:about="#datePublished"/>
<owl:ObjectProperty rdf:about="#hasGenre"/>
<owl:ObjectProperty rdf:about="#hasGrade"/>
<owl:ObjectProperty rdf:about="#hasName"/>
<owl:ObjectProperty rdf:about="#hasSize"/>
<owl:ObjectProperty rdf:about="#hasType"/>
<owl:ObjectProperty rdf:about="#isTitled"/>
<owl:ObjectProperty rdf:about="#publishedBy"/>
<owl:ObjectProperty rdf:about="#timePublished"/>
<owl:ObjectProperty rdf:about="#writtenBy"/>
</rdf:RDF>

```

3.3.2.3. portal.owl

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rdf:RDF [

```

```

<!ENTITY owl "http://www.w3.org/2002/07/owl#">
<!ENTITY portal "http://127.0.0.1/ontology/portal.owl#">
<!ENTITY portal.owl "http://127.0.0.1/ontology/portal.owl">
<!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#">
<!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#">
<!ENTITY support.owl "http://127.0.0.1/ontology/support.owl">
<!ENTITY xsd "http://www.w3.org/2001/XMLSchema#">
]>
<rdf:RDF xml:base="&portal.owl;"
  xmlns:owl="&owl;"
  xmlns:portal="&portal;"
  xmlns:rdf="&rdf;"
  xmlns:rdfs="&rdfs;">

  <!-- Ontology Information -->
  <owl:Ontology rdf:about=""
    owl:versionInfo="2.0">
    <rdfs:label>AKT Reference Ontology (Portal Ontology)</rdfs:label>
    <rdfs:seeAlso rdf:resource="http://d3e.open.ac.uk/akt/2002/ref-onto.html"/>
    <owl:imports>
      <owl:Ontology rdf:about="&support.owl;" />
    </owl:imports>
  </owl:Ontology>

  <!-- Classes -->
  .
  .
  .
  .
  <owl:Class rdf:about="#ISBN">
    <rdfs:subClassOf rdf:resource="#Publication-Number"/>
  </owl:Class>
  .
  .
  .
</rdf:RDF>

```

3.3.3. Αναζήτηση

Κατά τρόπο αντίστοιχο με τα άλλα δύο πρότυπα, η σε βάθος εξέταση των περιγραφών διαδικτυακών υπηρεσιών OWL-S ακολουθείτε από τον προσδιορισμό της χρήσιμης για αναζήτηση πληροφορίας.

Εφόσον τα δύο στοιχεία που προσδιορίθηκαν αρχικά στις περιγραφές WSDL ήταν εξωτερικά, δηλαδή δεν άνηκαν στην περιγραφή, είναι δυνατό να προσδιορισθούν και για τις περιγραφές OWL-S. Η χρησιμότητα τους δεν αναλύεται ξανά αφού εξυπηρετούν τον ίδιο ακριβώς σκοπό όπως στις περιγραφές WSDL. Τα δύο αυτά στοιχεία είναι:

- Η διεύθυνση όπου εντοπίστηκε η περιγραφή διαδικτυακής υπηρεσίας.
- Το όνομα του αρχείου της περιγραφής.

Τα υπόλοιπα στοιχεία μιας περιγραφής OWL-S που παρουσιάζουν ιδιαίτερο ενδιαφέρον για την περιγραφή της λειτουργίας μιας διαδικτυακής υπηρεσίας βρίσκονται στο περιεχόμενο της περιγραφής. Συγκεκριμένα, η πληροφορία αυτή παρουσιάζεται στο τμήμα Profile μιας περιγραφής OWL-S, του οποίου όπως προσδιορίζεται και από το πρότυπο αυτοσκοπός είναι η

διευκόλυνση της ανακάλυψης μιας περιγραφής. Τα στοιχεία του service profile που μπορούν να χρησιμοποιηθούν για την περιγραφή μιας διαδικτυακής υπηρεσίας είναι:

- Το όνομα της υπηρεσίας που προσφέρεται (serviceName).

Το όνομα δρα συνήθως σαν αναγνωριστικό της υπηρεσίας και μάλιστα είναι περιγραφικού χαρακτήρα, είθισται δηλαδή να προσδιορίζει όσο είναι δυνατό την λειτουργία και τον σκοπό της υπηρεσίας.

- Η τεκμηρίωση-περιγραφή της υπηρεσίας (textDescription).

Παρέχει μια σύντομη περιγραφή της υπηρεσίας. Αναφέρει τι προσφέρει, τι χρειάζεται για να δουλέψει, και πρόσθετες πληροφορίες που ο πάροχος της υπηρεσίας πιθανόν να επιθυμεί να γνωστοποιήσει για αυτή στους μελλοντικούς καταναλωτές.

- Οι είσοδοι και οι έξοδοι της υπηρεσίας (inputs - outputs).

Οι είσοδοι και οι έξοδοι μιας υπηρεσίας αποτελούν σημαντικό στοιχείο αφού πρακτικά προσδιορίζουν την λειτουργικότητα που προσφέρεται ορίζοντας τον τρόπο με τον οποίο η εισερχόμενη πληροφορία μετατρέπεται από την υπηρεσία.

- Οι προϋποθέσεις και τα αποτελέσματα της υπηρεσίας (preconditions - results)

Οι προϋποθέσεις και τα αποτελέσματα όπως προσδιορίζουν και τα ονόματά τους καθορίζουν υπό ποιες συνθήκες η υπηρεσία μπορεί να χρησιμοποιηθεί και αν αυτό είναι εφικτό τι αντίκτυπο θα έχει στην προηγούμενη κατάσταση.

Σχεδιασμός Μηχανής Αναζήτησης

Το παρόν Κεφάλαιο παρουσιάζει την ανάλυση του συστήματος που αποτελεί το αντικείμενο της παρούσας πτυχιακής εργασίας. Πρόκειται μια μηχανή αναζήτησης εξειδικευμένη για περιγραφές διαδικτυακών υπηρεσιών, που στοχεύει στην βελτίωση των αποτελεσμάτων σε σχέση με υπάρχουσες λύσεις. Ταυτόχρονα, παρουσιάζεται η αρχιτεκτονική του συστήματος, τα συστατικά στοιχεία του, οι βασικές λειτουργικές απαιτήσεις του καθενός, καθώς και οι προκλήσεις που πρέπει να αντιμετωπιστούν για την επιτυχή επίλυση του προβλήματος της αναζήτησης περιγραφών διαδικτυακών υπηρεσιών.

4.1. Περιγραφή Συστήματος

4.1.1. Στόχος

Η διεξοδική εξέταση των υπάρχοντων μηχανών αναζήτησης, τόσο γενικού σκοπού όσο και στοχευμένων, ως προς την αναζήτηση διαδικτυακών υπηρεσιών, όπως περιγράφηκε στο κεφάλαιο 3, υποδεικνύει πως η έρευνα στο συγκεκριμένο τομέα είναι ακόμα σε πρώιμο στάδιο, ενώ υπάρχουν δυνατότητες ανάπτυξης επιπλέον λειτουργικότητας. Μια εξειδικευμένη μηχανή αναζήτησης διαδικτυακών υπηρεσιών που λαμβάνει υπόψη της τις πρόσφατες εξελίξεις τόσο στον χώρο της αναζήτησης πληροφορίας στο διαδίκτυο, όσο και στον χώρο των προτύπων διαδικτυακών υπηρεσιών, μπορεί να παρέχει βελτιστοποιημένες υπηρεσίες ανακάλυψης και αναζήτησης.

Η βελτιστοποίηση αυτή σε σχέση με τις μηχανές αναζήτησης γενικού σκοπού έγκειται στην επίτευξη καλύτερης ακρίβειας (precision) και ανάκλησης (recall) στα αποτελέσματα των αναζητήσεων. Σε αντίθεση με τις μηχανές αναζήτησης γενικού σκοπού, μια μηχανή αναζήτησης για διαδικτυακές υπηρεσίες επιστρέφει μόνο περιγραφές υπηρεσιών, χωρίς αυτές να βρίσκονται ανάμεσα σε χιλιάδες άλλα αποτελέσματα, ενώ η βαθμονόμηση και άρα η κατάταξη των αποτελεσμάτων προκύπτει λαμβάνοντας υπόψη μόνο εκείνα τα στοιχεία μιας περιγραφής που έχουν χρήσιμη πληροφορία, κάτι που προς το παρόν δεν κάνουν οι μηχανές αναζήτησης γενικού σκοπού για τις περιγραφές διαδικτυακών υπηρεσιών. Επιπλέον, παρόλο που οι μηχανές αναζήτησης γενικού σκοπού παρέχουν σημαντικές δυνατότητες παραμετροποίησης της

αναζήτησης, αυτό γίνεται με γνώμονα στοιχεία διαφορετικά από εκείνα που θα βοηθούσαν στην εύρεση καταλλήλων διαδικτυακών υπηρεσιών.

Αναφορικά με τις εξειδικευμένες μηχανές αναζήτησης, μερικές εκ των οποίων παρέχουν υπηρεσίες που αντιμετωπίζουν τα παραπάνω μειονεκτήματα των συμβατικών μηχανών αναζήτησης, η παρούσα εργασία στοχεύει στην βελτίωσή τους όσον αφορά τον εντοπισμό των διαδικτυακών υπηρεσιών στον Παγκόσμιο Ιστό, την εκμετάλλευση της σημασιολογικής πληροφορίας που περιέχεται στις περιγραφές τους, αλλά και τους αλγορίθμους δεικτοδότησης (indexing) και βαθμολόγησης (ranking) που χρησιμοποιούν για την κατάταξη των αποτελεσμάτων που παρουσιάζουν στους χρήστες.

Το αντικείμενο της παρούσας πτυχιακής εργασίας, που ερευνά τη δημιουργία μιας νέας εξειδικευμένης μηχανής αναζήτησης περιγραφών διαδικτυακών υπηρεσιών, υποστηρίζεται όχι μόνο από την παρατήρηση των τρόπων που επιδέχονται βελτίωση οι υπάρχουσες λύσεις αναζήτησης διαδικτυακών υπηρεσιών, αλλά και από την προσδοκώμενη εδραίωση του προτύπου των διαδικτυακών υπηρεσιών σαν το πλέον διαδεδομένο τρόπο επικοινωνίας μεταξύ κατανεμημένων συστημάτων, καθώς τα πλεονεκτήματα που προσφέρουν είναι ιδιαίτερα σημαντικά. Επιπλέον κίνητρο για την έρευνα σχετικά με την ανάκτηση διαδικτυακών υπηρεσιών αποτελεί ο συνεχώς αυξανόμενος αριθμός τους, ο οποίος καθιστά ιδιαίτερα δύσκολο τον εντοπισμό της απαιτούμενης λειτουργικότητας με μη αυτοματοποιημένο τρόπο ή με απλή περιήγηση.

Η συγκεκριμένη μηχανή αναζήτησης φιλοδοξεί να εντοπίζει, να δεικτοδοτεί και να παρέχει ως αποτελέσματα περιγραφές διαδικτυακών υπηρεσιών με αυξημένη ανάκληση και ακρίβεια, μέσω μιας εύχρηστης, σύγχρονης διεπαφής, που θα δίνει την δυνατότητα στους χρήστες να αναζητήσουν υπηρεσίες με διάφορα κριτήρια, εκμεταλλευόμενοι και την σημασιολογική πληροφορία των περιγραφών τους. Οι περιγραφές των διαδικτυακών υπηρεσιών που συγκεντρώνονται θα συμμορφώνονται με τα πλέον διαδεδομένα πρότυπα τόσο για σημασιολογική όσο και για μη σημασιολογική περιγραφή που περιγράφηκαν στην προηγούμενη ενότητα, δηλαδή τα WSDL και OWL-S. Εκθέτοντας με εύκολο τρόπο τις διαδικτυακές υπηρεσίες και την σημασιολογική πληροφορία τους, η μηχανή αναζήτησης θα αποτελέσει ένα χρήσιμο εργαλείο για την εύρεση και χρήση δεδομένων που εκμεταλλεύονται τις πλέον ανερχόμενες τάσεις στον Παγκόσμιο Ιστό, συμβάλλοντας έτσι και στην εδραίωση και διάδοση των προτύπων αυτών.

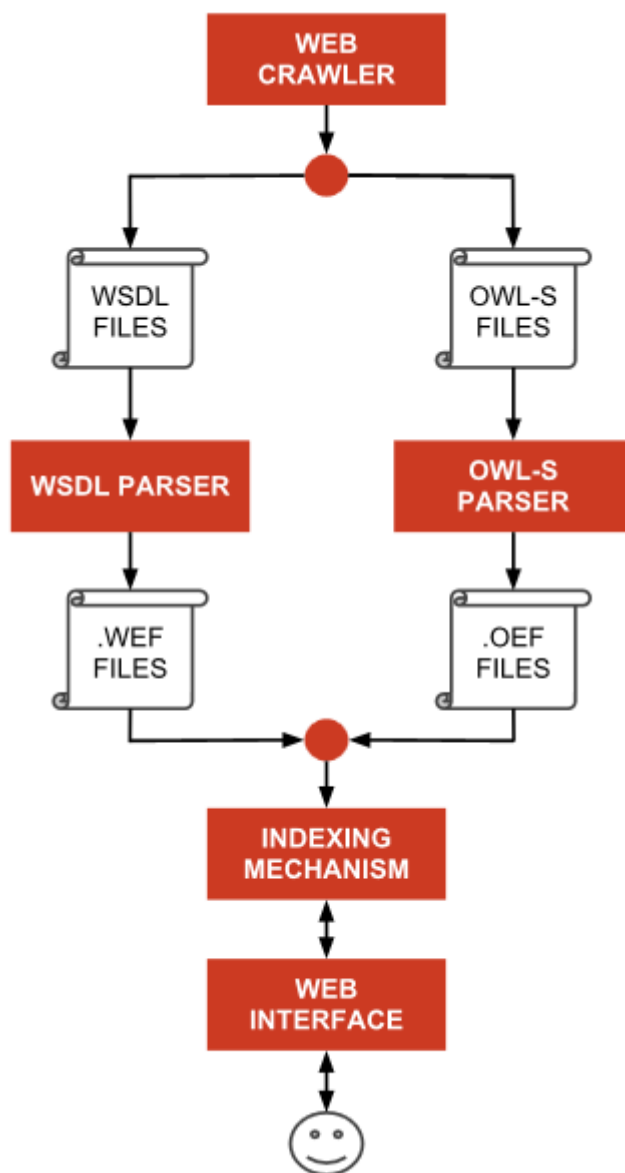
4.1.2. Αρχιτεκτονική Συστήματος

Η αρχιτεκτονική μιας μηχανής αναζήτησης, την οποία ακολουθεί η παρούσα εργασία, περιλαμβάνει τρία βασικά συστατικά στοιχεία (components):

- το στοιχείο που «περιηγείται» στον Παγκόσμιο Ιστό και συλλέγει έγγραφα, δηλαδή τον crawler,
- το στοιχείο που οργανώνει και δεικτοδοτεί την συλλεγόμενη πληροφορία ώστε να επιτρέπεται η αποδοτική αναζήτησή της, δηλαδή το index, και
- το στοιχείο που επιτρέπει την αλληλεπίδραση με τον χρήστη, δηλαδή το interface.

Στην περίπτωση της παρούσας μηχανής αναζήτησης, στα τρία αυτά υποσυστήματα προστίθενται επιπλέον συστατικά στοιχεία τα οποία δρουν ως μεσάζοντες ανάμεσα στον crawler και το index, και φιλτράρουν μόνο την χρήσιμη πληροφορία από τις περιγραφές διαδικτυακών υπηρεσιών που έχουν συλλεχθεί, οι parsers.

Στην Εικόνα 12 που ακολουθεί φαίνεται διαγραμματικά ο τρόπος με τον οποίο τα προαναφερθέντα υποσυστήματα αλληλεπιδρούν μεταξύ τους, καθώς και η σειρά με την οποία συμβαίνει αυτό.



Εικόνα 12. Αρχιτεκτονική μηχανής αναζήτησης για διαδικτυακές υπηρεσίες.

Η αρχιτεκτονική του συστήματος είναι αρθρωτή (modular), επομένως τα υποσυστήματα που το απαρτίζουν μπορούν να αντιμετωπιστούν σαν ξεχωριστές, ανεξάρτητες οντότητες. Η ιδιότητα αυτή προσδίδει στα υποσυστήματα ορισμένα σημαντικά πλεονεκτήματα που καθιστούν την ανεξαρτησία τους ιδιαίτερα ελκυστική. Πρώτον, εφόσον κάθε τμήμα είναι ανεξάρτητο, είναι δυνατό κάποιο από αυτά να αντικατασταθεί με εύκολο τρόπο και χωρίς την ανάγκη αλλαγών στα υπόλοιπα υποσυστήματα. Η αντικατάσταση αυτή ενός τμήματος μπορεί να υπαγορεύεται από διάφορους παράγοντες μεταξύ των οποίων βρίσκονται η ανάγκη διεξαγωγής συγκριτικών πειραμάτων ανάμεσα σε συστήματα που διεκπεραιώνουν την ίδια εργασία, η παρώχηση ενός

τμήματος λόγω των ερευνητικών εξελίξεων και η μη λειτουργία του με τον επιθυμητό τρόπο. Δεύτερον, η ανεξαρτησία κάθε τμήματος προάγει την επαναχρησιμοποίηση, ώστε να μπορούν να χρησιμοποιηθούν και από τρίτους με τη φιλοσοφία του ανοικτού κώδικα (open source).

Ο τρόπος λειτουργίας καθενός από τα υποσυστήματα καθορίζεται μέσω αρχείων ρυθμίσεων (configuration files). Με αυτόν τον τρόπο δίνεται η δυνατότητα ελέγχου της λειτουργίας των υποσυστημάτων χωρίς την ανάγκη απευθείας επέμβασης στον κώδικα. Τα αρχεία ρυθμίσεων για κάθε υποσύστημα θα μελετηθούν κατά περίπτωση στη συνέχεια.

Ακολουθεί η αναλυτική περιγραφή και ο προσδιορισμός των λειτουργικών απαιτήσεων κάθε υποσυστήματος. Σε αυτές περιλαμβάνονται οι αρχικές απαιτήσεις που προσδιορίστηκαν αλλά και οι απαιτήσεις που προέκυψαν κατά την υλοποίηση και την δοκιμαστική λειτουργία του συστήματος. Ο τρόπος με τον οποίο οι απαιτήσεις αυτές ικανοποιήθηκαν και οι τεχνικές και μέθοδοι που χρησιμοποιήθηκαν για την αντιμετώπιση των προβλημάτων που εμφανίσθηκαν κατά την ανάπτυξη του συστήματος περιγράφονται αναλυτικά στο κεφάλαιο 5.

4.2. Web Crawler

Σκοπός κάθε μηχανής αναζήτησης είναι η παροχή εύκολης και άμεσης πρόσβασης σε πληροφορίες. Έτσι, πρώτο βήμα για την κατασκευή μιας μηχανής αναζήτησης αποτελεί η μεθοδική συλλογή των απαραίτητων δεδομένων που θα διατίθενται στους χρήστες μέσω της αναζήτησης. Στην πιο συνηθισμένη περίπτωση, κατά την οποία τα δεδομένα αποτελούν τμήμα του Παγκόσμιου Ιστού, ο όγκος των δεδομένων ανάμεσα στα οποία βρίσκονται οι περιγραφές διαδικτυακών υπηρεσιών είναι τόσο μεγάλος που η αναζήτηση πρέπει να πραγματοποιείται με αυτοματοποιημένο τρόπο. Το συστατικό στοιχείο του συστήματος που επιτελεί αυτή την λειτουργία ονομάζεται web crawler.

Οι συμβατικές μηχανές αναζήτησης έχουν σαν στόχο την περιήγηση σε όσο μεγαλύτερο μέρος του Παγκοσμίου Ιστού είναι δυνατόν και την ευρετηριοποίηση της πληροφορίας που συναντούν στον δρόμο τους. Αυτό διαφέρει θεμελιωδώς από το στόχο της εξειδικευμένης μηχανής αναζήτησης που κατασκευάζεται στο πλαίσιο της συγκεκριμένης εργασίας, ο οποίος αφορά την ευρετηριοποίηση μόνο των περιγραφών διαδικτυακών υπηρεσιών που υπάρχουν διαθέσιμες στον Παγκόσμιο Ιστό.

Είναι εμφανές πως εφόσον η διαφοροποίηση αυτή οδηγεί σε διαφορετικό τρόπο λειτουργίας του βασικού συστατικού στοιχείου συλλογής της πληροφορίας, δηλαδή του web crawler. Η βασική διαφοροποίηση έγκειται στη μη αποθήκευση του συνόλου των πληροφοριών στις οποίες αποκτά πρόσβαση ο crawler κατά την περιήγησή του, αφού στόχος δεν είναι η αδιάκριτη ευρετηριοποίηση πληροφορίας. Κάθε σελίδα που συναντάται να μην διαβάζεται και σαρώνεται για συνδέσμους, ανατροφοδοτώντας έτσι τον crawler για να συνεχίσει την περιήγηση, αλλά δεν αποθηκεύεται. Οι μόνες σελίδες που αποθηκεύονται είναι εκείνες που πρόκειται να ευρετηριοποιηθούν, δηλαδή εκείνες οι οποίες περιέχουν περιγραφές διαδικτυακών υπηρεσιών, σε κάποια από τα πρότυπα που εξετάζονται στο πλαίσιο της παρούσας εργασίας. Πρέπει να σημειωθεί ότι όσον αφορά το εύρος της περιήγησης, δεν υπάρχει διαφοροποίηση σε σχέση με έναν τυπικό crawler, καθώς εκτιμάται ότι όσο μεγαλύτερο μέρος του Ιστού εξεταστεί, τόσες περισσότερες πιθανότητες υπάρχουν για τον εντοπισμό ενδιαφέρουσας πληροφορίας, που στη συγκεκριμένη περίπτωση είναι περιγραφές διαδικτυακών υπηρεσιών.

Πρέπει να σημειωθεί ότι, αν και υπάρχουν διαθέσιμες υλοποιήσεις web crawlers ανοικτού λογισμικού, στο πλαίσιο της παρούσας εργασίας προτιμήθηκε ο σχεδιασμός και η υλοποίηση εξ αρχής ενός νέου crawler, καθώς οι ανάγκες ήταν ιδιαίτερα εξειδικευμένες. Με τον τρόπο αυτό, υποστηρίζεται πλήρης προσαρμογή και παραμετροποίηση του crawler σε σχέση με τον σκοπό της συλλογής περιγραφών διαδικτυακών υπηρεσιών.

Εφόσον ο Παγκόσμιος Ιστός μπορεί να αναπαρασταθεί σαν κατευθυνόμενος γράφος, ο crawler μπορεί να επιτελέσει αναζήτηση βασιζόμενος σε κάποιον γνωστό και καλά εδραιωμένο αλγόριθμο αναζήτησης. Ο βασικός αλγόριθμος που χρησιμοποιείται στην παρούσα εργασία είναι η Αναζήτηση κατά Πλάτος (Breadth First Search - BFS) [11]. Η κάθε ιστοσελίδα αντιμετωπίζεται σαν κόμβος του γράφου, ενώ οι υπερσύνδεσμοι που αυτή περιέχει παράγουν τα «παιδιά» του κάθε κόμβου. Σύμφωνα με τον συγκεκριμένο αλγόριθμο, έχοντας ως κόμβο εκκίνησης κάποια συγκεκριμένη ιστοσελίδα, ο crawler θα πρέπει να επισκεφθεί πρώτα όλες τις σελίδες στις οποίες δείχνουν οι υπερσύνδεσμοι της αρχικής, πριν προχωρήσει με την αναζήτηση στο επόμενο επίπεδο που βρίσκεται σε μεγαλύτερο βάθος.

Εκτός από τον αλγόριθμο αναζήτησης, οι βασικές λειτουργικές απαιτήσεις για τον web crawler είναι εκείνες που διαδραματίζουν καθοριστικό ρόλο στην επίτευξη του στόχου του, δηλαδή την συλλογή όσο το δυνατόν περισσότερων περιγραφών διαδικτυακών υπηρεσιών στην παρούσα περίπτωση. Η επίτευξη αυτού του στόχου καθορίζεται από τις πολιτικές (policies) που καθορίζουν τη συμπεριφορά ενός web crawler, οι οποίες μελετήθηκαν στην υποενότητα 1.2.3

και κάθε φορά πρέπει να προσαρμόζονται ανάλογα με τον σκοπό του crawling και τα διαθέσιμα μέσα. Η προσαρμογή των εν λόγω πολιτικών για τους σκοπούς της αναζήτησης περιγραφών διαδικτυακών υπηρεσιών περιγράφεται στις επόμενες ενότητες.

4.2.1. Πολιτικές

4.2.1.1. Επιλογή (Selection)

Ο ρυθμός με τον οποίο ένας web crawler εξετάζει ιστοσελίδες είναι πολύ σημαντικός. Όσο ταχύτερος είναι ο crawler, τόσες περισσότερες σελίδες θα προσπελαύνει και τόσες περισσότερες σελίδες που ικανοποιούν τις προδιαγραφές του θα συναντά. Λαμβάνοντας υπόψη πως ακόμη και οι μεγαλύτερες μηχανές αναζήτησης παγκοσμίως έχουν ευρετηριοποιήσει μόνο ένα μέρος του Παγκόσμιου Ιστού, σε συνδυασμό με τους περιορισμούς του μηχανήματος που θα τρέξει ο crawler και τους χρονικούς περιορισμούς που τίθενται από την περίοδο εκπόνησης της πτυχιακής εργασίας, είναι εμφανές πως ο crawler θα προσπελάσει έναν σημαντικό αλλά περιορισμένο αριθμό ιστοσελίδων. Επομένως, πρέπει να καταβληθεί κάθε δυνατή προσπάθεια ώστε

- a) να μεγιστοποιηθεί ο αριθμός των σελίδων που έχει επισκεφθεί συνολικά ο crawler
- b) να μεγιστοποιηθεί το ποσοστό των ενδιαφέρουσων σελίδων μέσα στο σύνολο των επισκεφθέντων (περιγραφές διαδικτυακών υπηρεσιών)

Ο συνολικός αριθμός των σελίδων που έχει επισκεφθεί ο crawler κατά βάση επηρεάζεται από την ταχύτητα με την οποία ο crawler προσπελαύνει τις σελίδες, η οποία πρέπει να είναι η μέγιστη δυνατή που μπορεί να επιτευχθεί στο μηχανήμα στο οποίο τρέχει και με τη διαδικτυακή σύνδεση η οποία είναι διαθέσιμη. Εξίσου σημαντικός είναι και ο τρόπος με τον οποίο ο crawler θα επιλέγει τις επόμενες προς επίσκεψη σελίδες από το σύνολο των σελίδων που έχει στην λίστα προς επίσκεψη (μέτωπο αναζήτησης).

Όντας ένα πρόγραμμα που περιπλανάται και διαβάζει σελίδες από τον αχανή Παγκόσμιο Ιστό, ο web crawler είναι καταδικασμένος να συναντήσει κάθε λογής προβλήματα. Τα προβλήματα αυτά μερικές φορές έχουν να κάνουν με την ίδια τη φύση του περιεχομένου του Παγκόσμιου Ιστού. Για παράδειγμα, κάθε σελίδα των ενός τυπικού forum παρέχει έως και εκατοντάδες συνδέσμους προς άλλες σελίδες του ίδιου forum, με παρόμοιου είδους περιεχόμενο. Αυτό έχει ως αποτέλεσμα την αυξημένη πιθανότητα εγκλωβισμού μέσα στο forum ενός web crawler που

δεν χρησιμοποιεί κάποια πολιτική αποφυγής ιστοσελίδων. Η κατάσταση είναι παρόμοια και σε άλλες σελίδες παρόμοιου τύπου, όπου το περιεχόμενο παράγεται εν μέρει δυναμικά, όπως είναι τα κοινωνικά δίκτυα και τα μεγάλα ηλεκτρονικά καταστήματα. Επιπροσθέτως, στον Παγκόσμιο Ιστό υπάρχουν προσχεδιασμένες παγίδες ειδικά για crawlers [71]. Οι παγίδες αυτές είναι σχεδιασμένες επίτηδες ώστε να έχουν το ίδιο αποτέλεσμα εγκλωβισμού που περιγράφηκε παραπάνω. Χρησιμοποιώντας διάφορες τεχνικές προσπαθούν είτε να εγκλωβίσουν τον web crawler σε ατέρμονες βρόχους, μειώνοντας την ταχύτητα και την απόδοση του, είτε να τον κάνουν να τερματίσει τη λειτουργία του.

Είναι σαφές πως για να είναι σταθερός και άρα αποτελεσματικός ένας crawler πρέπει να αντιμετωπίζει αποτελεσματικά αυτά τα προβλήματα. Αν ο crawler σταματούσε την λειτουργία του κάθε φορά που συναντούσε ένα τέτοιο πρόβλημα, η πρόοδος που τελικά θα σημείωνε θα ήταν ανεπαρκής. Για αυτόν τον λόγο, πρέπει να προβλεφθεί μηχανισμός που θα ελέγχει την πρόοδο του crawler και δεν θα επιτρέπει την παγίδευσή του, αλλά και κάποια τεχνική που θα του δίνει την δυνατότητα να κρίνει εάν ένα σύνολο σελίδων είναι επιβλαβές ή χωρίς κάποιο αναμενόμενο κέρδος.

Εφόσον εκ των προτέρων είναι αδύνατο να είναι γνωστό κάθε πρόβλημα που μπορεί να εμφανισθεί, απαιτείται επίσης ένας τρόπος που θα διασφαλίζει ότι η πρόοδος του crawler δεν θα χάνεται σε περίπτωση αντιμετώπισης κάποιου απρόβλεπτου προβλήματος, δηλαδή κάποιος μηχανισμός αποθήκευσης της τρέχουσας κατάστασης και του ιστορικού του crawler.

Επιπλέον, ο προσδιορισμός του ενδιαφέροντος / χρησιμότητας μιας ιστοσελίδας, αν δηλαδή η συγκεκριμένη σελίδα πληρεί τις προϋποθέσεις του crawler ώστε να αποθηκευτεί, είναι σημαντικός παράγοντας στην επίτευξη του στόχου. Η μέθοδος ή ο συνδυασμός μεθόδων που θα χρησιμοποιηθούν θα πρέπει να διασφαλίζουν πως δεν θα χαθούν ενδιαφέρουσες ιστοσελίδες με περιγραφές διαδικτυακών υπηρεσιών από αυτές που θα συναντηθούν.

Τέλος, πρέπει να εξεταστούν μηχανισμοί που επιτρέπουν στον crawler να αλλάζει την προκαθορισμένη πορεία του μέσα στον γράφο αναζήτησης του Ιστού, βασισμένοι σε ευριστικούς κανόνες που ενισχύουν την πιθανότητα να εντοπίσει περισσότερες ενδιαφέρουσες σελίδες πιο νωρίς κατά την αναζήτηση. Ο μηχανισμός ο οποίος εφαρμόστηκε στην παρούσα πτυχιακή εργασία αφορά την τοπική αλλαγή του αλγορίθμου αναζήτησης και περιγράφεται στο κεφάλαιο 5.

4.2.1.2. Ευγένεια (Politeness)

Οι πόροι που απαιτεί ένας web crawler αλλά και η βασική λειτουργία του, δηλαδή η αποστολή αιτήσεων σε διαδικτυακούς εξυπηρετές (web servers) με ταχύ και αυτοματοποιημένο τρόπο, είναι δυνατό να προκαλέσουν προβλήματα στους εξυπηρετές. Επιπλέον, είναι πιθανό, αν δεν τεθούν περιορισμοί, ο crawler να επισκεφθεί σελίδες στις οποίες δεν είναι θεμιτό να έχει πρόσβαση, ή οι δημιουργοί τους κρίνουν την επίσκεψη τέτοιου είδους προγραμμάτων ανεπιθύμητη. Για τους παραπάνω λόγους, είναι ανάγκη να ληφθούν κάποια μέτρα που θα αποτρέψουν την εφαρμογή των προαναφερθεισών προβληματικών συμπεριφορών και κακών πρακτικών.

Το ειδικά θεσπισμένο πρωτόκολλο αποκλεισμού ρομπότ (robots exclusion protocol) [60] εξυπηρετεί αυτόν τον σκοπό, λαμβάνοντας υπόψη τις επιθυμίες των ιδιοκτητών των ιστοτόπων. Σύμφωνα με το πρωτόκολλο, κατά την επίσκεψη ενός web crawler σε μια ιστοσελίδα, ο πρώτος απαιτείται αρχικά να αναζητήσει ένα συγκεκριμένο αρχείο με το όνομα “robots.txt”. Στο συγκεκριμένο αρχείο οι διαχειριστές του ιστοχώρου (web masters) χρησιμοποιώντας καθορισμένη σύνταξη μπορούν να προσδιορίσουν αν και σε ποιούς web crawlers επιτρέπουν την πρόσβαση, καθώς και τμήματα του ιστοτόπου τους που τυχόν δεν επιθυμούν να προσπελαστούν. Το συγκεκριμένο πρωτόκολλο εφαρμόζεται στην παρούσα εργασία για την αποφυγή της επίσκεψης του crawler σε ιστοχώρους στους οποίους είναι ανεπιθύμητος.

Επιπλέον, εξετάστηκε η δυνατότητα να οριστεί ένα χρονικό διάστημα ανάμεσα στις αιτήσεις προς τον ίδιο εξυπηρετή, ώστε να μην προστίθεται σημαντικός φόρτος από τον crawler. Αυτό απορρίφθηκε ως πρακτική, καθώς κρίθηκε ότι έρχεται σε αντίφαση με την λειτουργική απαίτηση ταχύτητας του crawler. Επομένως, ο έλεγχος των αιτήσεων του crawler προς έναν συγκεκριμένο εξυπηρετή αφήνεται στην ευχέρεια του διαχειριστή.

Τέλος, εφόσον οι πόροι που απαιτεί ο crawler στη συγκεκριμένη περίπτωση ανήκουν στο Πανεπιστήμιο, αποφασίστηκε να πραγματοποιηθούν δοκιμές με διαφορετικές ρυθμίσεις ώστε να επιλεγούν οι βέλτιστες, εκείνες δηλαδή που δεν προκαλούν κάποιο πρόβλημα στην κοινότητα του Πανεπιστημίου σε συνεννόηση με το Κέντρο Πληροφορικής και Δικτύων.

4.2.1.3. Επανεπίσκεψη (Revisit)

Σε τυπικές μηχανές αναζήτησης, η επικαιρότητα των ευρετηριοποιημένων ιστοσελίδων αποτελεί ζωτικό ζήτημα. Ωστόσο, κάτι τέτοιο δεν ισχύει για μια μηχανή αναζήτησης για διαδικτυακές υπηρεσίες, καθώς η ανάπτυξη των υπηρεσιών και η δημοσίευση των περιγραφών τους δεν είναι δεδομένα τα οποία αλλάζουν με εξαιρετικά συχνό ρυθμό. Επομένως, η πολιτική επανεπίσκεψης

δεν αποτέλεσε πρωτεύον ζήτημα για τον τρόπο λειτουργίας του web crawler που δημιουργήθηκε.

Η μόνη περίπτωση κατά την οποία έχει νόημα η κατά βούληση συχνή επανεπίσκεψη ιστοσελίδων, είναι για ιστοσελίδες που λειτουργούν σαν κατάλογοι διαδικτυακών υπηρεσιών, οι οποίες ανανεώνονται συχνά με νέες υπηρεσίες, όπως τα UDDIs [75]. Πρέπει να σημειωθεί, ωστόσο, ότι τα δημόσια UDDIs έχουν σταματήσει να υποστηρίζονται από τις εταιρείες που τα έφεραν στο προσκήνιο και πλέον δεν ανανεώνονται.

Λαμβάνοντας υπόψη τα παραπάνω, και δεδομένου ότι η επανεπίσκεψη ιστοσελίδων μειώνει την απόδοση του crawler, κρίθηκε ότι δεν υπάρχει λόγος να εφαρμοστεί κάποια συγκεκριμένη πολιτική επανεπίσκεψης ιστοσελίδων. Αυτό ωστόσο δεν αποκλείει την επανεπίσκεψη, αφού ούτως ή άλλως κατά την περιήγηση είναι πολύ πιθανό ο crawler να επισκεφθεί ξανά μια ιστοσελίδα πέρα από την πρώτη φορά κάποια στιγμή ώστε να γνωρίζει το ενημερωμένο περιεχόμενό της.

Ενώ όμως η επανεπίσκεψη σελίδων δεν αποτέλεσε λειτουργική απαίτηση, ιδιαίτερη σημασία δόθηκε σε μηχανισμούς που θα απέτρεπαν τον crawler από την συνεχή επίσκεψη των ίδιων σελίδων. Αυτό συνέβη επειδή αφενός υπάρχει το ενδεχόμενο εγκλωβισμού του web crawler σε κύκλους και αφετέρου με την επανεπίσκεψη ο αριθμός των διαφορετικών ιστοσελίδων που επισκέπτεται ο crawler μειώνεται δραστικά, πράγμα που μειώνει την αποδοτικότητά του.

4.2.1.4. Παραλληλοποίηση (Parallelization)

Η ταχύτητα ενός web crawler αποτελεί πολύ σημαντικό παράγοντα για την απόδοσή του. Η παραλληλοποίηση είναι ο ευκολότερος και πιο αποτελεσματικός τρόπος να μεγιστοποιηθεί ο ρυθμός προσπέλασης ιστοσελίδων, και άρα η απόδοση. Η παραλληλοποίηση όμως είναι μια τεχνική που υπονοεί την εκτέλεση τμημάτων της εφαρμογής σε διαφορετικούς πόρους. Δεδομένων των συνθηκών, η ύπαρξη ενός μόνο ηλεκτρονικού υπολογιστή για την εκτέλεση της εφαρμογής υπαγορεύει την ανάγκη προσδιορισμού μιας άλλης μεθόδου για την βελτιστοποίηση της αποδοτικότητας. Για τον σκοπό αυτό αποφασίστηκε η πολυνηματική (multi-threaded) υλοποίηση του web crawler, τεχνική που θα εξασφάλιζε την ταυτόχρονη εκτέλεση τμήματος του αλγορίθμου από πολλά νήματα, συμβάλλοντας έτσι στον σκοπό.

4.2.2. Ρυθμίσεις (Configuration)

Στο πλαίσιο των πειραματισμών ώστε να προσδιοριστούν οι παράμετροι εκείνες που βελτιστοποιούν την απόδοση του crawler και ενδεχόμενης επαναχρησιμοποίησης του προγράμματος στο μέλλον, απαιτήθηκε η δημιουργία ενός αρχείου ρυθμίσεων (configuration file). Τα περισσότερα προγράμματα που κυκλοφορούν προσφέρουν ένα αρχείο τέτοιου τύπου, παρέχοντας έτσι στον χρήστη τη δυνατότητα να τροποποιήσει με εύκολο τρόπο την λειτουργία του, σύμφωνα με τις δικές τους ανάγκες, χωρίς να χρειάζεται να καταφύγει σε αλλαγή του κώδικα.

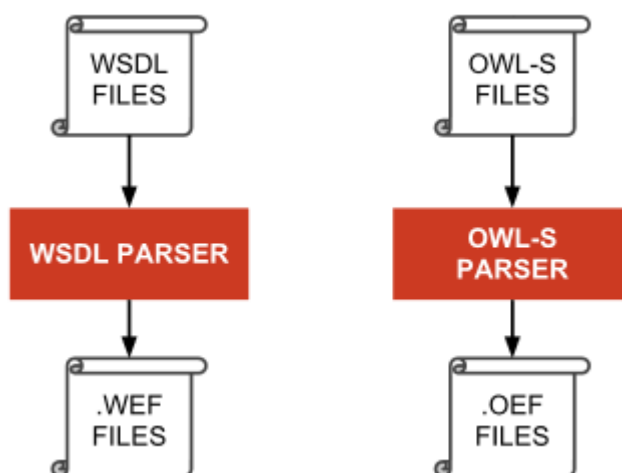
Οι ρυθμίσεις που γίνονται μέσω αυτού του αρχείου περιλαμβάνουν τα ακόλουθα:

- Αριθμός των νημάτων (threads)
- Ιστοσελίδες από τις οποίες θα εκκινηθεί η αναζήτηση
- Μηχανισμός αποθήκευσης κατάστασης και ιστορικού
- Διάφορα μεγέθη που επηρεάζουν την απόδοση του crawler.

4.3. Parsing

Σκοπός του συγκεκριμένου υποσυστήματος είναι η επεξεργασία των περιγραφών διαδικτυακών υπηρεσιών που συλλέχθηκαν από τον crawler και το φιλτράρισμα της ενδιαφέρουσας πληροφορίας που περιέχουν, η οποία στη συνέχεια θα ευρετηριοποιηθεί στο ευρετήριο.

Αν και η εργασία που επιτελεί είναι αναγκαία, το συγκεκριμένο υποσύστημα δεν είναι απαραίτητο καθώς η λειτουργία του θα μπορούσε να ενσωματωθεί στις εργασίες του υποσυστήματος του ευρετηρίου πριν την ευρετηριοποίηση. Ωστόσο, στο πλαίσιο της πολιτικής που ακολουθείται και αφορά τον σχεδιασμό της αρχιτεκτονικής του συστήματος με βάση ανεξαρτησία των υποσυστημάτων, έτσι ώστε να μπορούν να επαναχρησιμοποιηθούν ή να αντικατασταθούν, αποφασίστηκε να δημιουργηθεί ένα ξεχωριστό υποσύστημα για την συγκεκριμένη εργασία. Στην Εικόνα 13 που ακολουθεί φαίνονται η είσοδος και η έξοδος του υποσυστήματος.



Εικόνα 13. Είσοδος και έξοδος του υποσυστήματος.

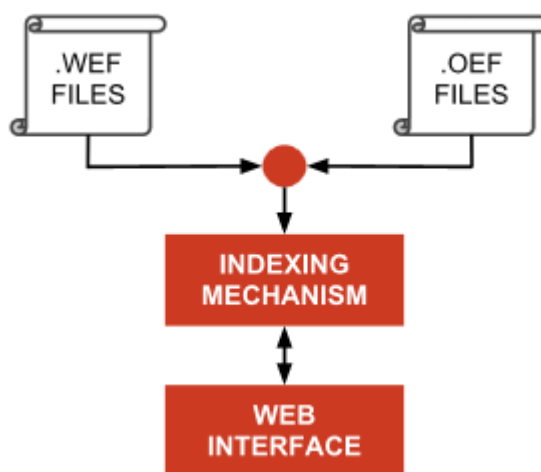
Εφόσον οι περιγραφές που συλλέχθηκαν είναι δύο τύπων, WSDL και OWL-S, υπάρχουν δύο διακριτοί parsers. Και για τις δύο περιπτώσεις, κατά την επεξεργασία τους από τους parsers, οι περιγραφές διαδικτυακών υπηρεσιών υπόκεινται σε δύο βασικές διεργασίες επεξεργασίας:

- Η πρώτη πολύ σημαντική διεργασία έχει να κάνει με την απομόνωση των τμημάτων εκείνων της περιγραφής που θεωρούνται χρήσιμα για την αναζήτηση όπως είναι η είσοδος και η έξοδος της υπηρεσίας. Τα τμήματα αυτά προσδιορίστηκαν αναλυτικά κατά την ανάλυση των προτύπων των περιγραφών που αναζητούνται και αποτελούν τα δεδομένα που θα ευρετηριοποιηθούν. Για αυτό τον λόγο είναι απαραίτητη η αποτελεσματική απομόνωσή τους από τις περιγραφές χωρίς να χάνεται καθόλου πληροφορία.
- Η δεύτερη διεργασία αφορά την οργάνωση της συγκεκριμένης πληροφορίας με τρόπο που να διευκολύνει την επεξεργασία της σε επόμενο στάδιο και τελικά της αποθήκευσής της σε αρχεία, τα οποία στην συνέχεια θα επεξεργαστεί το ευρετήριο.

Όπως φαίνεται στην Εικόνα 12, οι διεργασίες των parsers πραγματοποιούνται αμέσως μετά την συλλογή των περιγραφών διαδικτυακών υπηρεσιών και είναι ανεξάρτητες από κάθε διεργασία που ακολουθεί, πραγματοποιούνται δηλαδή ασύγχρονα (offline). Αυτό, σε συνδυασμό με το γεγονός πως οι parsers επεξεργάζονται μία μόνο φορά την κάθε περιγραφή, και δεν χρειάζεται επιπλέον επεξεργασία για κάθε επερώτηση, δεν καθιστούν επιτακτική την ανάγκη για υψηλή ταχύτητα επεξεργασίας.

4.4. Indexing

Αφού συλλεχθούν τα δεδομένα που η μηχανή αναζήτησης θα διαθέτει στους χρήστες της, είναι ανάγκη αυτά να υποστούν προεπεξεργασία και να ταξινομηθούν με τέτοιο τρόπο που θα επιτρέπει την γρήγορη και με ακρίβεια ανάκτηση πληροφορίας. Υπεύθυνος για την διεκπαιρέωση αυτής της απαιτητικής και υψηλής σπουδαιότητας εργασίας είναι ο μηχανισμός του index. Ο τρόπος με τον οποίο το ευρετήριο αλληλεπιδρά με το υπόλοιπο σύστημα φαίνεται στην Εικόνα 14.



Εικόνα 14. Τρόπος αλληλεπίδρασης του ευρετηρίου με το υπόλοιπο σύστημα.

Στο εισαγωγικό Κεφάλαιο 1, όπου κάθε τμήμα μιας μηχανής αναζήτησης εξετάστηκε λεπτομερώς παρατηρήθηκε πως υπάρχουν ευρετήρια διαφόρων τύπων. Για την κατασκευή του ευρετηρίου της μηχανής αναζήτησης για διαδικτυακές υπηρεσίες επιλέχθηκε το ευρετήριο να είναι ανεστραμμένο και να βρίσκεται στην μνήμη cache. Οι επιλογές αυτές υποστηρίζονται από το γεγονός ότι τα ανεστραμμένα ευρετήρια προσφέρουν άμεση πρόσβαση στα έγγραφα που περιλαμβάνουν τις λέξεις που περιέχει μια επερώτηση (query), απλοποιώντας και επιταχύνοντας έτσι την διαδικασία, ενώ το μέγεθος του ευρετηρίου που θα κατασκευαστεί αναμένεται να είναι τέτοιο ώστε να επιτρέπει την διατήρηση του στην μνήμη cache.

Η ταχύτητα με την οποία ο χρήστης θα έχει τη δυνατότητα να ανακαλύψει μια υπηρεσία αλλά και η ποιότητα των εμφανιζόμενων αποτελεσμάτων αποτελούν τις λειτουργίες εκείνες της μηχανής αναζήτησης οι οποίες θα την καταστήσουν αποδεκτή, οπότε γίνεται κατανοητό πως θα πρέπει να ληφθεί μέριμνα ώστε να επιτευχθούν βέλτιστα αποτελέσματα αναφορικά με τα δύο χαρακτηριστικά.

Φυσικά, με τον ίδιο τρόπο που το υποσύστημα του ευρετηρίου θα πρέπει να είναι συμβατό με τα αρχεία που εξάγουν οι parsers, θα πρέπει να είναι και συμβατό με το άλλο υποσύστημα με το οποίο επικοινωνεί άμεσα και περιγράφεται παρακάτω, το web interface. Η επικοινωνία ανάμεσα στα δύο μπορεί να επιτευχθεί είτε απευθείας είτε χρησιμοποιώντας κάποιον ενδιάμεσο παράγοντα όπως τις διαδικτυακές υπηρεσίες (web services), με την ταχύτητα βέβαια να διαδραματίζει καθοριστικό ρόλο.

Σε αυτό το σημείο υπενθυμίζεται πως οι περιγραφές διαδικτυακών υπηρεσιών που συγκεντρώθηκαν έχουν ήδη υποστεί προεπεξεργασία από τους parsers. Το ευρετήριο πλέον καλείται να επεξεργαστεί αρχεία ενδιάμεσης μορφής για περιγραφές WSDL και OWL-S στα οποία περιέχεται μόνο χρήσιμη ως προς την αναζήτηση πληροφορία. Πρώτη λοιπόν απαίτηση για το ευρετήριο είναι να υποστηρίζει και τους δύο αυτούς τύπους αρχείων τόσο στο διάβασμά τους και την περαιτέρω επεξεργασία τους όσο και στην ευρετηριοποίησή τους.

Η προεπεξεργασία των δύο αυτών ενδιάμεσων τύπων αρχείων αποτελεί δραστηριότητα εξέχουσας σημασίας, καθώς συνεπάγεται αυξημένη ταχύτητα και ακρίβεια των αποτελεσμάτων που θα επιστρέφει το ευρετήριο. Αφού ο μηχανισμός διαβάσει τις ήδη επεξεργασμένες από τους parsers περιγραφές, στη συνέχεια καλείται να τις επεξεργαστεί περαιτέρω ώστε να τις φέρει στην βέλτιστη δυνατή μορφή για να εισαχθούν στο ευρετήριο. Η προεπεξεργασία χρειάζεται να περιλαμβάνει διεργασίες όπως ο διαχωρισμός των όρων, η επιλογή της ρίζας των όρων, η αφαίρεση των μη χρήσιμων stop words κλπ. Όπως αναφέρθηκε η συγκεκριμένη διεργασία βοηθά στην επίτευξη καλύτερης ταχύτητας και ακρίβειας αλλά δεν αποτελεί την μοναδική μέριμνα για τη διασφάλισή τους.

Για την επίτευξη καλύτερης ακρίβειας στα επιστρεφόμενα αποτελέσματα, θα χρησιμοποιηθούν διάφορα δεδομένα που χρειάζεται να εξάγει το ευρετήριο όπως η συχνότητα του κάθε όρου και το βάρος του ανάλογα με την θέση του στο έγγραφο. Αποφασίσθηκε να χρησιμοποιηθεί το βάρος, αφού προσθέτει αρκετά σημαντική πληροφορία η οποία μπορεί να υποβοηθήσει την αναζήτηση, καθώς προσδιορίζει το γεγονός πως δεν είναι όλοι οι όροι που εμφανίζονται σε ένα έγγραφο το ίδιο σημαντικοί. Επιπλέον, η χρησιμοποίηση βαρών σε συνδυασμό με τα υπόλοιπα

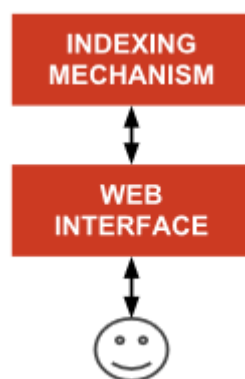
στοιχεία κάνουν δυνατή την προσθήκη μιας επιπλέον δυνατότητας στην μηχανή αναζήτησης, την σύνθετη αναζήτηση υπηρεσιών με βάση διάφορα κριτήρια που μπορεί να θέσει ο χρήστης, και όχι μόνο με βάση απλών λέξεων-κλειδιών.

Ας σημειωθεί πως το ανεστραμμένο ευρετήριο δεν πρόκειται να δημιουργείται εξ' αρχής αλλά με βάση ένα αρχικό κανονικό ευρετήριο. Η επιλογή αυτή γίνεται καθώς η ύπαρξη του επιταχύνει την φάση της ανανέωσης του ευρετηρίου. Αν και στο πλαίσιο της πτυχιακής δεν θα αναπτυχθεί η λειτουργία της ανανέωσης του ευρετηρίου ενώ συνεχίζει να λειτουργεί, ενδέχεται στο μέλλον αυτό να αλλάξει οπότε αυτή η ιδιότητα θα αποτελέσει χρήσιμο βοήθημα.

4.5. Web Interface

Το web interface αποτελεί την γραφική διεπαφή του συστήματος με την χρήση της οποίας πραγματοποιείται η αλληλεπίδραση της μηχανής αναζήτησης με τον χρήστη.

Εφόσον πρόκειται για διαδικτυακή εφαρμογή, η διεπαφή δεν θα μπορούσε παρά να είναι μια δυναμική ιστοσελίδα, προσβάσιμη από οποιονδήποτε web browser. Σε αυτήν, όπως φαίνεται και στην Εικόνα 15, ακολουθώντας την γνώριμη λειτουργικότητα των μηχανών αναζήτησης, ο χρήστης θα έχει την δυνατότητα να εισάγει επερωτήσεις (queries), είτε απλές είτε σύνθετες, και να λαμβάνει τα αποτελέσματα που επιστρέφει το ευρετήριο με εύληπτο τρόπο.



Εικόνα 11. Τρόπος αλληλεπίδρασης του web interface με το υπόλοιπο σύστημα.

Επιπλέον, θα πρέπει να παρέχει πληροφορίες για την μηχανή αναζήτησης και για τον τρόπο χρήσης της. Τέλος, θα πρέπει να δίνεται η δυνατότητα στον χρήστη να αιτηθεί την ευρετηριοποίηση μιας περιγραφής διαδικτυακής υπηρεσίας παρέχοντας την διεύθυνση όπου αυτή βρίσκεται.

Εμφανισιακά η σελίδα απαιτείται να είναι απλή ώστε να εστιάζει στην αναζήτηση. Οι δυνατότητες που προσφέρονται πρέπει να παρουσιάζονται με τον πρότυπο τρόπο που χρησιμοποιούν οι υπάρχουσες μηχανές αναζήτησης ώστε να είναι εμφανείς με διαισθητικό τρόπο, χωρίς να απαιτείται προσπάθεια από τον χρήστη.

Ο χρόνος απόκρισης στις ερωτήσεις των χρηστών της μηχανής αναζήτησης δεν πρέπει να είναι μεγάλος.

Τέλος, είναι απαραίτητο η ιστοσελίδα να είναι σε μορφή που να διευκολύνει την επικοινωνία με το index. Η αλληλεπίδραση ανάμεσα στα δύο υποσυστήματα μπορεί να επιτευχθεί είτε απευθείας είτε χρησιμοποιώντας κάποιο ενδιάμεσο επίπεδο. Μια τεχνική πραγματοποίησης της δεύτερης μεθόδου είναι η χρήση διαδικτυακών υπηρεσιών. Με αυτό τον τρόπο η διασύνδεση ανάμεσα στα δύο υποσυστήματα γίνεται τελείως ανεξάρτητη, ενώ η διαδικτυακή υπηρεσία μπορεί να επαναχρησιμοποιηθεί σε άλλα συστήματα.

ΚΕΦΑΛΑΙΟ 5

Υλοποίηση Μηχανής Αναζήτησης

Στο παρόν Κεφάλαιο περιγράφεται η υλοποίηση της λειτουργικότητας που παρουσιάστηκε στο Κεφάλαιο 4. Για το κάθε υποσύστημα παρουσιάζεται η βασική λειτουργικότητά του, τα ιδιαίτερα χαρακτηριστικά που αναπτύχθηκαν κατά την υλοποίησή του, και οι τεχνολογίες, εργαλεία και περιβάλλοντα ανάπτυξης που χρησιμοποιήθηκαν. Επιπλέον, για κάθε υποσύστημα παρουσιάζεται και περιγράφεται το διάγραμμα λειτουργίας του.

5.1. Web Crawler

Σύμφωνα με τους Shkapenyuk και Suel [68], η δημιουργία ενός crawler υψηλών προδιαγραφών που περιηγείται σε εκατοντάδες εκατομμύρια σελίδες σε μερικές εβδομάδες παρουσιάζει ένα σύνολο προκλήσεων που αφορούν τον σχεδιασμό του συστήματος, την αποδοτική είσοδο, έξοδο και πιθανώς επικοινωνία με τον δίσκο, τη σταθερότητά του και την διαχειρισσιμότητά του. Γίνεται λοιπόν αντιληπτό πως εξαιτίας της ίδιας της φύσης του συγκεκριμένου προγράμματος κατά τον σχεδιασμό και την υλοποίηση του ήταν αναμενόμενο να εμφανισθούν πολλές προκλήσεις και προβλήματα που αφορούσαν διαφορετικές παραμέτρους και έχριζαν αντιμετώπισης.

Στην ενότητα αυτή αρχικά περιγράφεται η υλοποίηση του web crawler για περιγραφές διαδικτυακών υπηρεσιών, όπως αναπτύχθηκε στο πλαίσιο της παρούσας εργασίας. Στην συνέχεια ερευνώνται σε βάθος οι επιλογές που έγιναν για την υλοποίηση του web crawler και η επιχειρηματολογία πίσω από αυτές, τα προβλήματα που εμφανίσθηκαν και οι λύσεις που εφαρμόστηκαν, καθώς και οι τεχνολογίες που χρησιμοποιήθηκαν. Τέλος, παρουσιάζεται το διάγραμμα ροής του web crawler το οποίο επεξηγείται αναλυτικά.

5.1.1. Βασική λειτουργικότητα

Ο web crawler για διαδικτυακές υπηρεσίες εκκινεί την λειτουργία του επισκεπτόμενος μια αρχική λίστα από URLs που καλούνται *σπόροι* (*seeds*). Κάθε φορά που ο web crawler επισκέπτεται μια ιστοσελίδα ελέγχει αν αυτή αποτελεί περιγραφή μιας διαδικτυακής υπηρεσίας,

και σε αυτή την περίπτωση την αποθηκεύει με σκοπό την μελλοντική της ευρετηρίαση. Σε αντίθετη περίπτωση, ο web crawler σαρώνει την ιστοσελίδα, εξάγοντας τους υπερσυνδέσμους (hyperlinks) που αυτή διαθέτει, τους οποίους προσθέτει σε μια λίστα ειδική για αυτό το σκοπό, η οποία καλείται μέτωπο αναζήτησης (*crawl/search frontier*). Ο τρόπος εισαγωγής των υπερσυνδέσμων στο μέτωπο αναζήτησης καθορίζει και τον αλγόριθμο αναζήτησης που υλοποιεί ο crawler. Στην περίπτωση της παρούσας εργασίας, οι νέοι υπερσύνδεσμοι προστίθενται στο τέλος του μετώπου αναζήτησης, γεγονός που σημαίνει ότι ο crawler πρέπει να ολοκληρώσει την επεξεργασία όλων των σελίδων ενός επιπέδου πριν προχωρήσει στο επόμενο επίπεδο βάθους, υλοποιώντας έτσι τον αλγόριθμο αναζήτησης κατά πλάτος (*breadth first search – BFS*). Όταν ο crawler ολοκληρώσει την επεξεργασία μιας σελίδας, αυτή πλέον εισάγεται στο κλειστό σύνολο (*closed set*), έτσι ώστε να αποφεύγεται η επανεπεξεργασία της.

Η παραπάνω διαδικασία ακολουθείται αναδρομικά για τους υπερσυνδέσμους που βρίσκονται στο μέτωπο αναζήτησης σύμφωνα με τις πολιτικές που έχουν οριστεί, ανατροφοδοτώντας έτσι τον crawler με τους επόμενους προορισμούς του. Οι πολιτικές αυτές, όπως περιγράφηκαν στην υποενότητα 1.2.3, καθορίζουν την συμπεριφορά και την απόδοση ενός web crawler και αντιστοιχούν σε επιπλέον διεργασίες που παρεμβάλλονται στην παραπάνω διαδικασία. Συγκεκριμένα, ο web crawler είναι προτιμότερο να λειτουργεί με παράλληλο τρόπο ώστε να είναι ταχύς, σεβόμενος όμως τους κανόνες που έχουν θεσμοθετηθεί από το ειδικό πρωτόκολλο *robots exclusion protocol* ώστε να μην προκαλεί προβλήματα στις σελίδες που επισκέπτεται. Ταυτόχρονα χρειάζονται κάποιες τεχνικές που θα του επιτρέπουν να λειτουργεί επιλεκτικά βελτιώνοντας την απόδοσή του. Η επιλεκτικότητα αυτή μεταξύ άλλων αφορά την αποφυγή ιστοσελίδων παγίδων, την αποφυγή σελίδων που έχει επισκεφθεί πρόσφατα και άλλων που πιθανότατα δεν παρέχουν περιγραφές διαδικτυακών υπηρεσιών.

5.1.2. Πρόσθετα Χαρακτηριστικά

Κατά την ανάπτυξη του web crawler αντιμετωπίστηκαν δυσκολίες που προέρχονταν κυρίως από δύο παράγοντες:

- τους περιορισμένους πόρους που διέθετε το μηχάνημα στο οποίο εκτελούνταν ο web crawler (μνήμη, δίσκος, σύνδεση στο διαδίκτυο)

- την ετερογενή φύση του διαδικτύου, με πολλά διαφορετικά πρωτόκολλα, πρότυπα και τύπους περιεχομένου.

Στην παρούσα ενότητα περιγράφονται τροποποιήσεις, επεκτάσεις, προσθήκες και βελτιστοποιήσεις οι οποίες επήλθαν στον web crawler σταδιακά, με σκοπό να αντιμετωπιστούν τα θέματα που προέκυψαν κατά την δοκιμαστική λειτουργία του.

Περισσότερα στοιχεία για τη λειτουργία του web crawler και την απόδοσή του μετά τις αλλαγές που περιγράφονται στην παρούσα ενότητα αναφέρονται στο Κεφάλαιο 6.

5.1.2.1. Προσδιορισμός περιγραφών διαδικτυακών υπηρεσιών

Ο τρόπος με τον οποίο προσδιορίζεται η καταλληλότητα μιας ιστοσελίδας προς αποθήκευση, δηλαδή η ύπαρξη περιγραφής διαδικτυακής υπηρεσίας σε αυτήν, αποτέλεσε σημαντικό ζήτημα κατά τον σχεδιασμό του web crawler. Χωρίς την χρήση μιας αποτελεσματικής μεθόδου για τον σκοπό αυτό, ο στόχος του, δηλαδή η συλλογή περιγραφών διαδικτυακών υπηρεσιών από τον Παγκόσμιο ιστό, δεν θα ήταν εφικτός. Έτσι, μελετήθηκαν διάφορες τεχνικές οι οποίες θα μπορούσαν να χρησιμοποιηθούν για τον προσδιορισμό της καταλληλότητας μιας ιστοσελίδας.

Η πρώτη και πιο απλή τεχνική αφορά τον έλεγχο της κατάληξης της επισκεπτόμενης ιστοσελίδας, τον τύπο δηλαδή του αρχείου που φιλοξενείται σε μια διεύθυνση. Οι περιγραφές των διαδικτυακών υπηρεσιών που αναζητώνται έχουν καθορισμένες καταλήξεις, οι οποίες είναι “.wsdl” για τις περιγραφές στο πρότυπο WSDL και “.owls” για τις περιγραφές στο πρότυπο OWL-S.

Επιπρόσθετα, ο τύπος του περιεχομένου της ιστοσελίδας είναι δυνατό να προσδιοριστεί από την κεφαλίδα (header) της απόκρισης (HTTP response) που επιστρέφει η σελίδα στην αίτηση (HTTP request) που της αποστέλει ο crawler, και συγκεκριμένα από το πεδίο της απόκρισης “content-type”. Το συγκεκριμένο πεδίο περιέχει το MIME (Multipurpose Internet Mail Extensions) [39] type του περιεχομένου που επιστρέφεται, το οποίο είναι ένα αναγνωριστικό ειδικά για τον σκοπό αυτό. Η μέθοδος αυτή, αν και στη γενική περίπτωση είναι ορθότερη, δεν εξασφαλίζει την εγκυρότητα των αποτελεσμάτων, αφού το συγκεκριμένο πεδίο στην κεφαλίδα δεν είναι υποχρεωτικό. Επιπλέον, διαπιστώθηκε πειραματικά ότι στην περίπτωση των περιγραφών διαδικτυακών υπηρεσιών η τιμή του συγκεκριμένου πεδίου για τα αρχεία .wsdl πολλές φορές είναι “text/xml”, που αποτελεί ευρεία κατηγοριοποίηση και δεν διευκρινίζει αν η συγκεκριμένη σελίδα αντιστοιχεί σε περιγραφή διαδικτυακής υπηρεσίας.

Η τρίτη μέθοδος που θα μπορούσε να χρησιμοποιηθεί περιλαμβάνει τον έλεγχο της κεφαλίδας του κώδικα (source) της ιστοσελίδας που ο web crawler έχει επισκεφθεί, όπου μεταξύ άλλων προσδιορίζεται ο τύπος της. Η συγκεκριμένη μέθοδος, αν και θα μπορούσε να οδηγήσει σε ορθά αποτελέσματα, προσθέτει επιπλέον πολυπλοκότητα σε μια από τις πλέον συχνές διαδικασίες που πραγματοποιεί ο web crawler, εισάγοντας σημαντική καθυστέρηση στην όλη διαδικασία, και καθιστώντας το όφελος από τη χρήση της αμελητέο.

Η μέθοδος που τελικά επιλέχθηκε και χρησιμοποιείται είναι η πρώτη, αφού είναι ακριβής και δεν εμφανίζει τα μειονεκτήματα που αναφέρθηκαν στις υπόλοιπες δύο.

Κατά τις δοκιμές που πραγματοποιήθηκαν, διαπιστώθηκε πως όταν η παραγωγή του web service πραγματοποιείται με αυτόματο τρόπο μέσω ειδικών για τον σκοπό αυτό εργαλείων ολοκληρωμένων προγραμμάτων ανάπτυξης, όπως είναι για παράδειγμα το Ant του Netbeans, και η υπηρεσία έχει γίνει μέσω αυτών deployed σε κάποιον application server, η περιγραφή στο πρότυπο WSDL δημιουργείται και αυτή με αυτοματοποιημένο τρόπο χωρίς την παρέμβαση του χρήστη. Σε αυτή τη περίπτωση, η περιγραφή της υπηρεσίας είναι διαθέσιμη μέσω του application server σε μια διεύθυνση που είναι ίδια ακριβώς με την διεύθυνση που τρέχει το web service με την προσθήκη όμως της κατάληξης “?wsdl”. Η συγκεκριμένη ιδιορρυθμία, δεν προκαλεί προβλήματα στην τεχνική που επιλέχθηκε, αφού ο application server συμπεριφέρεται σαν web server στα HTTP αιτήματα που στέλνει ο web crawler, επιστρέφοντας το αρχείο .wsdl, μην απαιτώντας έτσι κάποια επιπλέον αλλαγή πέρα από την προσθήκη της κατάληξης “?wsdl” σε αυτές που αναζητούνται.

5.1.2.2. Αποφυγή αρχείων

Με ανάλογο τρόπο με αυτόν που περιγράφηκε στην υποενότητα 5.1.2.1 παραπάνω για τον προσδιορισμό των περιγραφών διαδικτυακών υπηρεσιών υλοποιήθηκε μια εκ διαμέτρου αντίθετη διεργασία.

Κατά την διάσχιση του παγκόσμιου ιστού ένας web crawler καλείται να αντιμετωπίσει πλήθος διαφορετικών προβλημάτων. Οι ιστοσελίδες που διασχίζει φιλοξενούν κάθε τύπου αρχεία, τα οποία ορισμένες φορές μπορούν να προκαλέσουν προβλήματα στη λειτουργία του, ενώ άλλες φορές δεν προσφέρουν κάτι στην αναζήτηση. Διαπιστώθηκε πως ειδικά τα αρχεία μεγάλου μεγέθους που συναντάει ο web crawler, εξαιτίας αυτής της ιδιομορφίας τους, καθυστερούν την πρόοδό του, μειώνοντας έτσι την ταχύτητα και την αποδοτικότητα του. Επιπλέον, τα αρχεία αυτά δεν περιέχουν υπερσυνδέσμους οι οποίοι θα μπορούσαν να χρησιμοποιηθούν για την

προώθηση της αναζήτησης. Για τους λόγους αυτούς κρίθηκε αναγκαία η χρήση της παραπάνω μεθόδου, ώστε να αποφεύγονται συγκεκριμένοι τύποι αρχείων που συνήθως έχουν σημαντικό μέγεθος, όπως τα αρχεία ήχου και βίντεο, ή/και δεν διαθέτουν υπερσυνδέσμους, όπως τα αρχεία εικόνων. Ακολουθεί στον Πίνακα 1 η λίστα των τύπων εκείνων που αποφεύγονται με αυτόν τον τρόπο.

Πίνακας 1. Καταλήξεις προς αποφυγή αρχείων.

Κατάληξη	Τύπος αρχείου	Κατάληξη	Τύπος αρχείου
.wmv	βίντεο	.flac	ήχου
.mov	βίντεο	.ppt	παρουσίασης
.mpeg	βίντεο	.pptx	παρουσίασης
.mpg	βίντεο	.doc	κειμένου
.avi	βίντεο	.docx	κειμένου
.wma	ήχου	.pdf	κειμένου
.msi	εκτελέσιμο	.zip	συμπίεσης
.flv	βίντεο	.rar	συμπίεσης
.run	εκτελέσιμο	.jar	εκτελέσιμο/βιβλιοθήκη java
.dmg	εικόνας	.tar	συμπίεσης
.pkg	πακέτο εγκατάστασης	.tar.gz	συμπίεσης
.tgz	συμπίεσης	.7z	συμπίεσης
.gz	συμπίεσης	.png	εικόνας
.bz2	συμπίεσης	.bmp	εικόνας
.ogv	βίντεο	.gif	εικόνας
.csv	κειμένου	.jpg	εικόνας
.aac	ήχου	.jpeg	εικόνας
.3gp	ήχου/βίντεο	.swf	flash
.m4v	βίντεο	.iso	εικονικού δίσκου
.m4a	ήχου/βίντεο	.exe	εκτελέσιμο
.mp2	ήχου	.mp4	ήχου/βίντεο
.mp3	ήχου	.wav	ήχου

5.1.2.3. Πολυνηματική υλοποίηση

Για την βελτίωση της ταχύτητας του web crawler, η οποία διαδραματίζει καθοριστικό ρόλο στην απόδοσή του, το κλειδί ήταν η ταυτόχρονα εκτέλεση τμήματος του αλγορίθμου του. Η χρήση ενός μόνο μηχανήματος και η ευκολία με την οποία η συγκεκριμένη τεχνική μπορεί να χρησιμοποιηθεί στην γλώσσα υλοποίησης του συστήματος υπαγόρευαν μια πολυνηματική υλοποίηση.

Η ταυτόχρονη αυτή εκτέλεση τμήματος του αλγορίθμου έπρεπε να υλοποιηθεί με τέτοιο τρόπο ώστε να διασφαλίζει την ορθότητα και την αυξημένη απόδοσή του. Γι' αυτό τον λόγο, έπρεπε να αποτραπεί η ταυτόχρονη πρόσβαση των νημάτων στις δομές που χρησιμοποιεί ο αλγόριθμος.

Έτσι διασφαλίζεται για παράδειγμα, ότι δεν θα μπορούσε το ίδιο URL να ανατεθεί σε δύο νήματα, πράγμα που αν συνέβαινε συχνά θα καθυστέρουσε σημαντικά την εκτέλεση του προγράμματος.

Για να γίνει κατανοητός ο τρόπος που επιλέχθηκε να πραγματοποιηθεί η πολυνηματική υλοποίηση, ο αλγόριθμος του web crawler χρειάζεται να εξετασθεί σε μεγαλύτερο βάθος. Ο αλγόριθμος ουσιαστικά αποτελείται από δύο τμήματα:

- την αλληλεπίδραση με το μέτωπο αναζήτησης και το κλειστό σύνολο με σκοπό την επιλογή της επόμενης προς επίσκεψη διεύθυνσης
- το διάβασμα του περιεχομένου της προς επίσκεψη διεύθυνσης και την εξαγωγή των συνδέσμων της ή την αποθήκευσή της

Με γνώμονα τα δύο αυτά τμήματα του αλγορίθμου, παρέχονται δύο εναλλακτικοί τρόποι.

Ο πρώτος τρόπος αφορά την πολυνηματική υλοποίηση ολόκληρου του αλγορίθμου. Κάτι τέτοιο είναι εφικτό στην Java, καθώς παρέχονται δυνατότητες συγχρονισμού (δομή synchronized) ώστε οι δομές που ενδέχεται να προκαλέσουν επιπλοκές να καταστούν εικονικά ανεξάρτητες. Ωστόσο, ο συγχρονισμός αυτός είναι δύσκολος, γιατί πρέπει να ληφθεί υπόψη κάθε περίπτωση στον αλγόριθμο όπου ενδέχεται να προκύψει επιπλοκή. Επιπλέον, το κόστος επικοινωνίας μεταξύ των νημάτων, καθώς και οι περίοδοι αναμονής για την υλοποίηση του συγχρονισμού ήταν μεγάλα σε σχέση με το κέρδος που θα προέκυπτε.

Η δεύτερη επιλογή, η οποία είναι και αυτή που τελικά επιλέχθηκε, αφορά την πολυνηματική υλοποίηση μονάχα εκείνου του τμήματος του αλγορίθμου που μπορεί να είναι ανεξάρτητο από την φύση του. Το δεύτερο τμήμα του αλγορίθμου, δηλαδή το διάβασμα του περιεχομένου μιας σελίδας και η εξαγωγή των συνδέσμων της ή η αποθήκευσή της, είναι μια διεργασία ανεξάρτητη που θα μπορούσε να επιτελεί το κάθε νήμα ξεχωριστά, χωρίς να χρησιμοποιείται κάποια κοινή δομή, αποφεύγοντας έτσι τις επιπλοκές. Ταυτόχρονα, ας σημειωθεί πως ούτως ή άλλως η συγκεκριμένη διεργασία αποτελεί το πλέον χρονοβόρο κομμάτι του αλγορίθμου, οπότε το κέρδος είναι σημαντικό.

5.1.2.4. Διαχείριση και ελαχιστοποίηση χρήσης μνήμης

Ένα από τα προβλήματα που εμφανίζονταν συχνά κατά την δοκιμαστική λειτουργία του web crawler ήταν ο τερματισμός της λειτουργίας του εξαιτίας σφαλμάτων μνήμης (heap space) τα οποία σχετίζονται με την ανεπάρκεια της μνήμης που διατίθεται στο πρόγραμμα.

Η ανεπάρκεια της μνήμης αποτελεί συνιστώσα διαφορετικών παραγόντων που σχετίζονται με την συμπεριφορά του crawler, το μηχάνημα στο οποίο τρέχει και άλλους παράγοντες. Η αναγκαιότητα ο web crawler να γνωρίζει τις σελίδες που έχει ήδη επισκεφθεί (κλειστό σύνολο) αλλά και τις σελίδες που είναι προγραμματισμένο να επισκεφθεί στο μέλλον (μέτωπο αναζήτησης) καθιστά την διατήρηση των δύο αυτών συνεχώς αυξανόμενων δομών στην μνήμη απαραίτητη. Η ιδιομοφία αυτή απαιτεί από μόνη της σημαντικό μέρος της μνήμης που διατίθεται στο πρόγραμμα, περιορίζοντάς έτσι το μέγεθος που είναι διαθέσιμο για άλλους σκοπούς όπως η προσωρινή αποθήκευση σελίδων. Αφού λοιπόν οι δύο αυτές δομές εξαπλώθηκαν αρκετά παρατηρήθηκε σημαντική πτώση της απόδοσης του προγράμματος, γεγονός που ώθησε στην ανάθεση περισσότερης μνήμης σε αυτό από το αντίστοιχο ρυθμιστικό αρχείο του προγραμματιστικού περιβάλλοντος.

Η τελευταία ενέργεια παρείχε πίστωση χρόνου και σίγουρα θα βελτίωνε την απόδοση του web crawler σε περίπτωση που η συνεχής εξάπλωση των δύο δομών περιοριζόταν κάποια στιγμή. Το γεγονός όμως ότι η μνήμη του μηχανήματος όπου ο web crawler τρέχει έχει πεπερασμένο μέγεθος, το οποίο είναι μάλιστα περιορισμένο συγκριτικά με τα μηχανήματα που χρησιμοποιούνται σε αντίστοιχες εταιρικές προσπάθειες, οδήγησε στην αναζήτηση διαφορετικών τρόπων διαχείρισης του μετώπου αναζήτησης και του κλειστού συνόλου που εξασφαλίζουν καλύτερη διαχείριση της διαθέσιμης μνήμης.

5.1.2.5. Διαχωρισμός μετώπου αναζήτησης σε αρχεία

Όπως ήδη αναφέρθηκε, λόγω του πεπερασμένου μεγέθους της μνήμης του μηχανήματος υπήρχε η ανάγκη διαχείρισης του μετώπου αναζήτησης με τέτοιο τρόπο που θα εξασφάλιζε την βέλτιστη διαχείρισή της.

Η φύση όμως του μετώπου αναζήτησης το καταδικάζει να εξαπλώνεται συνεχώς, δυσχεραίνοντας τη διαχείριση της διαθέσιμης μνήμης. Αυτό συμβαίνει λόγω της δομής του Παγκόσμιου Ιστού που αποτελείται από διασυνδεδεμένες ιστοσελίδες. Επειδή λοιπόν κάθε ιστοσελίδα περιέχει πολλούς υπερσυνδέσμους – δείχνει σε πολλές άλλες, η προσθήκη νέων διευθύνσεων στο μέτωπο αναζήτησης πραγματοποιείται με ρυθμό ταχύτερο από αυτόν με τον

οποίο ο web crawler επισκέπτεται τις σελίδες και άρα τις αφαιρεί από αυτό, αυξάνοντας διαρκώς το μέγεθος του με εκθετικό ρυθμό.

Για να αντιμετωπιστεί αυτό το ζήτημα είναι αναγκαίος ο περιορισμός του μεγέθους του μετώπου αναζήτησης χωρίς όμως αυτό να έχει αρνητικές επιπτώσεις στην αναζήτηση. Έτσι, αποφασίστηκε να χρησιμοποιηθεί ο σκληρός δίσκος σαν ενδιάμεσο αποθηκευτικό μέσο, χρησιμοποιώντας προσωρινά αρχεία. Η χρήση αρχείων θα εξασφάλιζε τη χρήση περιορισμένου ποσοστού της διαθέσιμης μνήμης από το μέτωπο αναζήτησης χωρίς να επηρεάζει τη ροή της αναζήτησης, αφού η αλλαγή αυτή δεν θα επέφερε κάποια σημαντική διαφορά στον τρόπο λειτουργίας του. Αυτό συμβαίνει επειδή τα μόνα «σημεία ενδιαφέροντος» του μετώπου αναζήτησης είναι η αρχή του, από την οποία λαμβάνεται η επόμενη προς επίσκεψη σελίδα, και το τέλος του, στο οποίο αποθηκεύονται οι νέες σελίδες προς επίσκεψη που προκύπτουν από τους υπερσυνδέσμους.

Για να υλοποιηθεί η συγκεκριμένη λειτουργία κρίθηκε απαραίτητος ο διαχωρισμός του μετώπου αναζήτησης σε δύο τμήματα, την κεφαλή και την ουρά. Η κεφαλή αποτελεί το τμήμα από το οποίο ο web crawler λαμβάνει τους συνδέσμους που θα επισκεφθεί, ενώ στην ουρά προστίθενται οι σύνδεσμοι που εξάγονται από τις σελίδες που επισκέπτεται. Όταν η ουρά γεμίσει, το περιεχόμενο της γράφεται σε ένα αρχείο και σημειώνεται η σειρά του, ενώ όταν η κεφαλή αδειάσει, ξαναγεμίζει από το επόμενο στη σειρά αρχείο.

Η συγκεκριμένη βελτιστοποίηση επέτρεψε τη θεαματική αύξηση του μεγέθους του μετώπου αναζήτησης σε σχέση με την λειτουργία του στην κύρια μνήμη, αποδεσμεύοντας έτσι ένα σημαντικό μέρος της το οποίο είναι πλέον διαθέσιμο σε άλλες λειτουργίες του προγράμματος. Ταυτόχρονα, βρέθηκε το ιδανικό μέγεθος λιστών και αρχείων ώστε η επικοινωνία με τον δίσκο να μην εισάγει σημαντική καθυστέρηση στη λειτουργία του προγράμματος.

5.1.2.6. Συμπύεση

Η χρήση του σκληρού δίσκου σαν αποθηκευτικό μέσο των προσωρινών αρχείων του μετώπου αναζήτησης βοήθησε ώστε το μήκος του να επιμηκυνθεί σημαντικά χωρίς σημαντική επίπτωση στην αναζήτηση. Παρόλα αυτά, κατ' αντιστοιχία με την προηγούμενη κατάσταση, ακόμα και ο σκληρός δίσκος του υπολογιστή θα γέμιζε αφού το μέγεθος του είναι πεπερασμένο και ο ρυθμός με τον οποίο ο web crawler διασχίζει τις σελίδες δεν μπορεί να πλησιάσει τον ρυθμό με τον οποίο προστίθενται νέες στο μέτωπο αναζήτησης.

Μια τεχνική που χρησιμοποιήθηκε για την περαιτέρω επιμήκυνση του μετώπου αναζήτησης ήταν η συμπίεση (compression). Τα προσωρινά αρχεία του μετώπου αναζήτησης, όντας απλά αρχεία κειμένου με διευθύνσεις ήταν δυνατό να συμπιεσθούν σημαντικά, εξοικονομώντας έτσι επιπλέον χώρο. Για να γίνει κατανοητό το μέγεθος της εξοικονόμησης αναφέρεται ενδεικτικά πως μια συστάδα χιλίων αρχείων έχουν μέγεθος που αγγίζει τα 14 GB, ενώ μετά τη συμπίεση το μέγεθός τους μειώνεται κατά περίπου 94%, και περιορίζεται στα 900 MB.

Η συγκεκριμένη μέθοδος είναι δυνατό να αυτοματοποιηθεί με την ενσωμάτωσή της στον αλγόριθμο του web crawler.

5.1.2.7. Περιορισμός κλειστού συνόλου

Ο περιορισμός του κλειστού συνόλου ήταν αναγκαίος ώστε να συνεχίσει την αποδοτική λειτουργία του ο crawler. Ωστόσο, σε αντίθεση με το μέτωπο αναζήτησης όπου η στροφή στην χρήση του δίσκου δεν επέφερε ουσιαστικές αλλαγές στον τρόπο με τον οποίο λειτουργούσε, η συγκεκριμένη αλλαγή είχε ουσιαστικό αντίκτυπο στην λειτουργία του κλειστού συνόλου.

Σε αυτό το σημείο υπενθυμίζεται πως ο ρόλος του κλειστού συνόλου είναι να διατηρεί τις σελίδες που ο web crawler έχει επισκεφθεί έτσι ώστε να αποφεύγονται μελλοντικές ενδεχόμενες επανεπισκέψεις. Πλέον, μετά τον περιορισμό του μεγέθους του κλειστού συνόλου, στην μνήμη κρατείται μέρος των σελίδων που ο crawler έχει επισκεφθεί, που αντιστοιχούν στις πιο πρόσφατες, επιτρέποντας την αποφυγή μόνο των συγκεκριμένων σελίδων που έχει επισκεφθεί ο web crawler.

Τελικά, παρόλο που εξαιτίας της αλλαγής αυτής υπάρχει το ενδεχόμενο επανεπίσκεψης ιστοσελίδων που έχουν προσπελασθεί στο παρελθόν, το γεγονός αυτό δεν θεωρείται ότι επηρεάζει αρνητικά την αναζήτηση. Ο βασικός σκοπός του κλειστού συνόλου συνεχίζει να επιτελείται, αφού ο web crawler αποτρέπεται από τοπικούς ατέρμονες βρόγχους, ενώ η επίσκεψη ήδη προσπελασμένων ιστοσελίδων μετά από καιρό δεν αποτελεί μείζον θέμα, αφού ενδέχεται το περιεχόμενό τους να έχει αλλάξει.

5.1.2.8. Αποθήκευση κατάστασης

Καθώς ο web crawler πλοηγείται στο Διαδίκτυο, είναι ανάγκη να είναι ανθεκτικός στα περισσότερα προβλήματα που μπορούν να εμφανιστούν και να επηρεάσουν την λειτουργία του. Με αυτό το σκεπτικό προέκυψε η ανάγκη να πραγματοποιείται συχνά αποθήκευση της κατάστασής του σε δεδομένες χρονικές στιγμές με την μορφή αντιγράφων ασφαλείας (backups),

που αφορούν τόσο το μέτωπο αναζήτησης όσο και το κλειστό σύνολο, ώστε η πραγματοποιημένη πρόοδος να μην χάνεται σε κάθε απρόβλεπτο ενδεχόμενο. Η συγκεκριμένη δυνατότητα προστέθηκε, με την λίστα αναζήτησης και το κλειστό σύνολο να αποθηκεύονται σε αρχεία στον δίσκο περιοδικά, σε συνδυασμό με την δυνατότητα επανεκκίνησης του crawler από ένα συγκεκριμένο σημείο με χρήση των αντιγράφων ασφαλείας.

5.1.2.9. Ευριστικός κανόνας για τα domain και λίστα αποφυγής

Για την βελτίωση της απόδοσης του web crawler, πέρα από την ταχύτητα, σημαντικό ρόλο διαδραματίζει και ο τρόπος με τον οποίο επιλέγει τις σελίδες που επισκέπτεται ή αποφεύγει. Ο συγκεκριμένος ευριστικός κανόνας αφορά τα κριτήρια αποφυγής ενός domain. Το μέτρα αυτό εξασφαλίζει την αποφυγή σελίδων που είναι ειδικά σχεδιασμένες ώστε να εγκλωβίζουν τον web crawler σε ατέρμονες βρόχους εντός αυτών και την αποφυγή domains που δεν περιέχουν περιγραφές υπηρεσιών.

Ο κανόνας ορίζει πως αν ο web crawler επισκεφθεί έναν μεγάλο αριθμό συνεχόμενων σελίδων από το ίδιο domain χωρίς να συναντήσει κάποια περιγραφή διαδικτυακής υπηρεσίας ή το ποσοστό των περιγραφών διαδικτυακών υπηρεσιών στο σύνολο των σελίδων από το συγκεκριμένο domain που έχουν ήδη εξεταστεί είναι αμελητέο, τότε όλες οι υπόλοιπες σελίδες από το συγκεκριμένο domain μπορούν να αγνοηθούν, δηλαδή να αποφευχθεί η επίσκεψή τους, με σχετική βεβαιότητα ότι δεν θα παρήγαγαν χρήσιμα αποτελέσματα. Για να πραγματοποιηθεί αυτό η σελίδα προστίθεται σε μια λίστα με domains προς αποφυγή (ignore list). Η λίστα αυτή χρησιμοποιείται στη συνέχεια για φιλτράρισμα του μετώπου αναζήτησης.

5.1.2.10. Ευριστικός κανόνας για εύρεση περιγραφών WSDL από περιγραφές OWL-S

Έχει παρατηρηθεί ότι στους διακομιστές όπου στεγάζονται περιγραφές υπηρεσιών στο πρότυπο OWL-S, συνήθως βρίσκονται και αντίστοιχες περιγραφές στο πρότυπο WSDL σε παρόμοια διεύθυνση, που αντιστοιχούν στο grounding της OWL-S υπηρεσίας. Για αυτό τον λόγο, κάθε φορά που συναντάται μια υπηρεσία OWL-S, προστίθεται στο μέτωπο αναζήτησης και ένα κατασκευασμένο URL στο οποίο ίσως υπάρχει περιγραφή WSDL.

5.1.2.11. Τοπική αλλαγή αλγορίθμου αναζήτησης (Breadth-to-depth heuristic)

Ένας ακόμη ευριστικός κανόνας που εφαρμόζεται υπαγορεύει την τοπική μετατροπή του αλγορίθμου αναζήτησης από κατά πλάτος (breadth first search) σε κατά βάθος (depth first search) όταν συναντηθεί μια περιγραφή διαδικτυακής υπηρεσίας, με τη λογική ότι στον ίδιο

ιστοχώρο μπορεί να βρίσκονται κι άλλες περιγραφές διαδικτυακών υπηρεσιών. Ο κανόνας ορίζει πως αν εντοπιστεί μια διαδικτυακή υπηρεσία σε μια σελίδα, τότε παρακάμπτεται η σειρά του μετώπου αναζήτησης και αμέσως μετά ακολουθούνται οι υπόλοιποι σύνδρομοι που βρίσκονται σε αυτή τη σελίδα.

Η συγκεκριμένη μέθοδος εξασφαλίζει πως τυχόν περιγραφές διαδικτυακών υπηρεσιών που βρίσκονται στον ίδιο ιστοχώρο με αυτή που έχει ήδη βρεθεί θα συναντηθούν νωρίτερα κατά την αναζήτηση, αφού είναι σύνηθες να βρίσκονται κοντά, και πως δεν θα χαθούν εξαιτίας του περιορισμένου μήκους του μετώπου αναζήτησης.

5.1.2.12. Προγραμματιστικές Βελτιστοποιήσεις

Για την αύξηση της απόδοσης του crawler πραγματοποιήθηκαν επιπλέον προγραμματιστικές βελτιστοποιήσεις, οι τρεις σημαντικότερες εκ των οποίων αναφέρονται παρακάτω.

Η πρώτη από αυτές ήταν η αλλαγή του τύπου δεδομένων που χρησιμοποιείται για την διαχείριση των διευθύνσεων από URL σε String. Αρχικά για την αναπαράσταση των URLs είχε χρησιμοποιηθεί η έτοιμη ομότιτλη κλάση της Java για τον σκοπό αυτό. Αυτό έγινε γιατί έτσι θα μπορούσαν να χρησιμοποιηθούν οι έτοιμες βοηθητικές συναρτήσεις της κλάσης όπως για παράδειγμα εκείνες που δίνουν την δυνατότητα απομόνωσης συγκεκριμένων τμημάτων ενός URL. Η αλλαγή σε String αύξησε σημαντικά την ταχύτητα επεξεργασίας, ενώ σε πολύ ειδικές συνθήκες, η διεύθυνση μετατρέπεται από String σε URL προσωρινά.

Η αλλαγή του μεγέθους του buffer σύμφωνα με το οποίο ο crawler διαβάζει το περιεχόμενο της ιστοσελίδας επέφερε επίσης σημαντική βελτίωση στην απόδοση του crawler. Για την βελτιστοποίηση της συγκεκριμένης διαδικασίας χρησιμοποιείται ένα πεδίο αντίστοιχο με το “content-type” που βρίσκεται στην κεφαλίδα της http απόκρισης της ιστοσελίδας και ονομάζεται “content-length”. Το πεδίο αυτό προσδιορίζει το μήκος της σελίδας δίνοντας έτσι την δυνατότητα να χρησιμοποιηθεί για τον προσδιορισμό του μεγέθους του buffer ώστε να ελαχιστοποιούνται τα ενδιάμεσα διαβάσματα. Ορίζοντας το μέγεθος του buffer ίσο με το μέγεθος του περιεχομένου της ιστοσελίδας είναι δυνατό το άμεσο διάβασμα ολόκληρου του περιεχομένου.

Τέλος, βελτίωση παρατηρήθηκε και από την μείωση των συντμήσεων των συμβολοσειρών (String concatenations) τα οποία πραγματοποιούνται. Οι συμβολοσειρές στην Java έχουν την ιδιομορφία ότι είναι *αμετάβλητες* (immutable), το περιεχόμενο τους δηλαδή δεν μπορεί να τροποποιηθεί αφού δημιουργηθούν. Έτσι όταν στον κώδικα υπάρχει η ανάγκη να

πραγματοποιηθεί κάποια σύντμηση, απαιτείται να δημιουργηθεί ένα αντίγραφο της συμβολοσειράς, καταλαμβάνοντας έτσι πλεονάζοντα χώρο. Εφόσον αυτή η διαδικασία πραγματοποιείται συχνά κατά τη λειτουργία του crawler, για μεγάλου μήκους συμβολοσειρές εμφανίζονταν προβλήματα απόδοσης και μνήμης [28].

5.1.2.13. Ρυθμίσεις

Όπως αναφέρθηκε στην ενότητα 4, σκοπός του σχεδιασμού ήταν κάθε υποσύστημα της μηχανής αναζήτησης που κατασκευάστηκε να είναι δυνατό να παραμετροποιήσει την λειτουργία του σύμφωνα με τις απαιτήσεις του εκάστοτε χρήστη κατά τρόπο αντίστοιχο με προγράμματα της αγοράς, χωρίς να απαιτείται απευθείας παρέμβαση στον κώδικα. Αυτό συνήθως πραγματοποιείται χρησιμοποιώντας αρχεία ρυθμίσεων στα οποία ο χρήστης έχοντας επίγνωση των παραμέτρων μπορεί να τις αλλάξει και έτσι να προσαρμόσει το σύστημα στις ανάγκες του. Η μορφή του συγκεκριμένου αρχείου για τον web crawler που κατασκευάζεται φαίνεται στην Εικόνα 16 παρακάτω.

```
The following fields enable the user to configure the crawler in
order to meet his/her demands.

Select seeds source ( 0 for seeds.conf, 1 for search list and
closed set backups - SLT files )
source:[0]

If the search list backups have been selected, it is obligatory
that you must fill the following two fields. The "SLT to read"
field corresponds to the backup file from which the seed URLs
will be selected, while the "SLT to write" field corresponds to
the name of the next backup file that will be written.

SLT to read:[1223]
SLT to write:[1224]

Similarly, you can specify the Closed Set backup files. The "CS
to read" field enables the user to select the starting backup
file, while the "CS to write" specifies the name of the next
closed set backup file to be written.

CS to read:[29]
CS to write:[30]

Select the number of unique URLs checked between closed set
backups. ( up to 5 digits )
unique URLs:[20000]

You can specify the size of the search tail and the closed set
backup by filling the fields below. The actual size of the
closed set backup in memory will be double the amount you
select.

slt size:[200000]
csb size:[500000]

The number of threads running can be specified below.
( Restricted by the system's hardware )
number of threads:[5]
```

Εικόνα 16. Το αρχείο τροποποίησης των ρυθμίσεων “configuration.cfg”.

Όπως διακρίνεται, με τη χρήση του συγκεκριμένου αρχείου είναι δυνατή η τροποποίηση των παραμέτρων λειτουργίας του web crawler. Για την διευκόλυνση του χρήστη, τις παραμέτρους

συνοδεύουν σύντομες επεξηγηματικές σημειώσεις για τον τρόπο που κάθε παράμετρος επηρεάζει την λειτουργία του συστήματος αλλά και για το εύρος τιμών που θα είχε νόημα να λάβει η κάθε μια, όπου αυτό κρίνεται απαραίτητο. Οι παράμετροι οι οποίες μπορούν να οριστούν μέσω του συγκεκριμένου αρχείου φαίνονται στον Πίνακα 2.

Πίνακας 2. Παράμετροι αρχείου τροποποίησης ρυθμίσεων.

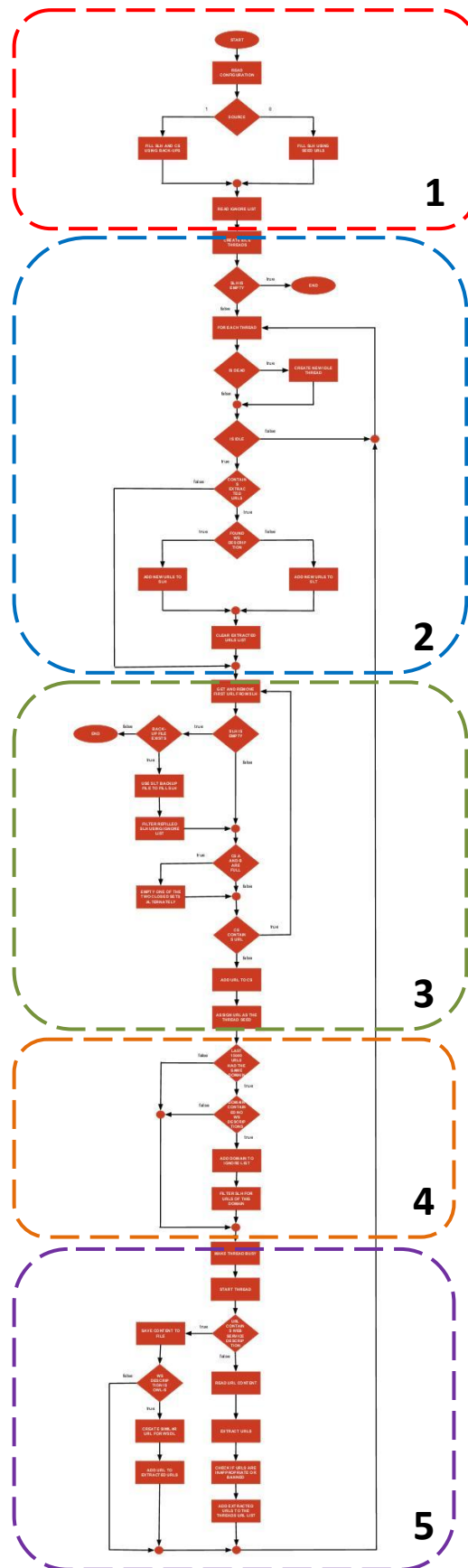
Source	Προσδιορίζει την πηγή των διευθύνσεων που θα επισκεφθεί ο web crawler μόλις εκκινήσει. Μπορεί να είναι είτε το αρχείο με τις διευθύνσεις σπόρους (seeds) είτε αποθηκευμένα αρχεία από προηγούμενα τρεξίματα.
SLT (Search List Tail) to read	Ο αριθμός του επόμενου προς ανάγνωση αρχείου επέκτασης του μετώπου αναζήτησης.
SLT to write	Ο αριθμός του επόμενου προς εγγραφή αρχείου επέκτασης του μετώπου αναζήτησης.
CS (Closed Set) to read	Ο αριθμός του επόμενου προς ανάγνωση αρχείου κλειστού συνόλου.
CS to write	Ο αριθμός του επόμενου προς ανάγνωση αρχείου κλειστού συνόλου.
Unique URLs	Ο αριθμός των διευθύνσεων ανάμεσα στις αποθηκεύσεις του κλειστού συνόλου.
SLT size	Ο αριθμός των διευθύνσεων που αποθηκεύονται σε κάθε αρχείου κλειστού συνόλου.
CS size	Ο αριθμός των διευθύνσεων που αποθηκεύονται σε κάθε αρχείου κλειστού συνόλου.
Number of threads	Ο αριθμός των νημάτων που χρησιμοποιούνται για την λειτουργία του web crawler.

5.1.3. Υποστηριζόμενη Λειτουργία

Στην Εικόνα 17 παρουσιάζεται το πλήρες διάγραμμα λειτουργίας του αλγορίθμου του web crawler για περιγραφές διαδικτυακών υπηρεσιών που σχεδιάστηκε και υλοποιήθηκε στο πλαίσιο της παρούσας πτυχιακής εργασίας.

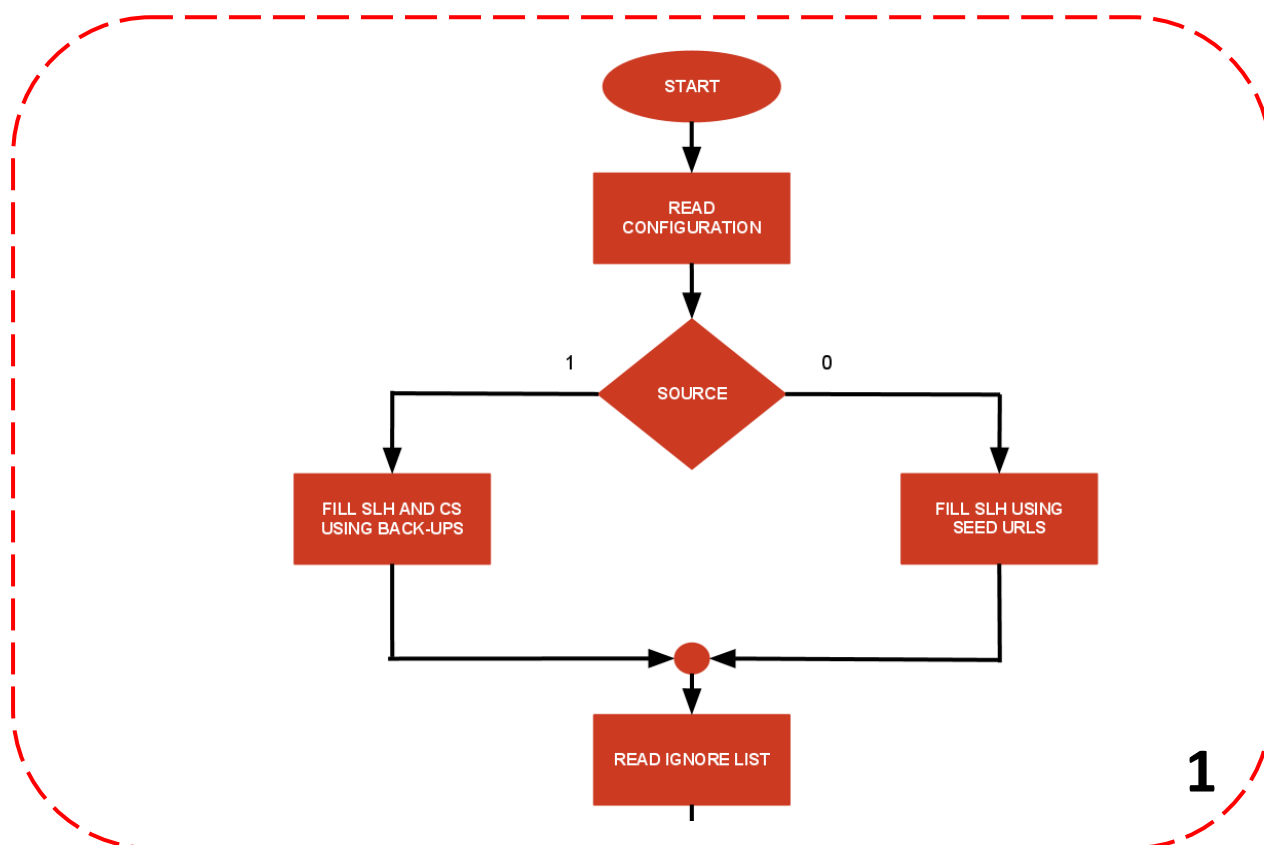
Λόγω του μεγέθους του αλγορίθμου το διάγραμμα είναι αρκετά μεγάλο καθιστώντας δύσκολη την διαδικασία παρουσίασής του. Έτσι, αρχικά παρουσιάζεται το πλήρες διάγραμμα διαχωρισμένο με διακεκομμένα πλαίσια σε υπομήματα και την γενική εικόνα ακολουθεί αναλυτική παρουσίαση, επεξήγηση και συσχετισμός του κάθε επιμέρους τμήματος του αλγορίθμου με τα επιπρόσθετα χαρακτηριστικά που προσδιορίστηκαν και αναπτύχθηκαν παραπάνω.

Σημειώνεται πως τα αρχικά CS (Closed Set), SLH (Search List Head) και SLT (Search List Tail) αντιστοιχούν στο κλειστό σύνολο, την κεφαλή και την ουρά του μετώπου αναζήτησης αντίστοιχα.



Εικόνα 17. Πλήρες διάγραμμα λειτουργίας του αλγορίθμου του web crawler.

Στην Εικόνα 18 φαίνονται οι ενέργειες που πραγματοποιεί ο web crawler αφού εκκινηθεί.



Εικόνα 18. Τμήμα 1 - Εκκίνηση του web crawler.

Πρώτο του μέλημα είναι η ανάγνωση του αρχείου παραμετροποίησης που τον συνοδεύει και ονομάζεται “configuration.cfg”. Οι παράμετροι που μπορούν να τροποποιηθούν μέσω του συγκεκριμένου αρχείου έχουν περιγραφεί στην Ενότητα 5.1.2.13. Η παράμετρος “source”, πρώτη στο αρχείο των ρυθμίσεων, συναντάται στην συνθήκη που ακολουθεί.

Όπως υποδηλώνει και το όνομα της παραμέτρου που ελέγχεται («πηγή») η πρώτη συνθήκη καθορίζει την πηγή από την οποία ο web crawler θα διαβάσει τις αρχικές διευθύνσεις-σπόρους της αναζήτησης. Η μεταβλητή “source” μπορεί να πάρει δύο τιμές, 1 και 0, οι οποίες οδηγούν τον crawler να διαβάσει τις διευθύνσεις από τα αντίστοιχα αρχεία. Αν η τιμή της μεταβλητής είναι 0, τότε διαβάζονται από ένα ειδικό αρχείο δημιουργημένο για αυτόν τον σκοπό, του οποίου το όνομα δεν θα μπορούσε να είναι άλλο παρά “seeds”. Αντίθετα, αν η τιμή της παραμέτρου είναι 1, όπως περιγράφεται και στο διάγραμμα, ο web crawler καλείται να διαβάσει τις διευθύνσεις από τα προσωρινά αποθηκευμένα αρχεία του μετώπου αναζήτησης και του κλειστού συνόλου.

Είναι κατανοητό πως η τιμή 0 είναι η κατάλληλη σε περίπτωση που ο crawler χρησιμοποιείται για πρώτη φορά, ενώ η τιμή 1 είναι η κατάλληλη επιλογή αν ο crawler έχει ήδη χρησιμοποιηθεί και σταματήσει για κάποιον λόγο, και το ζητούμενο είναι η επαναφορά του σε λειτουργία από κάποια προηγούμενη αποθηκευμένη κατάσταση, όπως περιγράφηκε στην Ενότητα 5.1.2.8.

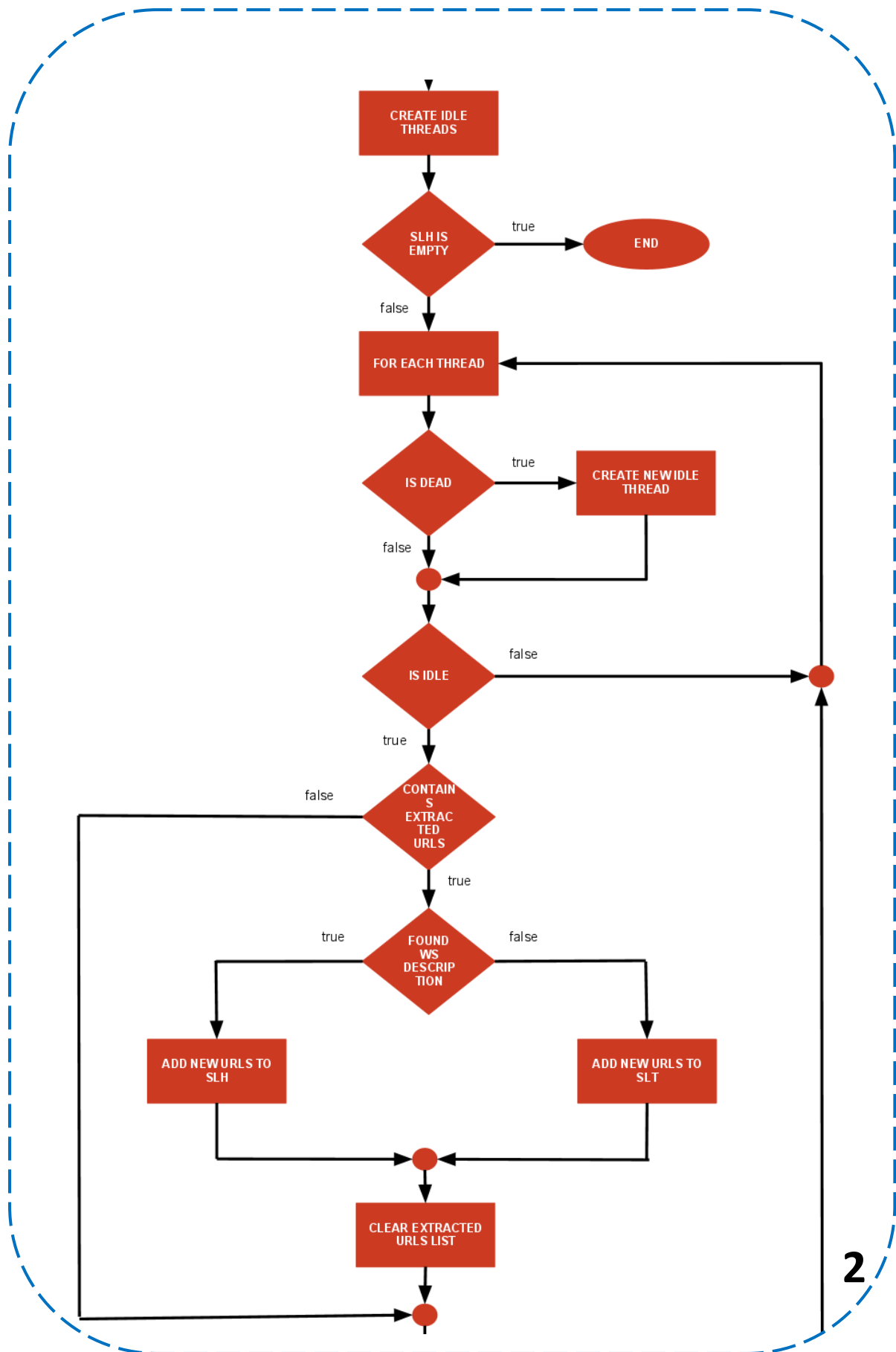
Σε κάθε περίπτωση, οι διευθύνσεις που διαβάζονται τοποθετούνται στο μέτωπο αναζήτησης στο οποίο προσδιορίζονται οι επόμενες σελίδες που θα επισκεφθεί ο web crawler.

Η επόμενη ενέργεια που σηματοδοτείται στην Εικόνα 18 αφορά την ανάγνωση ενός δευτέρου αρχείου που αντιστοιχεί στην ignore list, όπως αυτή περιγράφηκε στην Ενότητα 5.1.2.9. Στο συγκεκριμένο αρχείο βρίσκονται αποθηκευμένα domains τα οποία πρέπει να αποφευχθούν κατά την αναζήτηση, είτε επειδή δεν αναμένεται να περιέχουν περιγραφές διαδικτυακών υπηρεσιών είτε επειδή ίσως αποτελούν παγίδα ειδικά για web crawlers. Ο τρόπος με τον οποίο κρίνεται πως ένα domain εισάγεται σε αυτή τη λίστα αντιστοιχεί στον ευρετικό κανόνα για τα domains που περιγράφηκε στην Ενότητα 5.1.2.9, ενώ θα εξετασθεί ξανά στην συνέχεια του αλγορίθμου.

Αφού πραγματοποιήσει τις προηγούμενες ρυθμιστικές ενέργειες, ο web crawler δημιουργεί τα νήματα (threads) που χρησιμεύουν στην πολυνηματική υλοποίηση μέρους του αλγορίθμου. Η ενέργεια αυτή είναι η πρώτη που φαίνεται στην Εικόνα 19.

Εφόσον τα νήματα δημιουργηθούν η ροή προχωράει στο βασικό τμήμα του αλγορίθμου. Αρχικά, ελέγχει αν το μέτωπο αναζήτησης είναι κενό ώστε να αποφασισθεί αν θα συνεχίσει με τις επόμενες ενέργειες. Αν δεν υπάρχουν διευθύνσεις προς επίσκεψη είναι λογικό να σταματά η λειτουργία του web crawler, αφού αυτό σημαίνει ότι κάτι δεν λειτούργησε σωστά κατά την ανάγνωση των αρχείων που τις παρέχουν. Υπό φυσιολογικές συνθήκες το μέτωπο αναζήτησης διαθέτει τις διευθύνσεις σπόρους, οπότε ο αλγόριθμος προχωρά στα επόμενα βήματα που πραγματοποιούνται ξεχωριστά για κάθε νήμα.

Πριν αρχίσουν να επιτελούν κάποια εργασία, τα νήματα περνούν ένα στάδιο προετοιμασίας. Πρώτα το κάθε νήμα υπόκειται σε έναν έλεγχο για να διαπιστωθεί αν είναι ανενεργό, οπότε επαναδημιουργείται στην αρχική του κατάσταση. Ο συγκεκριμένος έλεγχος είναι απαραίτητος έτσι ώστε να επανεκκινούνται νήματα που έχουν σταματήσει να λειτουργούν. Το νήμα τότε εξετάζεται για να διαπιστωθεί αν επιτελεί κάποια εργασία ή είναι ελεύθερο. Είναι ευνόητο πως αν το νήμα είναι απασχολημένο αγνοείται και η διαδικασία προχωρά με το επόμενο στην σειρά. Διαφορετικά, προχωρούν τα επόμενα στάδια της προετοιμασίας ενός νήματος.



Εικόνα 19. Τμήμα 2 - Προετοιμασία του νήματος.

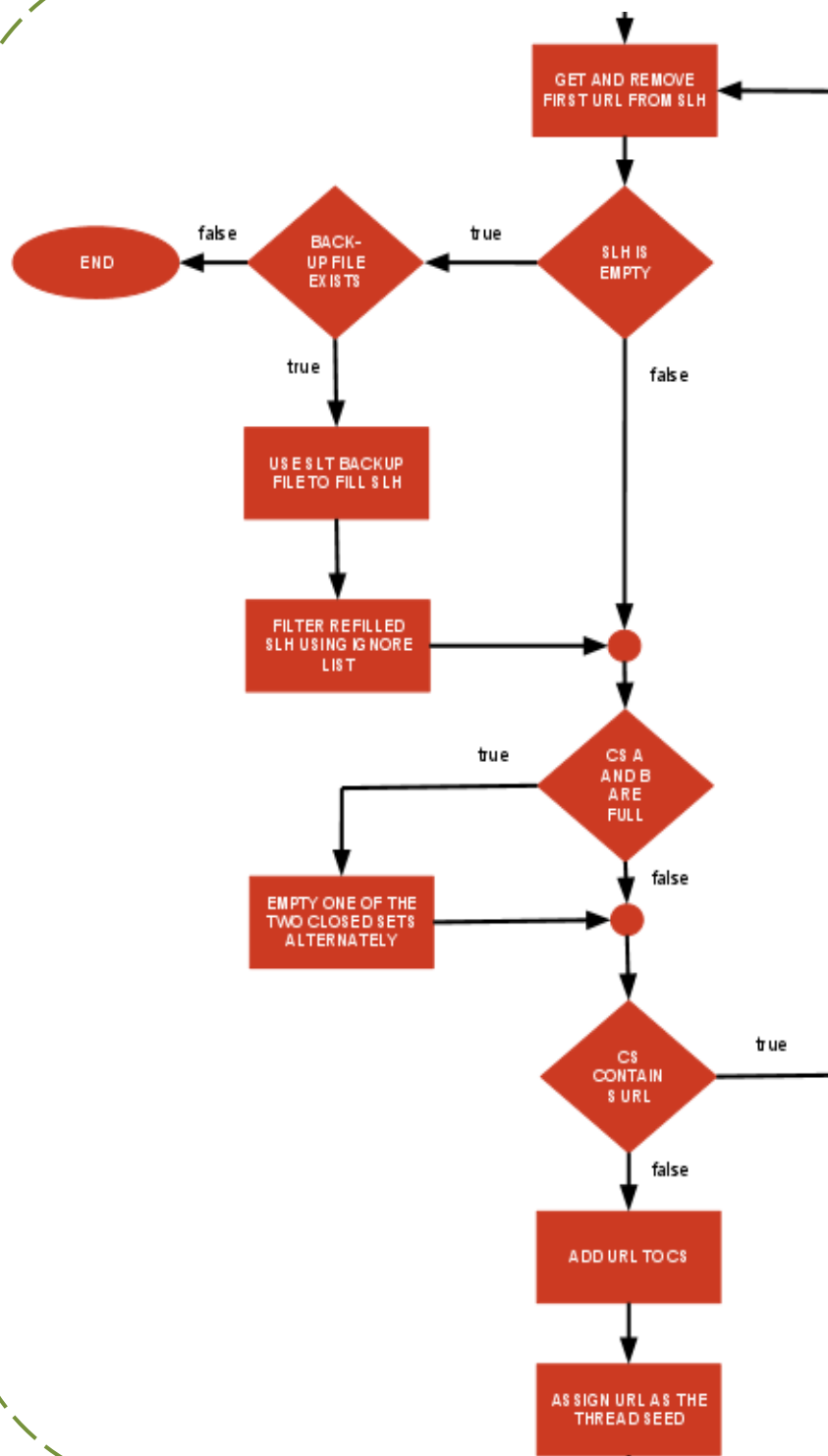
Το τρέχον νήμα υπόκειται σε έλεγχο για να διαπιστωθεί αν περιέχει διευθύνσεις που συνέλεξε από προηγούμενο τρέξιμο. Αν η συνθήκη αυτή αληθεύει, τότε οι διευθύνσεις αυτές αφαιρούνται από το νήμα και τοποθετούνται στο μέτωπο αναζήτησης. Μάλιστα, αν το νήμα κατά την προηγούμενη εργασία του εντόπισε τουλάχιστον μια περιγραφή διαδικτυακής υπηρεσίας, τότε εφαρμόζεται ο ευρετικός κανόνας που αλλάζει προσωρινά τον αλγόριθμο αναζήτησης από κατά πλάτος σε κατά βάθος, όπως περιγράφεται στην Ενότητα 5.1.2.11.

Στο τρίτο τμήμα του αλγορίθμου (Εικόνα 20), η πρώτη διεύθυνση του μετώπου αναζήτησης αφαιρείται από αυτό αλλά μεσολαβούν μια σειρά από άλλες διεργασίες πρώτου αυτό ανατεθεί στο έτοιμο πλέον νήμα και αυτό περάσει στην ουσιαστική εργασία του.

Αφού η πρώτη διεύθυνση αφαιρεθεί από το μέτωπο αναζήτησης, είναι απαραίτητο να πραγματοποιηθεί έλεγχος για να εξασφαλιστεί ότι το μέτωπο αναζήτησης μπορεί να υποστηρίξει τα επόμενα νήματα, δηλαδή ότι δεν έχει αδειάσει. Αν είναι κενό, τότε διαβάζεται κάποιο από τα προσωρινά αρχεία με διευθύνσεις που χρησιμοποιούνται για την επιμήκυνση του μετώπου πέρα από την μνήμη, στον δίσκο, όπως περιγράφηκε στην Ενότητα 5.1.2.5. Πλέον έχει επικυρωθεί ότι το μέτωπο αναζήτησης δεν είναι άδειο αλλά πρώτα πρέπει να φιλτραριστεί με βάση τις διευθύνσεις που υπάρχουν στην λίστα αποφυγής.

Την επεξεργασία του μετώπου αναζήτησης ακολουθούν ενέργειες που αφορούν το κλειστό σύνολο. Το κλειστό σύνολο αποτελείται από δύο ίσα τμήματα τα οποία διευκολύνουν τη διαχείριση του περιορισμένου μεγέθους του. Αν και τα δύο τμήματα είναι γεμάτα, τότε αδειάζει το ένα από τα δύο εναλλάξ ώστε να υποδεχθεί τις νέες σελίδες, ενώ στο άλλο παραμένουν οι πιο πρόσφατα εξετασμένες ιστοσελίδες.

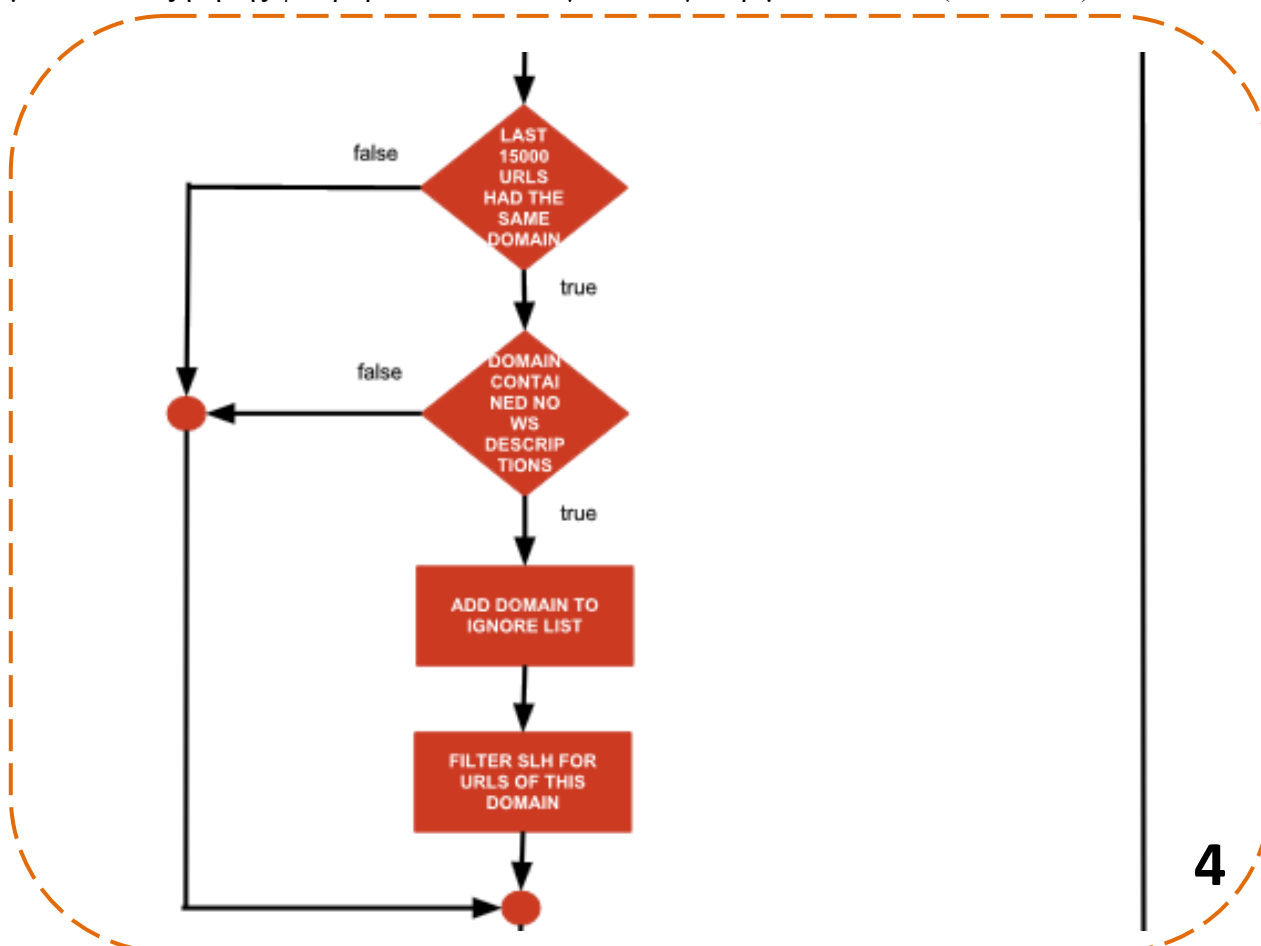
Αν η διεύθυνση που αφαιρέθηκε από το μέτωπο αναζήτησης υπάρχει στο διαμορφωμένο κλειστό σύνολο, τότε η ίδια διαδικασία επαναλαμβάνεται για την επόμενη διεύθυνση του μετώπου. Αν η διεύθυνση δεν προϋπάρχει στο κλειστό σύνολο τότε προστίθεται σε αυτό και τελικά ανατίθεται στο νήμα. Με αυτό τον τρόπο αποτρέπεται η επανεπίσκεψη σελίδων και ο εγκλωβισμός του crawler σε τοπικούς βρόγχους.



3

Εικόνα 20. Τμήμα 3 – Μηχανισμός διαχείρισης Μετώπου Αναζήτησης και Κλειστού Συνόλου και ανάθεση διεύθυνσης στο τρέχον νήμα.

Η τελευταία διεργασία που πραγματοποιείται πρώτου το νήμα ξεκινήσει την λειτουργία του αφορά τον ευρετικό κανόνα που ορίστηκε για τα domains και περιγράφηκε στην Ενότητα 5.1.2.9. Ο ευρετικός αυτός κανόνας ορίζει ότι αν το domain του συγκεκριμένου URL είναι ίδιο με τα domains των προηγούμενων 15000, στα οποία μάλιστα δεν έχει εντοπιστεί περιγραφή διαδικτυακής υπηρεσίας, τότε αυτό το domain προστίθεται στην λίστα αποφυγής και μάλιστα το μέτωπο αναζήτησης φιλτράρεται επιτόπου για το συγκεκριμένο domain (Εικόνα 21).

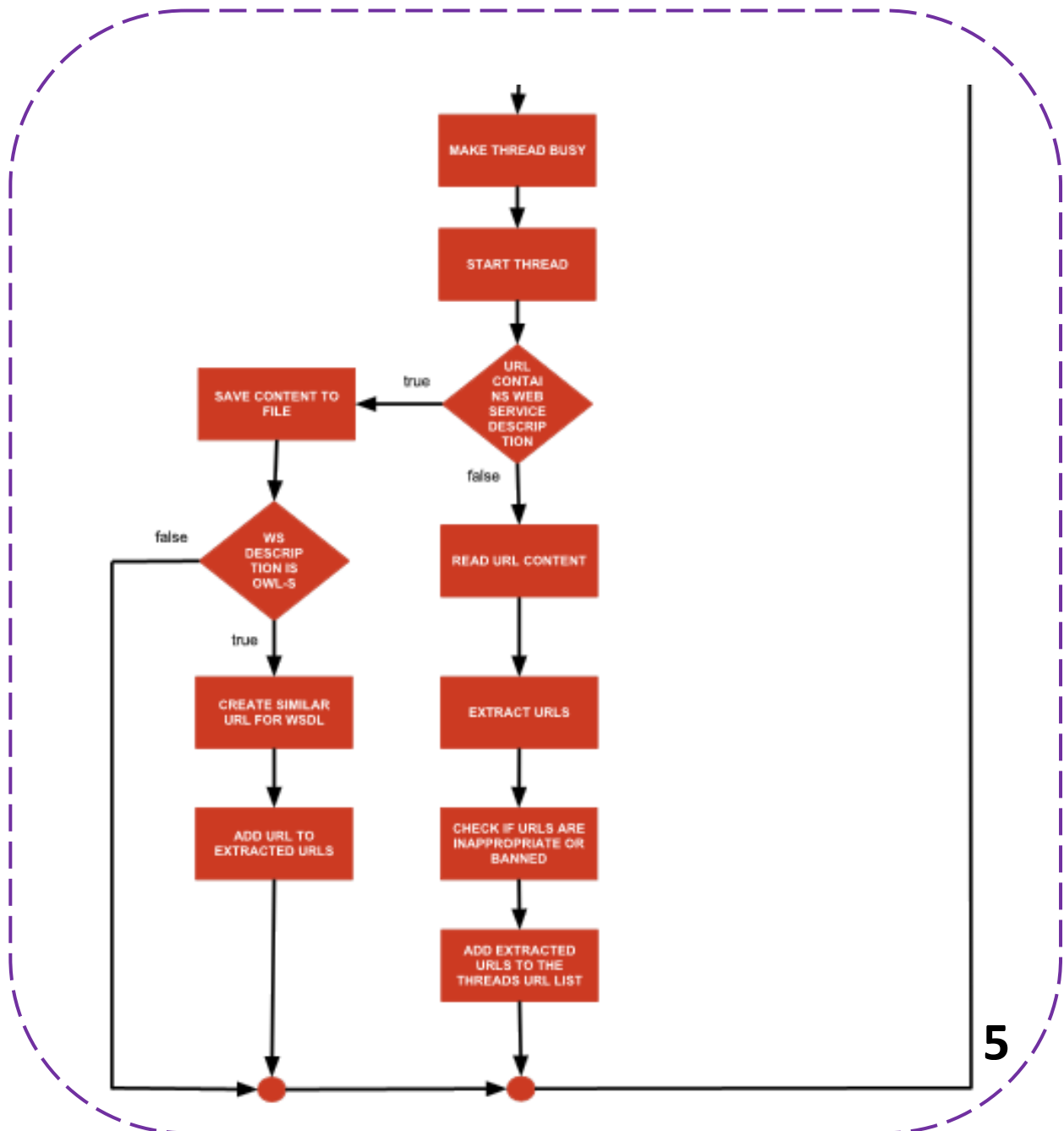


Εικόνα 21. Τμήμα 4 - Ευρετικός κανόνας για domains και εκκίνηση νήματος.

Αφού ολοκληρωθεί και αυτός ο ευριστικός κανόνας, το νήμα ορίζεται ως απασχολημένο και ξεκινά την λειτουργία του.

Εαν η σελίδα που ανατέθηκε στο νήμα περιέχει περιγραφή διαδικτυακής υπηρεσίας, τότε αποθηκεύεται και στην συγκεκριμένη περίπτωση όπου η περιγραφή αντιστοιχούσε σε περιγραφή υπηρεσίας OWL-S, εφαρμόζεται ο ευρετικός κανόνας για δημιουργία αντίστοιχης διεύθυνσης για WSDL που περιγράφηκε στην Ενότητα 5.1.2.10. Σύμφωνα με αυτόν τον κανόνα, δημιουργείται μια αντίστοιχη-όμοια διεύθυνση για περιγραφή WSDL η οποία προστίθεται στις διευθύνσεις που εξάγει ο web crawler (Εικόνα 22).

Αν η σελίδα που επισκέπτεται ο web crawler δεν αντιστοιχεί σε περιγραφή διαδικτυακής υπηρεσίας η διαδικασία διαφοροποιείται.



Εικόνα 22. Τμήμα 5 - Λειτουργία νήματος.

Το περιεχόμενο της σελίδας σαρώνεται ώστε να εντοπισθούν οι διευθύνσεις-υπερσύνδεσμοι που περιέχει. Οι υπερσύνδεσμοι αυτοί αφού περάσουν έλεγχο που εξασφαλίζει πως δεν δείχνουν σε υπερμεγέθη αρχεία, όπως περιγράφεται στην Ενότητα 5.1.2.2, τοποθετούνται στην λίστα με τις σελίδες που έχει εξάγει το νήμα. Η εργασία του νήματος έχει ολοκληρωθεί οπότε η διαδικασία προχωρά με το επόμενο.

Αφού όλα τα νήματα περάσουν από την συγκεκριμένη διαδικασία, η λειτουργία του web crawler τερματίζεται σε περίπτωση που δεν υπάρχουν άλλες διευθύνσεις στο μέτωπο αναζήτησης. Ειδικά, όλη η διαδικασία επαναλαμβάνεται από την αρχή.

5.2. Parsing

Η συλλογή των περιγραφών διαδικτυακών υπηρεσιών ακολουθείται από την επεξεργασία τους και τελικά την ευρετηριοποίησή τους. Το στάδιο προεπεξεργασίας των περιγραφών, είτε αυτές είναι περιγραφές WSDL είτε OWL-S, πραγματοποιείται στους αντίστοιχους parsers, των οποίων ο βασικός σκοπός είναι η απομόνωση της χρήσιμης για την αναζήτηση πληροφορίας.

Ο σχεδιασμός και η υλοποίηση των δύο parsers διαφέρει σε ελάχιστα σημεία, ενώ γενικά η διαδικασία είναι αρκετά απλούστερη συγκριτικά με τις αντίστοιχες ενέργειες που απαιτούνται από ένα πιο απαιτητικό σύστημα όπως ο web crawler που περιγράφηκε παραπάνω. Αυτό συμβαίνει γιατί στα συγκεκριμένα προγράμματα δεν συναντώνται ιδιαίτεροι τεχνικοί περιορισμοί, όπως για παράδειγμα η επίτευξη της μέγιστης δυνατής ταχύτητας, εφόσον η διαδικασία γίνεται μία φορά και off-line. Ωστόσο, κατά την ανάπτυξη των parsers αντιμετωπίστηκαν διάφορα θέματα που θα αναλυθούν στο υπόλοιπο της παρούσας ενότητας, μαζί με την περιγραφή της λειτουργίας τους και τα χαρακτηριστικά τους.

5.2.1. Βασική λειτουργικότητα

Η λειτουργία και των δύο parsers ξεκινά διαβάζοντας τις αντίστοιχες περιγραφές διαδικτυακών υπηρεσιών που ανακάλυψε και αποθήκευσε ο web crawler. Το περιεχόμενο των περιγραφών αυτών φιλτράρεται έτσι ώστε να απομονωθεί η χρήσιμη για την αναζήτηση πληροφορία, η οποία τελικά αποθηκεύεται με τρόπο που διευκολύνει την επεξεργασία της σε ενδιάμεσα αρχεία, τα οποία στη συνέχεια λαμβάνει σαν είσοδο το ευρετήριο. Μοναδική σημαντική διαφορά ανάμεσα στους αλγορίθμους των δύο parsers ήταν η ανάγκη υποστήριξης δύο διαφορετικών τύπων του προτύπου WSDL (1.1 και 2.0).

5.2.2. Πρόσθετα Χαρακτηριστικά

5.2.2.1. Στοιχεία αναζήτησης

Ο προσδιορισμός των στοιχείων των περιγραφών διαδικτυακών υπηρεσιών που πρέπει να απομονωθούν πραγματοποιήθηκε στις Ενότητες 3.1.3, 3.2.3 και 3.3.3, κατά την λεπτομερή περιγραφή των αντίστοιχων προτύπων. Συνοπτικά, τα στοιχεία που απομονώνονται για το κάθε πρότυπο είναι τα εξής:

- *WSDL*: τα κάθε τύπου ονόματα, η τεκμηρίωση και οι είσοδοι και έξοδοι των λειτουργιών
- *SAWSDL*: τα στοιχεία της WSDL περιγραφής και τις αναφορές μοντέλου
- *OWL-S*: το όνομα της υπηρεσίας, η περιγραφή, οι είσοδοι, οι έξοδοι, οι προϋποθέσεις και τα αποτελέσματά της

5.2.2.2. Επιπλοκές περιγραφών

Ενώ κατά την επεξεργασία των περιγραφών υπηρεσιών OWL-S δεν παρουσιάστηκε κάποιο πρόβλημα, δεν συνέβη το ίδιο με τις περιγραφές WSDL. Στις τελευταίες, παρόλο που το πρότυπο είναι καλά ορισμένο, παρατηρήθηκαν διαφορών τύπων ασυνέπειες στις αντίστοιχες περιγραφές, λόγω μη αυστηρής συμμόρφωσης των δημιουργών των υπηρεσιών με τις προσαγές του προτύπου. Ενδεικτικά προβλήματα και ασυνέπειες που έκαναν την υλοποίηση δυσκολότερη είναι η έλειψη στοιχείων των tags όπως το όνομα, η τοποθέτηση του τίτλου σε άλλο στοιχείο ενός tag και η έλειψη ολόκληρων tags.

Η δεύτερη ασυνέπεια είχε να κάνει με την χρήση των ίδιων tags με διαφορετικό τρόπο σε διάφορα αρχεία αλλάζοντας έτσι την πληροφορία των στοιχείων και κατά συνέπεια τον τρόπο δόμησης των περιγραφών.

5.2.2.3. Αρχεία εξόδου

Τα αρχεία εξόδου των δύο parsers είναι απαραίτητο να είναι σε μορφή που θα βοηθά την επεξεργασία τους από το ευρετήριο. Για αυτόν τον λόγο, τα στοιχεία απομονώνονται και αποθηκεύονται σε αρχεία με διαχωριστικά και επισημάνσεις που χρησιμοποιούνται από το ευρετήριο για την αναγνώρισή τους. Με αυτόν τον τρόπο, διαχωρίζεται η διαδικασία του parsing από την διαδικασία της ευρετηριοποίησης, επιταχύνοντας έτσι την κατασκευή του ευρετηρίου.

5.3. Index

Αν ο web crawler αποτελεί πολύ σημαντικό κομμάτι της μηχανής αναζήτησης γιατί συλλέγει τα δεδομένα που αυτή εκθέτει, τότε το ευρετήριο (index) είναι η καρδιά της μηχανής αναζήτησης. Πρόκειται για το υποσύστημα που επεξεργάζεται τα δεδομένα, τα ευρετηριοποιεί και τελικά τα βαθμονομεί αποκρινόμενο στις επερωτήσεις του χρήστη.

Κάθε μια από αυτές τις σημαντικές ενέργειες, μπορεί να υλοποιηθεί με διαφορετικούς τρόπους χρησιμοποιώντας προϋπάρχουσες ή μη τεχνικές και μεθόδους. Στην συγκεκριμένη υποενότητα εξετάζονται και επεξηγούνται οι τεχνικές αποφάσεις που ελήφθησαν για την υλοποίηση ενός ευρετηρίου (index) για περιγραφές διαδικτυακών υπηρεσιών που ανταποκρίνεται στις λειτουργικές απαιτήσεις που τέθηκαν.

Κατά το πρότυπο που ακολουθείτε μέχρι τώρα, αρχικά περιγράφεται συνοπτικά το σύστημα και στην συνέχεια μελετώνται αναλυτικότερα τα χαρακτηριστικά του. Τέλος, παρουσιάζεται το διάγραμμα ροής της λειτουργίας του.

5.3.1. Βασική λειτουργικότητα

Οι πρώτες ενέργειες που πραγματοποιεί το ευρετήριο είναι η ανάγνωση και το parsing των περιγραφών διαδικτυακών υπηρεσιών που εντόπισε ο web crawler. Αρχικά, οι περιγραφές αυτές περνούν διάφορα στάδια επεξεργασίας έτσι ώστε οι όροι τους να έρθουν σε μορφή κατάλληλη για ευρετηριοποίηση. Μόλις ολοκληρωθεί η επεξεργασία των όρων οι περιγραφές προστίθενται στο ευρετήριο από το οποίο δημιουργείται το ανεστραμμένο ευρετήριο. Αφού οικοδομηθούν τα ευρετήρια, για κάθε όρο κάθε εγγράφου υπολογίζονται διάφορα μεγέθη που θα χρησιμεύσουν στην βαθμονόμηση των εγγράφων. Τελικά, το ευρετήριο αναμαίνει τις επερωτήσεις των χρηστών με βάση τις οποίες χρησιμοποιώντας τα ανάλογα μεγέθη επιλέγει, βαθμονομεί και επιστρέφει τα ανάλογα αποτελέσματα στους χρήστες.

5.3.2. Πρόσθετα Χαρακτηριστικά

5.3.2.1. Ρυθμίσεις

Εφόσον το ευρετήριο αποτελεί ένα υποσύστημα στο οποίο υπάρχουν παράμετροι οι οποίες θα μπορούσαν να τροποποιηθούν, ακολουθώντας την πολιτική που προσδιορίστηκε στις λειτουργικές απαιτήσεις, συνοδεύεται από ένα αρχείο ρυθμίσεων. Το περιεχόμενο του συγκεκριμένου αρχείου για το ευρετήριο φαίνεται στην Εικόνα 23.

```

OWLS DOCUMENT WEIGHTS
(dependent on tag)
=====

    filename:[5]
    service name:[7]
    text description:[3]
    input:[5]
    output:[5]

QUERY WEIGHTS
(dependent on input field)
=====

o default search
    keywords:[4]

o advanced search
    inputs:[2]
    outputs:[12]

```

Εικόνα 23. Αρχείο ρυθμίσεων του μηχανισμού ευρετηριοποίησης.

Κατ’ αντιστοιχία με το αρχείο ρυθμίσεων του web crawler και το συγκεκριμένο είναι ένα αρχείο κειμένου στο οποίο προσδιορίζονται οι τροποποιήσιμες παράμετροι, οι οποίες μάλιστα επεξηγούνται για την διευκόλυνση του χρήστη. Στην συγκεκριμένη περίπτωση οι παράμετροι που έχει την δυνατότητα ο χρήστης να τροποποιήσει αποτελούν βάρη τα οποία επηρεάζουν την βαθμονόμηση των αποτελεσμάτων με τρόπο που περιγράφεται στην παράγραφο “βάρη” που ακολουθεί.

Όπως φαίνεται και στην εικόνα, τα βάρη που μπορεί ο χρήστης να τροποποιήσει είναι:

- για τα wsdl αρχεία: το βάρος σε περίπτωση εμφάνισης ενός όρου στον τίτλο, στην τεκμηρίωση, στις λειτουργίες ή στα ονόματα
- για τα owl-s αρχεία: το βάρος σε περίπτωση που ο όρος εμφανίζεται στο όνομα της περιγραφής, στο όνομα της υπηρεσίας, στην τεκμηρίωση και στις εισόδους/εξόδους της υπηρεσίας.

Τέλος, παρέχεται η δυνατότητα τροποποίησης των βαρών με βάση την θέση κάθε όρου στις λέξεις κλειδιά, στις εισόδους και στις εξόδους κατά την σύνθετη αναζήτηση.

5.3.2.2. Parsing

Οι περιγραφές διαδικτυακών υπηρεσιών που συνέλλεξε ο web crawler δεν χρησιμοποιούνται άμεσα από το ευρετήριο καθώς έχουν ήδη υποστεί επεξεργασία από τους parsers. Το περιεχόμενο των αρχείων .wef και .oef, εξόδων του wsdl και του owl-s parser αντίστοιχα, είναι αυτό που ευρετηριοποιείται. Σε πρώτη φάση λοιπόν, το ευρετήριο διαβάζει αυτά τα ενδιάμεσα αρχεία και τα κάνει parse έτσι ώστε τα δεδομένα να διατηρηθούν σε εσωτερικές δομές δεδομένων που διευκολύνουν τις διεργασίες που θα υποστούν στην συνέχεια.

5.3.2.3. Επεξεργασία δεδομένων

Αν και οι parsers έχουν ήδη επεξεργαστεί τα δεδομένα των περιγραφών, η εργασία τους αφορούσε μόνο το φιλτράρισμα των δεδομένων έτσι ώστε να απομονωθούν εκείνα τα τμήματα των περιγραφών που θα μπορούσαν να βοηθήσουν την αναζήτηση. Η συγκεκριμένη ενέργεια όμως σε καμία περίπτωση δεν αρκεί ώστε τα δεδομένα των περιγραφών να ευρετηριοποιηθούν. Πρωτού γίνει αυτό τα δεδομένα είναι ανάγκη να περάσουν από διάφορα στάδια επεξεργασίας που σκοπό έχουν την κανονικοποίηση τους ώστε να μειωθεί όσο είναι δυνατό ο κατακερματισμός του ευρετηρίου. Για βέλτιστα αποτελέσματα τα στάδια επεξεργασίας πρέπει να ακολουθήσουν μια συγκεκριμένη σειρά, η οποία εύκολα εξάγεται αν εξετασθούν οι λειτουργίες που επιτελεί το καθένα, που επιτρέπει την αποτελεσματική επεξεργασία των δεδομένων. Τα στάδια περιγράφονται αναλυτικά στις επόμενες ενότητες.

5.3.2.4. Αφαίρεση αριθμητικών ψηφίων και σημείων στίξης

Οι αριθμοί και τα σημεία στίξης αποτελούν στοιχεία που συναντώνται σε περιγραφές διαδικτυακών υπηρεσιών είτε σε ονόματα μεταβλητών είτε σε human readable text. Όμως τα στοιχεία αυτά δεν προσφέρουν εκμεταλλεύσιμη πληροφορία και άρα δεν είναι απαραίτητα για την αναζήτηση διαδικτυακών υπηρεσιών οπότε αφαιρούνται από τους όρους των ενδιάμεσων αρχείων.

5.3.2.5. Αφαίρεση camelCase και splitting

Ο όρος camelCase [15] χρησιμοποιείται για την περιγραφή του τρόπου γραφής σύμφωνα με τον οποίο περισσότερες από μια λέξεις γράφονται σε έναν σύνθετο όρο ο οποίος αποτελείται από τις

λέξεις αυτές συνεχόμενες, χωρίς κενά μεταξύ τους, με τις λέξεις που ακολουθούν την πρώτη να ξεκινούν με κεφαλαίο. Για να γίνει καλύτερα κατανοητός ο τρόπος γραφής ακολουθεί ένα παράδειγμα στο οποίο φαίνεται η φράση γραμμένη με φυσιολογικό τρόπο και στην συνέχεια με lower και upper camel case.

<i>Normal</i>	<i>Lower camel case</i>	<i>Upper camel case</i>
This is a test phrase	thisIsATestPhrase	ThisIsATestPhrase

Ο παραπάνω τρόπος γραφής αποτελεί συνήθη πρακτική εκ μέρους των προγραμματιστών αφού διευκολύνει την ονομασία σύνθετων όρων σε προγράμματα όμως αποτελεί δυσμενή παράγοντα για την ευρετηριοποίηση και την αναζήτηση αφού ένας camel case όρος αποτελείτε από περισσότερες λέξεις. Για τον λόγο αυτό ήταν ανάγκη να ληφθεί ειδική μέριμνα για τον διαχωρισμό των όρων, ώστε να εξαχθεί η περισσότερη δυνατή πληροφορία.

5.3.2.6. Case normalization

Στο πλαίσιο της κανονικοποίηση των όρων κάποια στιγμή έπρεπε οι όροι να έρθουν στην ίδια μορφή (κεφαλαία ή μικρά) χωρίς να έχει κάποια διαφορά η μια από την άλλη. Εφόσον ολοκληρώθηκε η διαδικασία του διαχωρισμού λόγω camelCase ήταν πλέον δυνατό να πραγματοποιηθεί αυτή η ενέργεια και αυθαίρετα όλα τα γράμματα επιλέχθηκε να μετατραπούν σε μικρά. Η μετρατροπή αυτή είναι απαραίτητη καθώς η κωδικοποίηση του ίδιου γράμματος διαφέρει σε κάθε μορφή με αποτέλεσμα δύο όροι που διαφέρουν μόνο με αυτόν τον τρόπο να αντιμετωπίζονται σαν διαφορετικοί από τον υπολογιστή. Για παράδειγμα ο υπολογιστής αντιμετωπίζει τους όρους «παράδειγμα» και «Παράδειγμα» με διαφορετικό τρόπο ενώ πρόκειται για την ίδια λέξη. Αν δεν λαμβανόταν το παρόν μέτρο η αναζήτηση με τον όρο «παράδειγμα» θα επέστρεφε διαφορετικά αποτελέσματα από την αναζήτηση με τον όρο «Παράδειγμα», πράγμα που δεν είναι επιθυμητό.

5.3.2.7. Αφαίρεση stop words

Το επόμενο στάδιο της επεξεργασίας των περιγραφών αφορά την αφαίρεση των λεγόμενων “stop words” [72] της αγγλικής γλώσσας. Πρόκειται για λέξεις οι οποίες είναι αρκετά συνηθισμένες και δεν προσφέρουν χρήσιμη πληροφορία. Η λίστα των stop words δεν είναι αυστηρά καθορισμένη αλλά οι λίστες που υπάρχουν παρουσιάζουν μικρές διαφορές μεταξύ τους. Οι λέξεις που χρησιμοποιήθηκαν παρατίθενται παρακάτω:

"a", "about", "above", "above", "across", "after", "afterwards",
"again", "against", "all", "almost", "alone", "along", "already",
"also", "although", "always", "am", "among", "amongst", "amoungst",
"amount", "an", "and", "another", "any", "anyhow", "anyone",
"anything", "anyway", "anywhere", "are", "around", "as", "at", "back",
"be", "became", "because", "become", "becomes", "becoming", "been",
"before", "beforehand", "behind", "being", "below", "beside",
"besides", "between", "beyond", "bill", "both", "bottom", "but", "by",
"call", "can", "cannot", "cant", "co", "con", "could", "couldnt",
"cry", "de", "describe", "detail", "do", "done", "down", "due",
"during", "each", "eg", "eight", "either", "eleven", "else",
"elsewhere", "empty", "enough", "etc", "even", "ever", "every",
"everyone", "everything", "everywhere", "except", "few", "fifteen",
"fifty", "fill", "find", "fire", "first", "five", "for", "former",
"formerly", "forty", "found", "four", "from", "front", "full",
"further", "get", "give", "go", "had", "has", "hasnt", "have", "he",
"hence", "her", "here", "hereafter", "hereby", "herein", "hereupon",
"hers", "herself", "him", "himself", "his", "how", "however",
"hundred", "ie", "if", "in", "inc", "indeed", "interest", "into",
"is", "it", "its", "itself", "keep", "last", "latter", "latterly",
"least", "less", "ltd", "made", "many", "may", "me", "meanwhile",
"might", "mill", "mine", "more", "moreover", "most", "mostly", "move",
"much", "must", "my", "myself", "name", "namely", "neither", "never",
"nevertheless", "next", "nine", "no", "nobody", "none", "noone",
"nor", "not", "nothing", "now", "nowhere", "of", "off", "often", "on",
"once", "one", "only", "onto", "or", "other", "others", "otherwise",
"our", "ours", "ourselves", "out", "over", "own", "part", "per",
"perhaps", "please", "put", "rather", "re", "same", "see", "seem",
"seemed", "seeming", "seems", "serious", "several", "she", "should",
"show", "side", "since", "sincere", "six", "sixty", "so", "some",
"somehow", "someone", "something", "sometime", "sometimes",
"somewhere", "still", "such", "system", "take", "ten", "than", "that",
"the", "their", "them", "themselves", "then", "thence", "there",
"thereafter", "thereby", "therefore", "therein", "thereupon", "these",
"they", "thick", "thin", "third", "this", "those", "though", "three",
"through", "throughout", "thru", "thus", "to", "together", "too",
"top", "toward", "towards", "twelve", "twenty", "two", "un", "under",

"until", "up", "upon", "us", "very", "via", "was", "we", "well", "were", "what", "whatever", "when", "whence", "whenever", "where", "whereafter", "whereas", "whereby", "wherein", "whereupon", "wherever", "whether", "which", "while", "whither", "who", "whoever", "whole", "whom", "whose", "why", "will", "with", "within", "without", "would", "yet", "you", "your", "yours", "yourself", "yourselves".

Μελετώντας τα στοιχεία που περιείχαν τα ενδιάμεσα, φιλτραρισμένα αρχεία εντοπίστηκαν και κάποιες άλλες λέξεις οι οποίες ήταν κοινές σε πολλές περιγραφές και δεν προσέφεραν κέρδος πληροφορίας. Οι λέξεις αυτές προστέθηκαν στις παραπάνω και ήταν οι:

"parameter", "header", "part", "result", "token".

5.3.2.8. Εξαγωγή ρίζας όρων

Οι κανονικοποιήσεις των όρων που προηγήθηκαν αφορούσαν την μορφοποίηση των όρων και την εξαγωγή ανεπιθύμητων δεδομένων. Οι τεχνικές αυτές συμβάλλουν ουσιαστικά στην αντιμετώπιση φαινομένων κατακερματισμού του ευρετηρίου αλλά δεν αρκούν. Για την δημιουργία ενός αποτελεσματικού ευρετηρίου οι όροι που περιέχονται στις περιγραφές είναι αναγκαίο να υποστούν και ένα είδος γλωσσικής επεξεργασίας η οποία θα μειώσει περαιτέρω το μέγεθος των δεδομένων και την πολυπλοκότητα των διαδικασιών..

Η γλωσσική αυτή επεξεργασία σκοπό έχει την εξαγωγή των πρωτογενών όρων από τις λέξεις που υπάρχουν στις περιγραφές και χρησιμοποιείται συχνά σε μηχανές αναζήτησης και σε άλλες εφαρμογές ανάκτησης πληροφορίας. Έτσι, αντιμετωπίζονται αποτελεσματικά φαινόμενα κατακερματισμού που οφείλονται σε γλωσσικά φαινόμενα όπως είναι οι κλίσεις και τα παράγωγα των λέξεων. Για παράδειγμα οι όροι "test", "tested" και "tests" θα ήταν βολικό να κατέληγαν στον πρωτογενή όρο "test".

Για την υλοποίηση του συγκεκριμένου σταδίου χρησιμοποιήθηκε ο αλγόριθμος stemming του Porter [46] ο οποίος είναι ο ευρύτερα χρησιμοποιούμενος αλγόριθμος stemming για την αγγλική γλώσσα.

5.3.2.9. Ευρετήριο

Εφόσον τα δεδομένα έχουν κανονικοποιηθεί είναι πλέον δυνατή η δημιουργία του ευρετηρίου των περιγραφών διαδικτυακών υπηρεσιών. Το κλειδί του ευρετηρίου είναι το όνομα της περιγραφής ενώ το περιεχόμενο αντιστοιχεί στους όρους που περιέχει κάθε περιγραφή. Όπως

όμως ειπώθηκε στις λειτουργικές προδιαγραφές το συγκεκριμένο ευρετήριο δεν καλύπτει τις ανάγκες της μηχανής αναζήτησης αφού παρέχει την πληροφορία με τρόπο που δεν είναι αρκετά αποδοτικός. Για να ξεπεραστεί το συγκεκριμένο πρόβλημα χρησιμοποιείται ένα ανεστραμμένο ευρετήριο το οποίο παράγεται απευθείας από το συμβατικό. Η διαφορά έγκειται στο γεγονός πως το ανεστραμμένο ευρετήριο σαν κλειδί χρησιμοποιεί τους όρους και σαν περιεχόμενο τις περιγραφές στις οποίες περιέχονται οι όροι, μειώνοντας την πολυπολοκότητα και άρα διευκολύνοντας τις αναζητήσεις.

5.3.2.10. TF-IDF

Το ανεστραμμένο ευρετήριο είναι πλέον διαμορφωμένο. Επόμενο βήμα είναι ο προσδιορισμός της μεθόδου που θα χρησιμοποιηθεί για την βαθμονόμηση ενός εγγράφου ως προς έναν ή περισσότερους όρους, τους όρους της επερώτησης.

Για τον σκοπό αυτό κρίθηκε σκόπιμο να χρησιμοποιηθεί η μέθοδος tf-idf [73]. Η μέθοδος αυτή προσδιορίζει την σημαντικότητα ενός εγγράφου ως προς έναν όρο χρησιμοποιώντας δύο μεγέθη, την συχνότητα του όρου στο έγγραφο (term frequency - tf) και της (inverse document frequency - idf).

Το πρώτο μέγεθος, η συχνότητα ενός όρου σε ένα έγγραφο, λογικά, αντιστοιχεί στον αριθμό των εμφανίσεων του όρου στο έγγραφο κανονικοποιημένη όμως ώστε να το μήκος του εγγράφου να μην επηρεάζει το αποτέλεσμα. Υπολογίζεται δηλαδή από τον τύπο:

$$Tf = f / l$$

Όπου f η συχνότητα ενός όρου στο έγγραφο και l ο αριθμός των όρων του εγγράφου.

Το δεύτερο μέγεθος, idf, αντιστοιχεί στην σημαντικότητα ενός όρου στο ευρετήριο και υπολογίζεται από τον λογάριθμο του πηλίκου του συνολικού αριθμού των εγγράφων του ευρετηρίου, προς τον αριθμό των εγγράφων που περιέχουν τον όρο. Ακολουθεί ο τύπος που υποδηλώνει τον υπολογισμό:

$$Idf = \log (|D| / 1 + d)$$

Είναι κατανοητό πως το D στον παραπάνω τύπο αντιστοιχεί στον αριθμό των εγγράφων του ευρετηρίου, ενώ το $1+d$ στον αριθμό των εγγράφων που περιέχουν τον όρο αυξημένα κατά ένα ώστε να αποφεύγονται διαιρέσεις με το μηδέν, εάν ο όρος δεν περιέχεται σε κανένα έγγραφο.

Τελικά τα δύο μεγέθη πολλαπλασιάζονται μεταξύ τους ώστε να λάβουμε το μέγεθος $tf-idf$:

$$Tf-idf = tf \times idf$$

Από τον τρόπο που υπολογίζονται τα μεγέθη, είναι σαφές πως ένα έγγραφο θα έχει μεγάλο $tf-idf$ ως προς έναν όρο, εάν ο όρος εμφανίζεται συχνά σε αυτό και σπάνια συνολικά στο ευρετήριο.

5.3.2.11. Επιπλέον βάρη

Σε ειδικές περιπτώσεις όπως οι ιστοσελίδες και οι περιγραφές διαδικτυακών υπηρεσιών που αποτελούν το αντικείμενο της μηχανής αναζήτησης που κατασκευάζεται, η σημαντικότητα ενός όρου σε ένα έγγραφο μπορεί να καθοριστεί και από έναν επιπλέον παράγοντα. Ο παράγοντας αυτός είναι η θέση του στο έγγραφο.

Για παράδειγμα στις ιστοσελίδες ένα όρος που περιέχεται στις κεφαλίδες (headers) είναι συνήθως σημαντικότερος- πιο περιγραφικός από έναν όρο που περιέχεται στο περιεχόμενο. Με τον ίδιο τρόπο, στις περιγραφές διαδικτυακών υπηρεσιών, ένας όρος που περιέχεται στο όνομα της υπηρεσίας είναι πιο σημαντικός από έναν όρο ο οποίος περιέχεται στην τεκμηρίωσή της.

Ακολουθώντας το σκεπτικό αυτό, κρίθηκε σκόπιμη η προσθήκη μιας ακόμη παραμέτρου στην μέθοδο $tf-idf$, του κανονικοποιημένου βάρους w που σχετίζεται με την θέση του όρου στην περιγραφή. Έτσι ο τελικός τύπος μετατράπηκε σε

$$Tf-idf = tf \times idf \times w$$

Αυξάνοντας έτσι το μέγεθος $tf-idf$ για όρους σε κομβικά σημεία των περιγραφών. Είναι αυτά τα βάρη, οι μεταβλητές που ο χρήστης μπορεί να ρυθμίσει στο αρχείο των ρυθμίσεων.

5.3.2.12. Εγγύτητα επερώτησης – εγγράφου

Με βάση τα υπολογιζόμενα μεγέθη $tf-idf$ για τους όρους του ευρετηρίου και της επερώτησης απαιτεί να βρεθεί η εγγύτητα των εγγράφων ως προς την επερώτηση. Η εγγύτητα αυτή αποτελεί το κριτήριο με βάση το οποίο τα αποτελέσματα τελικά βαθμονομούνται, ταξινομούνται και παρουσιάζονται στον χρήστη.

Ο υπολογισμός της εγγύτητας πραγματοποιείται χρησιμοποιώντας το μοντέλο διανυσματικού χώρου (vector space model). Σύμφωνα με το συγκεκριμένο μοντέλο κάθε έγγραφο μπορεί να αναπαρασταθεί σαν ένα διάνυσμα οι αποστάσεις του οποίου αντιστοιχούν σε κάθε όρο του εγγράφου και μάλιστα στην τιμή tf-idf που υπολογίστηκε προηγουμένως. Για τον υπολογισμό της η επερώτηση αναπαρίσταται σαν ένα ακόμα έγγραφο το οποίο συγκρίνεται με όλα τα υπόλοιπα που περιέχουν τουλάχιστον έναν όρο αυτής με τον τύπο που ακολουθεί:

$$\text{Cos}\theta = d \cdot q / \|d\| \|q\|$$

Όπου d ένα έγγραφο-διάνυσμα, q το έγγραφο-διάνυσμα που αντιστοιχεί στην επερώτηση, $d \cdot q$ το εσωτερικό γινόμενο των δύο διανυσμάτων και $\|d\| \|q\|$ το γινόμενο των μέτρων των δύο διανυσμάτων.

5.3.2.13. Απλή αναζήτηση

Η απλή αναζήτηση αποτελεί την προεπιλεγμένη υπηρεσία που προσφέρεται στην μηχανή αναζήτησης για διαδικτυακές υπηρεσίες. Κατά τα γνωστά πρότυπα ο χρήστης αναζητά με βάση μια ή περισσότερες λέξεις κλειδιά που εισάγει σε μορφή επερώτησης. Η επερώτηση αυτή φτάνει στο ευρετήριο με την χρήση μιας ειδικής διαδικτυακής υπηρεσίας (δες παρακάτω).

Περνά από τα στάδια επεξεργασίας που περιγράφηκαν παραπάνω για την επεξεργασία των περιγραφών διαδικτυακών υπηρεσιών, δηλαδή κανονικοποιείται, αφαιρούνται τα stop words κ.ο.κ.

Αφού η επερώτηση καταλήξει σε μορφή που να βελτιστοποιεί την αναζήτηση στο ευρετήριο, εντοπίζονται όλα τα σχετικά με αυτήν έγγραφα που φιλοξενούνται σε αυτό, δηλαδή όλα τα έγγραφα που περιέχουν τουλάχιστον έναν από τους όρους της επερώτησης.

Στην συνέχεια με βάση τα βάρη tf-idf που έχουν προϋπολογιστεί και χρησιμοποιώντας το μοντέλο διανυσματικού χώρου, για κάθε έγγραφο υπολογίζεται ένα score σύμφωνα με το οποίο τα έγγραφα αυτά κατατάσσονται και τελικά αποστέλονται στην ιστοσελίδα για να εμφανισθούν στον χρήστη.

5.3.2.14. Σύνθετη αναζήτηση

Η σύνθετη αναζήτηση διαφοροποιείται από την απλή αφού προσφέρει την επιπλέον δυνατότητα της αναζήτησης σε συγκεκριμένα τμήματα των υπηρεσιών.

Η επεξεργασία της επερώτησης ακολουθεί παρόμοια διαδικασία με την απλή αναζήτηση, ωστόσο, κατά τον υπολογισμό του score για κάθε έγγραφο διαφοροποιούνται σημαντικά τα βάρη στα τμήματα των περιγραφών που έχει επιλέξει ο χρήστης ώστε να αυξάνεται η πιθανότητα τα επιστρεφόμενα έγγραφα να πληρούν τις προδιαγραφές που ο χρήστης θέτει μέσω της σύνθετης επερώτησης.

5.3.2.15. Επικοινωνία Index – Web Interface

Η επικοινωνία ανάμεσα στα δύο αυτά υποσυστήματα πραγματοποιείται χρησιμοποιώντας μια διαδικτυακή υπηρεσία. Η υπηρεσία αυτή παρέχει δύο λειτουργίες που εξυπηρετούν την απλή και την σύνθετη αναζήτηση. Και στις δύο λειτουργίες η είσοδος είναι η επερώτηση του χρήστη, η μορφή της οποία όμως διαφέρει ανάλογα με τον τύπο της αναζήτησης. Σαν έξοδο οι λειτουργίες λαμβάνουν τα βαθμονομημένα αποτελέσματα.

5.4. Web Interface

Το web interface αποτελεί το μέσο αλληλεπίδρασης του χρήστη με την μηχανή αναζήτησης. Πρόκειται για μια δυναμική ιστοσελίδα που ακολουθεί τα πρότυπα των υπάρχοντων ιστοσελίδων μηχανών αναζήτησης η οποία παρέχει συγκεκριμένες δυνατότητες με εύληπτο, λειτουργικό και ελκυστικό τρόπο.

5.4.1. Βασική λειτουργικότητα

Οι βασικές λειτουργίες που προσφέρονται στον χρήστη μέσω της διεπαφής είναι οι ακόλουθες:

- δυνατότητα πραγματοποίησης απλής αναζήτησης για διαδικτυακές υπηρεσίες με βάση λέξεις κλειδιά
- δυνατότητα πραγματοποίησης σύνθετης αναζήτησης για διαδικτυακές υπηρεσίες με βάση λέξεις κλειδιά και άλλα χαρακτηριστικά μιας διαδικτυακής υπηρεσίας
- ενημέρωση για τον τρόπο λειτουργίας και χρήσης της μηχανής αναζήτησης

- προσθήκη της διεύθυνσης μιας περιγραφής διαδικτυακής υπηρεσίας χειροκίνητα, έτσι ώστε να συμπεριληφθεί στις προς επίσκεψη από τον web crawler σελίδες

5.4.2. Πρόσθετα Χαρακτηριστικά

5.4.2.1. Απλή αναζήτηση

Η αρχική οθόνη του web interface κατά αντιστοιχία με τις υπόλοιπες μηχανές αναζήτησης είναι ανάγκη να παρουσιάζει την βασική λειτουργία της μηχανής αναζήτησης για διαδικτυακές υπηρεσίες, δηλαδή την απλή αναζήτηση τους με τη χρήση λέξεων κλειδιών. Ο χρήστης προτρέπεται να εισάγει τους όρους με βάση τους οποίους θέλει να αναζητήσει μια υπηρεσία και στην συνέχεια να ξεκινήσει την αναζήτηση.

Όπως περιγράφηκε και στις τεχνικές απαιτήσεις του ευρετηρίου, κατά τη απλή αναζήτηση οι όροι της επρώτησης του χρήστη αντιμετωπίζονται όλοι με τον ίδιο τρόπο.

5.4.2.2. Σύνθετη αναζήτηση

Σε περίπτωση που οι χρήστες θέλουν να πραγματοποιήσουν αναζήτηση με διαφορετικά κριτήρια, προσφέρεται η δυνατότητα της σύνθετης αναζήτησης. Οι όροι της επρώτησης του χρήστη σε αυτή την περίπτωση δεν αντιμετωπίζονται με τον ίδιο τρόπο καθώς δίνεται η δυνατότητα στον χρήστη να προσδιορίσει σε ποιο σημείο μιας περιγραφής θα ήθελε να εμφανίζονται οι όροι της επρώτησης.

5.4.2.3. Άλλοι σύνδεσμοι

Πέρα από τις προαναφερθείσες λειτουργίες, που συνδέονται άμεσα με την λειτουργία μιας μηχανής αναζήτησης, στην ιστοσελίδα θα υπάρχουν και άλλοι σύνδεσμοι.

Στον χρήστη θα παρέχεται δυνατότητα να ενημερωθεί για την μηχανή αναζήτησης και τον τρόπο λειτουργίας της αλλά και να εισάγει έναν σύνδεσμο προς μια δική του περιγραφή διαδικτυακής υπηρεσίας. Η τελευταία λειτουργία δίνει την δυνατότητα σε χρήστες να καταστήσουν τις περιγραφές τους προσιτές μέσω της μηχανής αναζήτησης, αν τυχόν δεν έχουν βρεθεί ήδη.

5.4.2.4. Εμφάνιση

Μαζί με τα αποτελέσματα, η εμφάνιση αποτελεί το δεύτερο στοιχείο που επηρεάζει την ικανοποίηση του χρήστη σε μια ιστοσελίδα. Η σελίδα κατά σύμβαση θα είναι απλή, χωρίς υπερβολικό περιεχόμενο, ώστε να εξασφαλίζεται η εστίαση στον βασικό της σκοπό, την αναζήτηση. Στον χρήστη θα πρέπει να δίνεται η δυνατότητα να πραγματοποιήσει τις ενέργειες που θέλει με διαισθητικό τρόπο σε ένα ευχάριστο περιβάλλον. Για τον λόγο αυτό ακολουθούνται οι συμβατικές αρχές σχεδίασης μια ιστοσελίδας μηχανής αναζήτησης που παρατηρούνται σε όλες τις υπόλοιπες. Παραδείγματα εμφάνισης των σελίδων της γραφική διεπαφής της μηχανής αναζήτησης παρουσιάζονται στο πλαίσιο των περιπτώσεων χρήσης στο Κεφάλαιο 6.

5.5. Τεχνολογίες, Γλώσσες και Εργαλεία Ανάπτυξης

Η πρώτη σημαντική απόφαση που λήφθηκε για την υλοποίηση της μηχανής αναζήτησης είχε να κάνει με την γλώσσα προγραμματισμού που χρησιμοποιήθηκε για την υλοποίησή της.

Όσον αφορά τον web crawler, η επιλογή αυτή έπρεπε να λαμβάνει υπόψη δύο παράγοντες: το έργο που θα επιτελούσε το πρόγραμμα και τις ειδικές γνώσεις που απαιτούνται από την μεριά του προγραμματιστή. Ο πρώτος παράγοντας περιλαμβάνει την ικανότητα της γλώσσας να υποστηρίζει τις λειτουργίες που καλείται να εκτελέσει ο web crawler ώστε να επιτελέσει το έργο του. Μάλιστα, αφού ο web crawler αποτελεί απαιτητικό πρόγραμμα, όπου η ταχύτητα και η σταθερότητα διαδραματίζουν καθοριστικό ρόλο, έπρεπε η υποστήριξη των λειτουργιών να συνοδεύεται από αποδοτικότητα. Ο δεύτερος παράγοντας εξασφάλιζε ότι δεν θα χανόταν ένα σημαντικό χρονικό διάστημα μέχρι την εξοικείωση του προγραμματιστή με μια νέα γλώσσα (learning curve). Η προγραμματιστική γλώσσα που τελικά επιλέχθηκε για την υλοποίηση του web crawler είναι η γλώσσα Java [42] αφού πληρούσε τις προϋποθέσεις που τέθηκαν. Οι υποστηριζόμενη λειτουργικότητα από την γλώσσα Java κρίθηκε απόλυτα επαρκής, ενώ στο σύστημα όπου θα έτρεχε το πρόγραμμα, το κέρδος από την χρήση μιας αποδοτικότερης σε θέμα ταχύτητας γλώσσας ήταν αμελητέο. Αφού επιλέχθηκε η γλώσσα προγραμματισμού έπρεπε να αποφασιστεί ποια από τις διαθέσιμες βιβλιοθήκες της Java για επικοινωνία με ιστοσελίδες θα χρησιμοποιούνταν. Σημειώνεται πως η επικοινωνία αυτή πραγματοποιείται με τον κλασσικό τρόπο αίτησης και απόκρισης του πρωτοκόλλου HTTP (HTTP request-response). Οι επικρατέστερες διαθέσιμες επιλογές ήταν η native βιβλιοθήκη της Java για τον σκοπό αυτό, η java.net [55], και την αντίστοιχη βιβλιοθήκη της Apache η οποία λέγεται HttpClient [5]. Αν και

η HttpClient προσφέρει ένα αυξημένες δυνατότητες, κρίθηκε πως η java.net ήταν πιο απλή στην χρήση και επαρκούσε για το συγκεκριμένο σκοπό, οπότε επιλέχθηκε αυτή.

Όσον αφορά την υλοποίηση των δύο parsers χρησιμοποιήθηκε επίσης η γλώσσα Java. Η χρήση της ίδιας γλώσσας προάγει την διαλειτουργικότητα των συστατικών στοιχείων της μηχανής αναζήτησης. Επιπλέον, η συγκεκριμένη γλώσσα επιλέχθηκε γιατί παρέχει βιβλιοθήκες που καθιστούν την εργασία που απαιτείται από τους parsers, δηλαδή την επεξεργασία κειμένου ιδιαίτερα εύκολη, αφού παρέχουν δεκάδες έτοιμες συναρτήσεις οι οποίες επιτρέπουν την επεξεργασία των περιγραφών με κάθε απαιτούμενο τρόπο.

Για λόγους διαλειτουργικότητας, η γλώσσα Java χρησιμοποιήθηκε και για την ανάπτυξη του μηχανισμού ευρετηριοποίησης. Επιπλέον, η συγκεκριμένη γλώσσα προσφέρει έναν ιδιαίτερα εύχρηστο τρόπο και εργαλεία για την μετατροπή οποιουδήποτε προγράμματος σε διαδικτυακή υπηρεσία, κάτι που αποδείχθηκε ιδιαίτερα χρήσιμο κατά την επικοινωνία του μηχανισμού ευρετηριοποίησης με τη γραφική διεπαφή.

Η απαιτούμενη επικοινωνία ανάμεσα στο web interface και το ευρετήριο απαιτούσε την χρήση τεχνολογιών που διευκολύνουν την διαδικασία. Η σύνδεση ανάμεσα στα δύο συστήματα πραγματοποιείται χρησιμοποιώντας διαδικτυακές υπηρεσίες, μια για κάθε τύπο αναζήτησης οι οποίες απλά επιτρέπουν την ανταλλαγή δεδομένων ανάμεσα στα δύο υποσυστήματα. Η ίδια η ιστοσελίδα χρησιμοποιεί την τεχνολογία JSP(Java Server Pages) η οποία διευκολύνει την διαχείριση των δεδομένων που στέλνονται από το ευρετήριο.

Το αισθητικό κομμάτι της ιστοσελίδας υλοποιήθηκε χρησιμοποιώντας τα πλέον διαδεδομένα πρότυπα για τον σκοπό αυτό, CSS (Cascading Style Sheets) [17] και JavaScript [43]. Και οι δύο γλώσσες χρησιμοποιούνται για την βελτίωση του τρόπου παρουσίασης των δεδομένων μια ιστοσελίδας, με τη JavaScript να χρησιμοποιείται κυρίως για δυναμικές αισθητικές παρεμβάσεις.

Το περιβάλλον ανάπτυξης που χρησιμοποιήθηκε για τον προγραμματισμό της μηχανής αναζήτησης σε Java ήταν το IDE Netbeans [51] της Oracle (πρώην Sun), το οποίο αποτελεί ένα εύχρηστο ολοκληρωμένο περιβάλλον ανάπτυξης με υψηλές δυνατότητες παραμετροποίησης, αυτοματοποίησης και ολοκλήρωσης.

Επιπρόσθετα, για τον εντοπισμό και την αντιμετώπιση προβλημάτων κατά την ανάπτυξη του κώδικα, χρησιμοποιήθηκε το εργαλείο VisualVM [76]. Το συγκεκριμένο εργαλείο προσφέρει δυνατότητες κατασκευής προφίλ (profiling) του προγράμματος, που απεικονίζει τη δομή της

ροής του προγράμματος και τη διαχείριση μνήμης, και επιτρέπει τον εντοπισμό σφαλμάτων στη διαχείριση μνήμης (memory leaks). Επιπλέον, προσφέρει δυνατότητες ανάλυσης της απόδοσης των προγραμμάτων, διευκολύνοντας έτσι την βελτίωσή τους.

Το αισθητικό κομμάτι υλοποιήθηκε χρησιμοποιώντας το περιβάλλον ανάπτυξης Microsoft Expression Web [49] το οποίο χρησιμοποιείται ειδικά για τον σχεδιασμό ιστοσελίδων.

Έλεγχος Λειτουργικότητας, Απόδοση και Σενάρια Χρήσης

Στο παρόν Κεφάλαιο παρουσιάζονται στοιχεία για την εκτέλεση του συστήματος, πειράματα, καθώς και σενάρια χρήσης.

6.1. Περιβάλλον και Διαδικασία Εκτέλεσης Μηχανής Αναζήτησης

Τα τεχνικά χαρακτηριστικά του υπολογιστή όπου εκτελείται η μηχανή αναζήτησης είναι ικανοποιητικά για έναν υπολογιστή που χρησιμοποιείται σαν σταθμός εργασίας, αφού παρέχει σημαντική επεξεργαστική ισχύ με δύο πυρήνες, αρκετό αποθηκευτικό χώρο και μνήμη RAM 2GB. Η σύνδεση που παρέχεται στο Πανεπιστήμιο μέσω του διαπανεπιστημιακού δικτύου GRNet είναι T1.

Ωστόσο, η εκτέλεση μιας μηχανής αναζήτησης είναι αρκετά απαιτητικό έργο. Τα υποσυστήματα της είτε λόγω της ίδιας της λειτουργίας τους, είτε λόγω του όγκου των δεδομένων προς επεξεργασία, απαιτούν σημαντική υπολογιστική ισχύ, με έμφαση στην επεξεργαστική ισχύ, την μνήμη και τον αποθηκευτικό χώρο. Για τον λόγο αυτό συστήματα μεγάλης κλίμακας απαιτούν κέντρα διαχείρισης των δεδομένων όπου βρίσκονται πολλές φορές εκατοντάδες ή και χιλιάδες διακομιστές που συμβάλλουν στην εκτέλεση των ενεργειών χρησιμοποιώντας τεχνικές παραλληλοποίησης και κατανομής του φόρτου.

Η δυσκολία που παρουσιάζει η ανάπτυξη ενός λειτουργικά αποδοτικού web crawler αναλύθηκε εκτενώς στα προηγούμενα Κεφάλαια 4 και 5. Συνοπτικά υπενθυμίζεται πως πρέπει να είναι ταχύς, να διαχειρίζεται σωστά τους διαθέσιμους πόρους είτε υπολογιστικής ισχύος είτε δικτύου και να είναι ανθεκτικός στα τυχόν προβλήματα.

Ο web crawler που σχεδιάστηκε και υλοποιήθηκε στο πλαίσιο της παρούσας πτυχιακής εργασίας έτρεχε προσεγγιστικά για ένα διάστημα 5 μηνών. Ας σημειωθεί πως το μεγαλύτερο μέρος του πειραματικού διαστήματος αυτού αφορούσε τον εντοπισμό και την διόρθωση σφαλμάτων, την βελτιστοποίηση της απόδοσης και την γενική εποπτεία του. Η ουσιαστική χρονική περίοδος κατά την οποία έτρεξε όντας πια κοντά στην τελική του μορφή κυμαίνεται ανάμεσα σε 1-2 μήνες. Στο διάστημα αυτό ο web crawler εντόπισε περίπου 1500 περιγραφές διαδικτυακών υπηρεσιών εκ των οποίων το 97% αντιστοιχεί σε περιγραφές WSDL και το υπόλοιπο 3% σε

περιγραφές OWL-S. Οι τελευταίες βελτιστοποιήσεις που πραγματοποιήθηκαν πιθανότατα θα υποβοηθούσαν ακόμη περισσότερο το έργο του. Η πρόοδος που πραγματοποιήθηκε στο διάστημα αυτό γίνεται αντιληπτή αν αναλογιστεί κανείς πως ενώ στις αρχές ο αριθμός των σελίδων που εξέταζε σε 24 ώρες δεν ξεπερνούσε τις 100.000, τελικά ο αριθμός αυτός έπειτα από τις βελτιστοποιήσεις έφτασε σε μερικές περιπτώσεις να αγγίζει τις 1.500.000 σελίδες, αριθμό αρκούντως ικανοποιητικό δοθέντος και του διαθέσιμου υλικού. Στον Πίνακα 3 ακολουθούν μερικά προσεγγιστικά στοιχεία για τα μεγέθη των σελίδων που συνέλεξε κατά την πρόοδό του.

Πίνακας 3. Προσεγγιστικά μεγέθη διαδικασίας crawling ιστοσελίδων.

Αριθμός μοναδικών σελίδων που εξέτασε	9.300.000
Αριθμός σελίδων που εξέτασε (με διπλότυπα)	22.000.000
Αριθμός εξεταζόμενων σελίδων ανά 24 ώρες	1.500.000
Αριθμός διαδικτυακών υπηρεσιών που εντοπίστηκαν	1.500

Οι απαιτήσεις όσον αφορά στην απόδοση των υπολοίπων τμημάτων της μηχανής αναζήτησης δεν ήταν εξίσου σημαντική. Οι parsers και το ευρετήριο αποτελούν συστήματα των οποίων η λειτουργία και το γεγονός πως το κύριο μέρος αυτής πραγματοποιείται σε μεγάλο βαθμό off-line, δηλαδή όχι σε πραγματικό χρόνο, να μην απαιτούν ιδιαίτερη μέριμνα για την απόδοση τους σε επίπεδο ταχύτητας επεξεργασίας.

6.2. Διαδικασία Πειραματισμού - Απόδοση

Η πολυνηματική υλοποίηση του web crawler αποτέλεσε τον σημαντικότερο παράγοντα βελτίωσης της ταχύτητας του. Ο προσδιορισμός του αριθμού νημάτων που εξασφάλιζαν την βέλτιστη ταχύτητα πραγματοποιήθηκε με την διεξαγωγή πειραμάτων. Για την καλύτερη δυνατή αξιοπιστία των πειραμάτων αυτών, πραγματοποιήθηκαν υπό συγκεκριμένες συνθήκες. Τα τρεξίματα του web crawler πραγματοποιήθηκαν στον ίδιο υπολογιστή και δίκτυο μέσα στην ίδια ημέρα. Επίσης, χρησιμοποιήθηκε το ίδιο δείγμα ιστοσελίδων για εξέταση για να εγγυηθεί η ανεξαρτησία των αποτελεσμάτων από αυτόν τον παράγοντα. Τα πειράματα πραγματοποιήθηκαν με εναλλακτικά τρεξίματα του web crawler χρησιμοποιώντας διαφορετικό αριθμό νημάτων κάθε φορά. Τα στοιχεία που συλλέχθηκαν σχετίζονταν με τον χρόνο που απαιτούνταν ώστε ο web crawler να εξετάσει 10.000 ιστοσελίδες. Παρατηρήθηκε ότι ο χρόνος αυτός μειωνόταν καθώς

αυξανόταν ο αριθμός των νημάτων μέχρι που έφτανε σε κάποιο κατώφλι πέρα από το οποίο ο χρόνος αυξανόταν ξανά λόγω μεγάλου φόρτου και κόστους συγχρονισμού των νημάτων (bottleneck). Το κατώφλι αυτό οφείλεται κατα κύριο λόγο στον τρόπο με τον η υλοποίηση έγινε πολυνηματική που εξετάσθηκε στην υποενότητα 5.1.2.3. Κατά την προετοισία κάθε νήματος, αυτό καλείται να προσθέσει τους συνδέσμους που εξήγαγε από το τελευταίο URL που εξέτασε στο μέτωπο αναζήτησης. Όταν ο αριθμός των νημάτων είναι μεγάλος, λόγω και των περιορισμών του υλικού που υπήρχαν στην συγκεκριμένη περίπτωση, όλα τα νήματα πρέπει να ακολουθήσουν αυτή τη διαδικασία πρώτου τους ανατεθεί το επόμενο προς εξέταση URL, με αποτέλεσμα να εισάγεται η καθυστέρηση που δημιουργεί το κατώφλι.

Πειραματικά διαπιστώθηκε πως με την χρήση περίπου 90 νημάτων εξασφαλιζόταν η καλύτερη δυνατή ταχύτητα και άρα απόδοση του web crawler.

Με το ίδιο κίνητρο, την βελτιστοποίηση δηλαδή της απόδοσης, πραγματοποιήθηκαν πειράματα και στο υποσύστημα του ευρετηρίου. Σε αντίθεση με τον web crawler όμως, ο κρίσιμος παράγοντας σε αυτή την περίπτωση δεν ήταν η ταχύτητα αλλά η ακρίβεια και ανάκληση των αποτελεσμάτων.

Τα πειράματα σκοπό είχαν να υποβοηθήσουν τον προσδιορισμό των τιμών των βαρών που θα εξασφάλιζαν την καλύτερη δυνατή ακρίβεια και ανάκληση. Αντικείμενο των πειραμάτων ήταν ένα υποσύνολο των διαδικτυακών υπηρεσιών που ανακαλύφθηκαν με την χρήση του web crawler τα οποία μάλιστα είχαν παρεμφερή θεματολογία.

Η διεξαγωγή των πειραμάτων πιστοποίησε πως οι λογικές αποφάσεις για τα μεγέθη των βαρών επέφεραν τα βέλτιστα αποτελέσματα. Συγκεκριμένα, αν τα αντικείμενα στους δύο τύπους περιγραφών είναι η ονομασία, η περιγραφή, οι λειτουργίες και οι λοιποί όροι, μπορούν να ταξινομηθούν κατά σειρά σημαντικότητας με τον τρόπο που ακολουθεί:

ονομασία > λειτουργίες > περιγραφή > λοιποί όροι

Όπως φαίνεται, ο τίτλος και οι λειτουργίες των περιγραφών αποτελούν τα πλέον σημαντικά στοιχεία των περιγραφών περιγραφής που συμβάλλουν ουσιαστικά στην ανακάλυψή τους. Η συμπεριφορά αυτή μπορεί να θεωρηθεί αναμενόμενη καθώς αποτελούν τα κατεξοχήν στοιχεία που θα μπορούσαν να περιγράφουν την υπηρεσία που προσφέρεται και τον τρόπο με τον οποίο αυτό συμβαίνει, τις επιμέρους ενέργειες δηλαδή που καθιστούν την προσφορά της υπηρεσίας εφικτή. Σε συνδυασμό με την συνήθη καλή πρακτική των προγραμματιστών, που ακολουθείται για πρακτικούς λόγους όπως είναι η κατανόηση, η γρήγορη εκμάθηση, η αναγνωσιμότητα κλπ.

του κώδικα, να παρέχουν αντιπροσωπευτικά ονόματα σε κάθε τέτοιο στοιχείο δικαιολογούν την τάση αυτή.

Βεβαίως, η σημασία της διαθέσιμης περιγραφής αν υπάρχει πρέπει να λαμβάνεται σοβαρά υπόψη ενώ και οι άλλοι όροι διαδραματίζουν σημαντικό ρόλο και λόγω του πλήθους τους. Ακολουθούν (Εικόνα 24) τα βάρη στα οποία καταλήξαμε με βάση αυτές τις παρατηρήσεις. Περισσότερες λεπτομέρειες για τα βάρη παρουσιάζονται στην Ενότητα 5.2.3.11.

WSDL DOCUMENT WEIGHTS (depending on tag) =====	
title:	[7]
documentation:	[3]
operation:	[5]
word:	[2]
OWLS DOCUMENT WEIGHTS (depending on tag) =====	
filename:	[5]
service name:	[7]
text description:	[3]
input:	[5]
output:	[5]

Εικόνα 24. Τα προεπιλεγμένα βάρη των επιμέρους στοιχείων των δύο προτύπων περιγραφών διαδικτυακών υπηρεσιών.

Επιπλέον πειράματα είχαν ως σκοπό να διαπιστώσουν την συμπεριφορά του συστήματος ως προς τις διαφορετικές περιγραφές διαδικτυακών υπηρεσιών. Υπενθυμίζεται ότι αν και οι περιγραφές WSDL και OWL-S ευρετηριοποιούνται και διαχειρίζονται με το ίδιο μοντέλο, υπάρχει διαφορά στην πληροφορία που έχει εξαχθεί από αυτές. Πιο συγκεκριμένα οι πληροφορίες που εξάγονται από τις OWL-S περιγραφές είναι περισσότερες και είναι εντεταγμένες σε οντολογίες, επομένως αντιπροσωπεύουν με καλύτερο τρόπο τις αντίστοιχες έννοιες. Για αυτόν τον λόγο, τα αποτελέσματα αναζήτησης που περιγράφονται σε OWL-S στα πειράματα έχει παρατηρηθεί ότι συνήθως είναι καταλληλότερα για τις ανάγκες του χρήστη. Ωστόσο, πρέπει να σημειωθεί πως καθώς οι περιγραφές OWL-S που συλλέχθηκαν

αντιπροσωπεύουν ένα μικρό μόνο τμήμα των συνολικών περιγραφών είναι επιθυμητή η μελλοντική πραγματοποίηση περαιτέρω πειραμάτων ώστε να εξαχθούν ασφαλέστερα συμπεράσματα.

6.3. Σενάρια Χρήσης

6.3.1. Απλή αναζήτηση

Η απλή αναζήτηση αποτελεί την προεπιλεγμένη μορφή αναζήτησης που προσφέρεται στους χρήστες από την μηχανή αναζήτησης που κατασκευάστηκε. Χρησιμοποιώντας τις λέξεις κλειδιά που βρίσκονται στην επερώτηση του χρήστη και αντιπροσωπεύουν την ανάγκη του ως προς τις παροχές της υπηρεσίας, προσδιορίζει τις περιγραφές εκείνες με την μεγαλύτερη εγγύτητα σύμφωνα με τις τεχνικές που προσδιορίστηκαν στα κεφάλαια 5.3.2.10-5.3.2.12. Ο χρήστης, κατά τα καθιερωμένα πρότυπα, καλείται να εισάγει την επερώτηση του στο πεδίο εισόδου και στην συνέχεια να πατήσει το κουμπί αναζήτησης. Η οθόνη απλής αναζήτησης φαίνεται στην Εικόνα 25 που ακολουθεί:



Εικόνα 25. Η αρχική οθόνη στην οποία πραγματοποιείται η απλή αναζήτηση.

Η συγκεκριμένη περίπτωση αναζήτησης αποτελεί μια ευρύτερη εξέταση των περιγραφών που έχουν ευρετηριοποιηθεί συγκριτικά με την σύνθετη αναζήτηση που θα εξετασθεί παρακάτω. Σκοπός είναι να παρέχει στον χρήστη μια διευρενητική ματιά στις διαθέσιμες υπηρεσίες χωρίς να χρειαστεί να προσδιοριστούν λεπτομέρειες που δεν αφορούν καθαρά την παρεχόμενη υπηρεσία. Έτσι ο χρήστης μπορεί να ψάξει για παράδειγμα για μια διαδικτυακή υπηρεσία με θέμα και λειτουργία που να αφορά ευρεία θέματα όπως η πρόγνωση του καιρού (weather forecast) χωρίς να χρειαστεί να γίνει πιο συγκεκριμένος. Τα αποτελέσματα της συγκεκριμένης αναζήτησης φαίνονται στην Εικόνα 26.

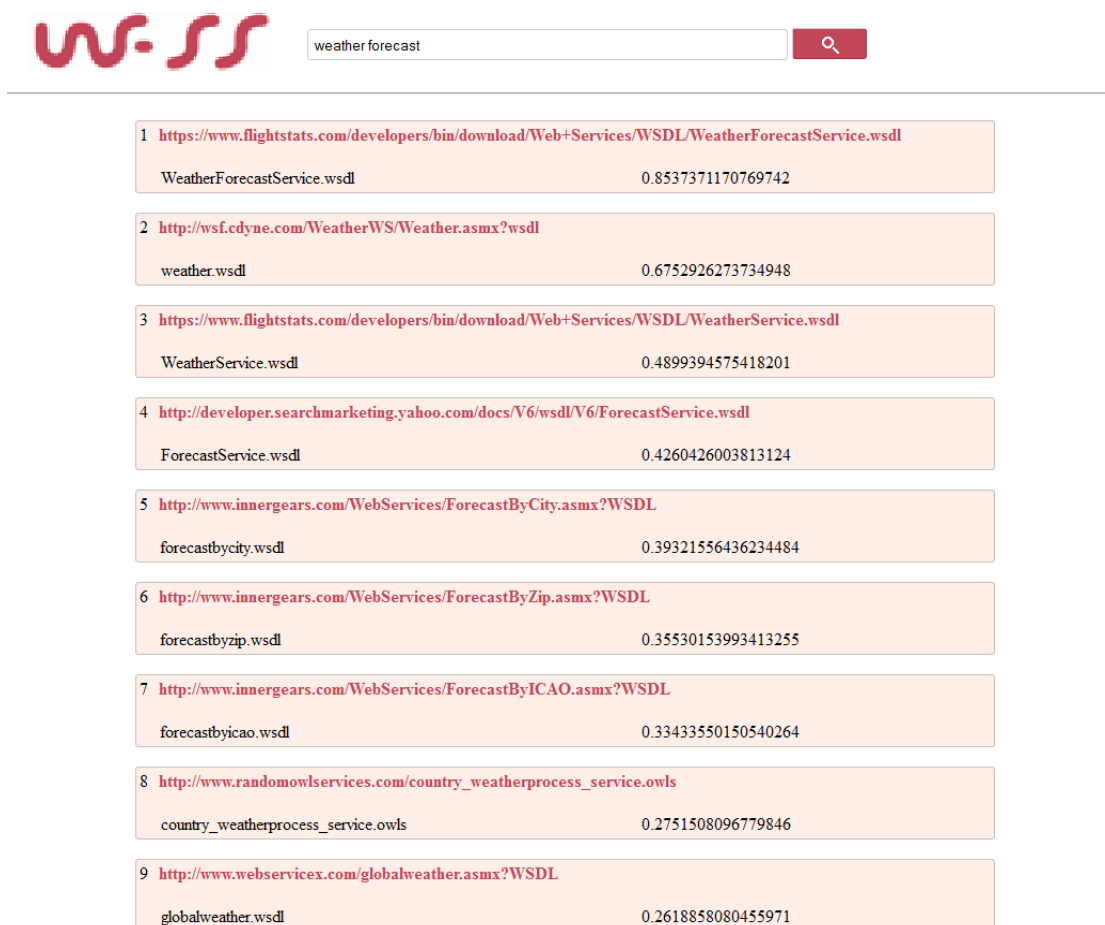
1	https://www.flightstats.com/developers/bin/download/Web+Services/WSDL/WeatherForecastService.wsdl	WeatherForecastService.wsdl	0.9568865390269219
2	http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl	weather.wsdl	0.7604589845925784
3	https://www.flightstats.com/developers/bin/download/Web+Services/WSDL/WeatherService.wsdl	WeatherService.wsdl	0.7183063605611286
4	http://developer.searchmarketing.yahoo.com/docs/V6/wsdl/V6/ForecastService.wsdl	ForecastService.wsdl	0.6513323650067248
5	http://www.randomowlsservices.com/country_weatherprocess_service.owl	country_weatherprocess_service.owl	0.2751508096779846
6	http://www.webservicex.com/globalweather.asmx?WSDL	globalweather.wsdl	0.22326697120007277
7	http://www.innergears.com/WebServices/ForecastByCity.asmx?WSDL	forecastbycity.wsdl	0.14495885476450587
8	http://www.innergears.com/WebServices/ForecastByZip.asmx?WSDL	forecastbyzip.wsdl	0.14279396566172195
9	http://www.innergears.com/WebServices/ForecastByICAO.asmx?WSDL	forecastbyicao.wsdl	0.14132525436183543

Εικόνα 26. Αποτελέσματα απλής αναζήτησης για υπηρεσίες που αφορούν την πρόγνωση του καιρού (weather forecast).

Τα αποτελέσματα παρουσιάζονται ταξινομημένα σε άξουσα σειρά αναφορικά με την εγγύτητα του καθενός στην επρώτηση. Αξίζει να σημειωθεί πως σε αυτά διακρίνονται περιγραφές και

των δύο υποστηριζόμενων προτύπων (wsdl, owl-s). Στο πλαίσιο της παροχής δυνατοτήτων τροποποίησης της λειτουργίας της μηχανής αναζήτησης, παρέχεται η δυνατότητα παρεμβολής στις μεθόδους που τα καθορίζουν. Ο τρόπος με τον οποίο αυτό επιτυγχάνεται είναι η προσαρμογή βαρών που αφορούν την σημασία των όρων ανάλογα με το πεδίο μιας περιγραφής στο οποίο ανήκουν χρησιμοποιώντας το ειδικό αρχείο για τον σκοπό αυτό. Λεπτομέρειες για τα βάρη παρουσιάζονται στο υποκεφάλαιο 5.3.2.11, ενώ η επιχειρηματολογία πίσω από τις προεπιλεγμένες τιμές τους φαίνεται στην αμέσως προηγούμενη υποενότητα, 6.2.

Οι επιπτώσεις της τροποποίησης των βαρών αυτών παρουσιάζονται στο επόμενο παράδειγμα. Η επόμενη αναζήτηση πραγματοποιείται με τις ίδιες λέξεις κλειδιά, αυτή τη φορά όμως τα βάρη έχουν τροποποιηθεί έτσι ώστε να δίνεται μεγάλη σημασία στην ονομασία της περιγραφής, ενώ οι υπόλοιποι όροι των περιγραφών έχουν ίσα αλλά αρκετά μικρότερα βάρη. Είναι ευνόητο λοιπόν, πως τα βάρη αυτά προκαθορίζουν πως οι υπηρεσίες που θα επιστραφούν και η σειρά τους θα εξαρτάται κατά κύριο λόγο από τον τίτλο της περιγραφής. Τα αποτελέσματα της αναζήτησης με τα τροποποιημένα βάρη φαίνονται στην Εικόνα 27.



	weather forecast	
1	https://www.flightstats.com/developers/bin/download/Web+Services/WSDL/WeatherForecastService.wsdl	
	WeatherForecastService.wsdl	0.8537371170769742
2	http://wsf.cdyne.com/WeatherWS/Weather.asmx?wsdl	
	weather.wsdl	0.6752926273734948
3	https://www.flightstats.com/developers/bin/download/Web+Services/WSDL/WeatherService.wsdl	
	WeatherService.wsdl	0.4899394575418201
4	http://developer.searchmarketing.yahoo.com/docs/V6/wsdl/V6/ForecastService.wsdl	
	ForecastService.wsdl	0.4260426003813124
5	http://www.innergears.com/WebServices/ForecastByCity.asmx?WSDL	
	forecastbycity.wsdl	0.39321556436234484
6	http://www.innergears.com/WebServices/ForecastByZip.asmx?WSDL	
	forecastbyzip.wsdl	0.35530153993413255
7	http://www.innergears.com/WebServices/ForecastByICAO.asmx?WSDL	
	forecastbyicao.wsdl	0.33433550150540264
8	http://www.randomowlservices.com/country_weatherprocess_service.owl	
	country_weatherprocess_service.owl	0.2751508096779846
9	http://www.webservicex.com/globalweather.asmx?WSDL	
	globalweather.wsdl	0.2618858080455971

Εικόνα 27. Αποτελέσματα αναζήτησης για την πρόγνωση του καιρού με τροποποιημένα βάρη.

Η σύγκριση των δύο συστάδων αποτελεσμάτων μας δείχνει ότι και πάλι τα αποτελέσματα αποτελούνται από τις ίδιες περιγραφές. Όμως το γεγονός αυτό είναι λογικό καθώς η επερώτηση παρέμεινε η ίδια και άρα επιστράφηκαν όλες οι περιγραφές που απαντούσαν σε αυτή. Αντίθετα, με μια προσεκτικότερη ματιά στα αποτελέσματα είναι φανερό πως η σειρά τους και η βαθμολογία του καθενός έχει αλλάξει. Τα δύο στοιχεία στα οποία αξίζει να σταθεί κανείς για να επαληθεύσει την θεμιτή λειτουργία της μηχανής είναι:

- Το γεγονός πως η βαθμολογία της περιγραφής «WeatherForecastService.wsdl» έχει αυξηθεί σημαντικά σε σχέση με την προηγούμενη αναζήτηση. Η συγκεκριμένη υπηρεσία είναι η μόνη που περιέχει και τους δύο όρους της επερώτησης στον τίτλο της οπότε η αυξανόμενη βαρύτητα των όρων του τίτλου θα έπρεπε να επιφέρει ακριβώς αυτό το αποτέλεσμα.
- Η διαπίστωση πως οι υπηρεσίες που έχουν μετατοπιστεί προς το τέλος της λίστας των αποτελεσμάτων να μην περιέχουν έναν από τους όρους της αναζήτησης αλλά η συγκεκριμένη λέξη είναι συνδεδεμένη μαζί με κάποια άλλη με αποτέλεσμα η μηχανή να τις αντιμετωπίζει σαν ένα όρο και να μην αναγνωρίζει την ύπαρξη του όρου της επερώτησης στον τίτλο.

6.3.2. Σύνθετη αναζήτηση

Σε αντίθεση με την απλή αναζήτηση, η σύνθετη αναζήτηση απευθύνεται σε χρήστες που έχουν σχηματίσει μια εικόνα για την περιγραφή υπηρεσίας που αναζητούν, είτε όσον αφορά τις τεχνικές της λεπτομέρειες είτε την λεπτομερή λειτουργικότητα της. Η οθόνη της σύνθετης αναζήτησης φαίνεται στην Εικόνα 28.



Input

Output

WSDL ☒ OWL-S ☒

[Default Search](#) | [Advanced Search](#) | [About WeSS](#) | [Submit your own Web Service](#)

Εικόνα 28. Η οθόνη της σύνθετης αναζήτησης.

Οι χαρακτηριστικές διαφοροποιήσεις που παρουσιάζει συγκριτικά με την οθόνη της απλής αναζήτησης υποδηλώνουν με ευνόητο τρόπο την πλεονάζουσα λειτουργικότητα που προσφέρεται. Το πρώτο σημαντικό στοιχείο που τραβάει την προσοχή είναι η ύπαρξη δύο πεδίων εισόδου. Τα πεδία αυτά όπως αναφέρουν και οι ετικέτες τους, είναι εξειδικευμένα και αφορούν τις εισόδους και εξόδους της υπηρεσίας. Οι εισοδοί και οι έξοδοι αποτελούν πολύ σημαντικά στοιχεία κάθε συστήματος, ενώ στην περίπτωση των περιγραφών διαδικτυακών υπηρεσιών βοηθούν στον προσδιορισμό της υπηρεσίας με τον πλέον πρακτικό τρόπο.

Πρωτού παρατεθεί το παράδειγμα όπου φαίνεται η επίδραση του ειδικού βάρους ανάλογα με το πεδίο στο οποίο ο χρήστης έχει εισάγει όρους επερώτησης, πραγματοποιείται η ίδια αναζήτηση με το βάρος αυτό να αγνοείται. Τα αποτελέσματα ακολουθούν στην Εικόνα 29.



☒ WSDL
 ☒ OWL-S

1	http://services.nexus6studio.com/StockQuoteService.asmx?WSDL	StockQuoteService.wsdl	0.6406151028421908
2	http://www.webservicex.com/stockquote.asmx?WSDL	stockquote.wsdl	0.2435778785782428
3	http://www.gama-system.com/webservices/stockquotes.asmx?wsdl	stockquotes.wsdl	0.19349679327873007

[Default Search](#) |
 [Advanced Search](#) |
 [About WeSS](#) |
 [Submit your own Web Service](#)

Εικόνα 29. Αποτελέσματα για περιγραφές με είσοδο μια μετοχή χωρίς να λαμβάνονται υπόψη τα βάρη πεδίου.

Το παράδειγμα αφορά την αναζήτηση μιας περιγραφής διαδικτυακής υπηρεσίας που θα δέχεται σαν είσοδο τον όρο μετοχή (stock). Η επρώτηση αυτή αγνοώντας τα βάρη πραγματοποιήθηκε ώστε με την σύγκριση των αποτελεσμάτων να διαπιστωθεί αν τα βάρη τελικά διαδραματίζουν σημαντικό ρόλο ή όχι. Τα αποτελέσματα της ίδιας αναζήτησης με τα βάρη που αντιστοιχούν για το πεδίο εισόδου φαίνονται στην Εικόνα 30. Οι υπηρεσίες στις θέσεις 2 και 3 έχουν αλλάξει θέση. Το γεγονός αυτό επεξηγείται αν δούμε το περιεχόμενο της περιγραφής “stockquote.wsdl” που έχασε έδαφος στην Εικόνα 31.



☒ WSDL
 ☒ OWL-S

1	http://services.nexus6studio.com/StockQuoteService.asmx?WSDL	StockQuoteService.wsdl	0.9609226542632863
2	http://www.gama-system.com/webservices/stockquotes.asmx?wsdl	stockquotes.wsdl	0.2902451899180951
3	http://www.webservicex.com/stockquote.asmx?WSDL	stockquote.wsdl	0.2435778785782428

[Default Search](#) |
 [Advanced Search](#) |
 [About WeSS](#) |
 [Submit your own Web Service](#)

Εικόνα 30. Αποτελέσματα για περιγραφές με είσοδο μια μετοχή.

```

<middleDoc>

<name>stockquote.WSDL</name>

<URL>http://www.webservices.com/stockquote.asmx?WSDL</URL>

<documentation>Get Stock quote for a company Symbol</documentation>

<wordList>GetQuote, symbol, GetQuoteResponse, GetQuoteResult, string, GetQuoteSoa

<operations>
  <operation>
    <name>GetQuote</name>
    <documentation>Get Stock quote for a company Symbol</documentation>
    <IO>
      <input>true</input>
      <IOname>null</IOname>
      <IOmessage>GetQuoteHttpGetIn</IOmessage>
      <IOelement>null</IOelement>
    </IO>
    <IO>
      <input>false</input>
      <IOname>null</IOname>
      <IOmessage>GetQuoteHttpGetOut</IOmessage>
      <IOelement>null</IOelement>
    </IO>
  </operation>
</operations>

</middleDoc>

```

Εικόνα 31. Περιεχόμενα της χρήσιμης για αναζήτηση πληροφορίας της περιγραφής "stockquote.wsdl" όπως δομείται στο αντίστοιχο αρχείο .wef.

Η πτώση της κατάταξης της συγκεκριμένης περιγραφής δικαιολογείται από το γεγονός πως ο όρος της αναζήτησης "stock" δεν εμφανίζεται στην είσοδο της υπηρεσίας εν αντιθέση με το αντίστοιχο έγγραφο για την περιγραφή "stockquotes.wsdl" (Εικόνα 32).

```

<middleDoc>

    <name>stockquotes.wsdl</name>

    <URL>http://www.gama-system.com/webservices/stockquotes.asmx?wsdl</URL>

    <documentation>WebService provides stock exchange information.</documentation>

    <wordList>GetStocksXMLSchema,GetStocksXMLSchemaResponse,GetStockDailyValuesByDateXML

    <operations>
        <operation>
            <name>GetStocksXMLSchema</name>
            <documentation>Method returns XML schema (XSD) in which all XML formatted st
            <IO>
                <input>true</input>
                <IOname>null</IOname>
                <IOmessage>GetStocksXMLSchemaSoapIn</IOmessage>
                <IOelement>null</IOelement>
            </IO>
            <IO>
                <input>>false</input>
                <IOname>null</IOname>
                <IOmessage>GetStocksXMLSchemaSoapOut</IOmessage>
                <IOelement>null</IOelement>
            </IO>
        </operation>
    </operations>

```

Εικόνα 32. Περιεχόμενα της χρήσιμης για αναζήτηση πληροφορίας της περιγραφής "stockquotes.wsdl" όπως δομείται στο αντίστοιχο αρχείο .wev.

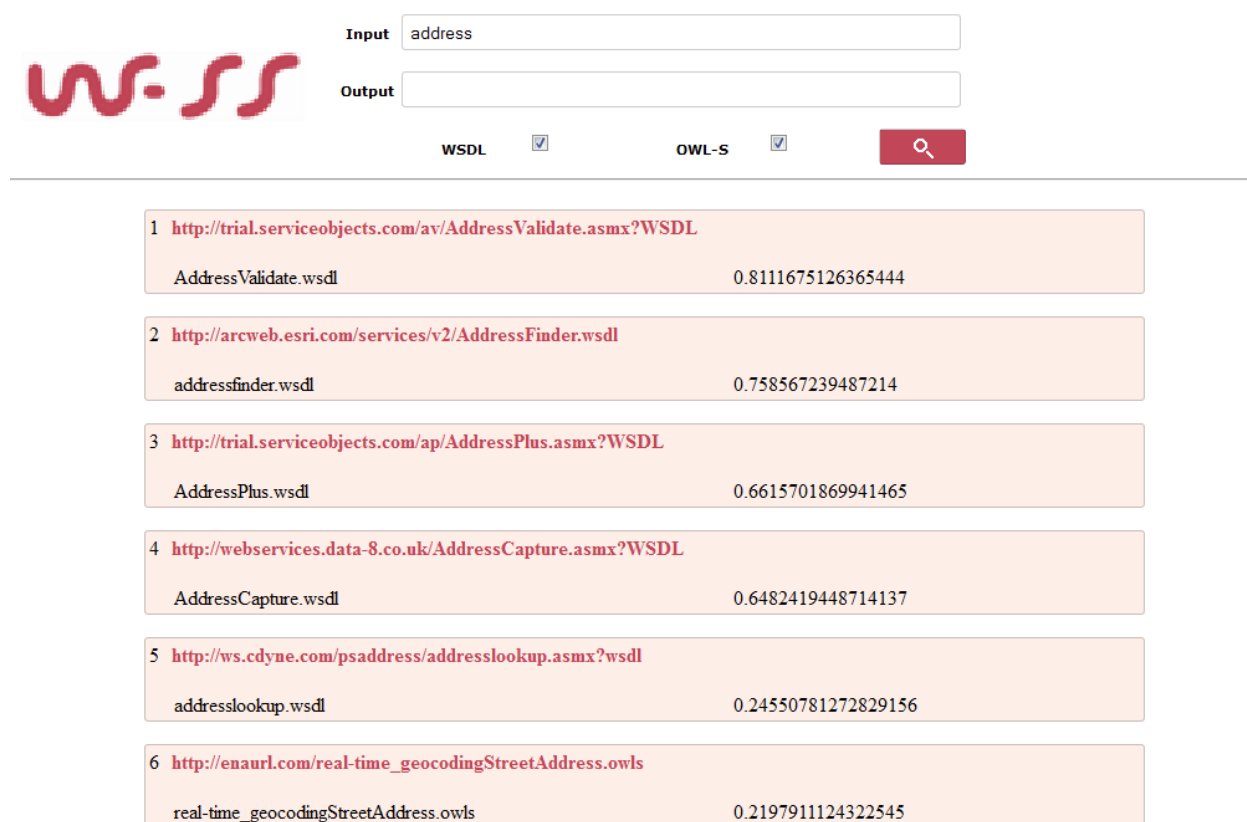
Σημειώνεται πως στα αποτελέσματα για λόγους πληρότητας περιλαμβάνονται περιγραφές που δεν περιέχουν τους όρους στα συγκεκριμένα στοιχεία με αρκετά χαμηλότερη όμως σειρά κατάταξης.

Η δεύτερη αλλαγή αφορά τα δύο προστιθέμενα κουμπιά επιλογής που αφορούν τα πρότυπα διαδικτυακών υπηρεσιών που παρέχονται. Η επιλογή τους σημαίνει την θεμιτή αναζήτηση υπηρεσιών των επιλεγμένων προτύπων ενώ η αντίθετη πράξη, η μη επιλογή τους, δηλώνει πως οι περιγραφές των μη επιλεγμένων προτύπων είναι επιθυμητό να αγνοηθούν.

Η λειτουργία αυτή παρουσιάζεται στο επόμενο παράδειγμα στο οποίο θα πραγματοποιηθούν δύο αναζητήσεις με την ίδια επερώτηση, μια με κάθε πρότυπο επιλεγμένο και μια με μόνο το ένα πρότυπο επιλεγμένο.

Έστω λοιπόν πως αναζητάται μια περιγραφή διαδικτυακής υπηρεσίας που θα λαμβάνει σαν είσοδο μια διεύθυνση. Η επερώτηση σε αυτή την περίπτωση διατυπώνεται με τον τρόπο που

φαίνεται στην Εικόνα 33 που ακολουθεί, ενώ φαίνονται και τα αποτελέσματα που επιστρέφονται:

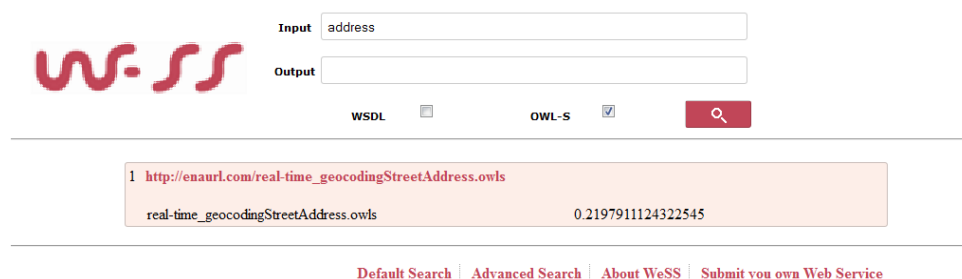


WSDL ☒ OWL-S ☒

1	http://trial.serviceobjects.com/av/AddressValidate.asmx?WSDL	AddressValidate.wsdl	0.8111675126365444
2	http://arcweb.esri.com/services/v2/AddressFinder.wsdl	addressfinder.wsdl	0.758567239487214
3	http://trial.serviceobjects.com/ap/AddressPlus.asmx?WSDL	AddressPlus.wsdl	0.6615701869941465
4	http://webservices.data-8.co.uk/AddressCapture.asmx?WSDL	AddressCapture.wsdl	0.6482419448714137
5	http://ws.cdyne.com/psaddress/addresslookup.asmx?wsdl	addresslookup.wsdl	0.24550781272829156
6	http://enauri.com/real-time_geocodingStreetAddress.owl	real-time_geocodingStreetAddress.owl	0.2197911124322545

Εικόνα 33. Αποτελέσματα αναζήτησης για περιγραφές κάθε τύπου με είσοδο μια διεύθυνση.

Παρατηρείται πως τα αποτελέσματα της παραπάνω εικόνας περιέχουν όπως ορίστηκε περιγραφές και των δύο υποστηριζόμενων προτύπων. Αν τώρα η ίδια επερώτηση επαναληφθεί με μόνο το ένα πρότυπο επιλεγμένο τα επιστρεφόμενα αποτελέσματα είναι τα κάτωθι (Εικόνα 34):



WSDL ☐ OWL-S ☒

1	http://enauri.com/real-time_geocodingStreetAddress.owl	real-time_geocodingStreetAddress.owl	0.2197911124322545
---	---	--------------------------------------	--------------------

[Default Search](#) | [Advanced Search](#) | [About WeSS](#) | [Submit your own Web Service](#)

Εικόνα 34. Αποτελέσματα αναζήτησης για περιγραφές OWL-S με είσοδο μια διεύθυνση.

Όπως ήταν αναμενόμενο περιέχουν μόνο την μια περιγραφή OWL-S η οποία βρισκόταν στην 6 θέση της προηγούμενης αναζήτησης επαληθεύοντας τον επιθυμητό στόχο.

ΚΕΦΑΛΑΙΟ 7

Συμπεράσματα και Μελλοντικές Κατευθύνσεις

Το παρόν Κεφάλαιο συνοψίζει την παρούσα πτυχιακή εργασία, εκθέτει τα συμπεράσματα που προέκυψαν κατά την εκπόνησή της και θέτει μελλοντικές κατευθύνσεις για την εξέλιξη της έρευνας στον εν λόγω τομέα.

7.1. Συμπεράσματα

Το πρότυπο των διαδικτυακών υπηρεσιών, με τα πλεονεκτήματα που το συνοδεύουν, μπορεί να αποτελέσει ένα πολύ ισχυρό εργαλείο για την κατασκευή διαδικτυακών εφαρμογών. Στόχος της παρούσας πτυχιακής εργασίας είναι να συμβάλλει στην βελτίωση της ανακάλυψης και χρήσης διαδικτυακών υπηρεσιών με την κατασκευή μιας μηχανής αναζήτησης εξειδικευμένης για περιγραφές διαδικτυακών υπηρεσιών.

Της υλοποίησης της μηχανής αναζήτησης προηγήθηκε εκτενής έρευνα στη βιβλιογραφία που αφορούσε τις διαθέσιμες επιλογές ανακάλυψης διαδικτυακών υπηρεσιών χρησιμοποιώντας μηχανές αναζήτησης. Οι μηχανές αυτές αφορούσαν είτε αντίστοιχες ερευνητικές προσπάθειες είτε συμβατικές μηχανές αναζήτησης γενικού σκοπού. Διαπιστώθηκε ότι οι τελευταίες παρουσιάζουν αδυναμίες στην επιτυχή ανάκληση και την ακρίβεια των επιθυμητών αποτελεσμάτων, ενώ οι προσπάθειες για μηχανές αναζήτησης στοχευμένες σε διαδικτυακές υπηρεσίες που έχουν εμφανιστεί μέχρι τη στιγμή εκπόνησης της παρούσας εργασίας κρίνονται ανεπαρκείς, αφού για τις περισσότερες δεν υπάρχουν διαθέσιμες υλοποιήσεις.

Εφόσον οι συνθήκες ευνοούσαν την δημιουργία μιας νέας μηχανής αναζήτησης για διαδικτυακές υπηρεσίες, το επόμενο βήμα ήταν η ενδελεχής εξέταση των επικρατέστερων προτύπων περιγραφής διαδικτυακών υπηρεσιών. Κατά την μελέτη αυτή προσδιορίστηκαν οι ιδιαιτερότητες των προτύπων περιγραφής WSDL και OWL-S καθώς και τα τμήματα εκείνα του καθενός που περιείχαν χρήσιμη πληροφορία για την αναζήτηση.

Η μηχανή αναζήτησης που αναπτύχθηκε στο πλαίσιο της παρούσας πτυχιακής εργασίας κρίθηκε σκόπιμο να έχει αρθρωτή αρχιτεκτονική, ώστε να είναι εύκολα τροποποιήσιμη και παραμετροποιήσιμη. Αποτελείται από τον web crawler, το ευρετήριο και τη γραφική διεπαφή που κατασκευάστηκαν εξ' ολοκλήρου στο πλαίσιο της παρούσας εργασίας. Όπως ήταν

αναμενόμενο, η υλοποίηση της μηχανής αναζήτησης ήταν μια αρκετά περίπλοκη και επίπονη διαδικασία, καθώς προέκυψαν θέματα που σχετίζονταν τόσο με την προσφερόμενη λειτουργικότητα όσο και με τα τεχνικά χαρακτηριστικά της, τα οποία αντιμετωπίστηκαν με επεκτάσεις και βελτιστοποιήσεις των διαφόρων τμημάτων της.

Ο web crawler λόγω της ίδιας της φύσης του αποδείχθηκε το πλέον χρονοβόρο και απαιτητικό τμήμα του συστήματος. Οι ιδιαίτερα αυξημένες του απαιτήσεις σε υπολογιστικούς πόρους και η πολυπλοκότητα της αποστολής του οδήγησαν σε μια σειρά βελτιστοποιήσεων ώστε να είναι δυνατή η εκτέλεσή του σε συμβατικά συστήματα. Επιπλέον, αναπτύχθηκαν και εφαρμόστηκαν ευριστικοί κανόνες οι οποίοι κατέστησαν την λειτουργία της συλλογής περιγραφών διαδικτυακών υπηρεσιών από τον Παγκόσμιο Ιστό αποδοτικότερη.

Σε ένα διάστημα 1-2 μηνών οπότε πλέον βρισκόταν στο τελικό του στάδιο, ο web crawler απέφερε 1500 περιγραφές διαδικτυακών υπηρεσιών. Τα αποτελέσματα αυτά θεωρούνται υποσχόμενα, ιδιαίτερα αν ληφθεί υπόψη το γεγονός πως πρόκειται για σελίδες οι οποίες ως επί το πλείστον αντιπροσωπεύουν την αντίστοιχη προσφερόμενη λειτουργικότητα, και όχι απλή πληροφορία, επομένως η δημιουργία και η διάθεσή τους είναι πιο χρονοβόρα και δύσκολη. Το 97% των περιγραφών που ανακαλύφθηκαν πρόκειται για περιγραφές WSDL, ενώ μόνο το υπόλοιπο 3% αντιστοιχούσε σε περιγραφές OWL-S. Η διαφορά αυτή στην εμφάνιση των δύο προτύπων οφείλεται σε δύο παράγοντες. Αφενός, το πρότυπο OWL-S είναι νέο και προς το παρόν όχι ιδιαίτερα διαδεδομένο. Αφετέρου, είναι δυνατή η παραγωγή αρχείων περιγραφής WSDL με αυτοματοποιημένο τρόπο χρησιμοποιώντας προγράμματα ανάπτυξης κώδικα, επομένως συχνότερα προτιμάται το πρότυπο WSDL. Ωστόσο, με την καθιέρωση της σημασιολογίας στον Παγκόσμιο Ιστό αναμένεται να υπάρξει σημαντική ώθηση στο πρότυπο OWL-S του οποίου μάλιστα η εξάρτηση αυτή από σημασιολογία και οντολογίες το καθιστούν αρκετά χρήσιμο για την αποτελεσματικότερη περιγραφή και κατ' επέκταση αναζήτηση υπηρεσιών.

Η μέθοδος ευρετηριοποίησης και αναζήτησης της χρήσιμης πληροφορίας των περιγραφών διαδικτυακών υπηρεσιών που χρησιμοποιήθηκε βασίστηκε στο ανεστραμμένο ευρετήριο σε συνδυασμό με το διανυσματικό μοντέλο (vector-space model) και τη μετρική term frequency – inverse document frequency (tf-idf). Ωστόσο, για την υποστήριξη επιπλέον λειτουργιών πέρα από την απλή αναζήτηση με την χρήση λέξεων κλειδιών κρίθηκε απαραίτητη η προσαρμογή του μοντέλου στην περίπτωση των διαδικτυακών υπηρεσιών με την χρήση βαρών, οι βέλτιστες τιμές των οποίων βρέθηκαν μέσω πειραμάτων.

Τα πειράματα που πραγματοποιήθηκαν στις περιγραφές που βρέθηκαν δεν έδειξαν κάποια ιδιαίτερη τάση ως προς τον τύπο περιγραφών που επιστρέφονταν στις επερωτήσεις του χρήστη. Ωστόσο, έγινε φανερή η ανάγκη διεξαγωγής επιπλέον πειραμάτων σε μεταγενέστερη χρονική στιγμή, όταν θα έχουν συγκεντρωθεί περισσότερες διαδικτυακές περιγραφές στο πρότυπο OWL-S.

Η γραφική διεπαφή που προσφέρεται στους χρήστες παρέχει τη δυνατότητα τόσο απλής αναζήτησης, μέσω λέξεων-κλειδιών, όσο και τη δυνατότητα πιο σύνθετων αναζητήσεων, που εκτός από συγκεκριμένες λέξεις κλειδιά, οι χρήστες, εκμεταλλεόμενοι τα προστιθέμενα βάρη στο μοντέλο tf-idf, μπορούν να υποδείξουν και το τμήμα της περιγραφής μιας διαδικτυακής υπηρεσίας στο οποίο επιθυμούν να βρίσκονται οι λέξεις (π.χ. είσοδο, έξοδο κτλ).

Η υλοποίηση ολοκληρώθηκε με πειράματα και περιπτώσεις χρήσης της μηχανής αναζήτησης που αναπτύχθηκε, ώστε να υποστηριχθεί η ορθή λειτουργία της και να επιδειχθεί η χρησιμότητά της και η αποτελεσματικότητά της σε σχέση με την απλή αναζήτηση μέσω μηχανών γενικού σκοπού.

Σημειώνεται πως η μεθοδολογία που χρησιμοποιήθηκε για τον σχεδιασμό και την υλοποίηση της μηχανής αναζήτησης ουσιαστικά αποτελεί μια μεθοδολογία στοχευμένης αναζήτησης και ανάκτησης πληροφορίας. Κατά τον ίδιο τρόπο που στο πλαίσιο της συγκεκριμένης πτυχιακής εργασίας αναζητούνται διαδικτυακές υπηρεσίες, είναι δυνατή η χρήση της για τη διερεύνηση και υλοποίηση μιας μηχανής αναζήτησης σε διαφορετικά πεδία με παρεμφερή χαρακτηριστικά και ημιδομημένο κείμενο, όπως για παράδειγμα οι οντολογίες. Οι αλλαγές που θα απαιτούσε κάτι τέτοιο θα αφορούσαν κυρίως τις ιδιαιτερότητες του συγκεκριμένου πεδίου, χωρίς να απαιτούν σημαντικές προγραμματιστικές τροποποιήσεις στον μηχανισμό crawling, ευρετηριοποίησης και αναζήτησης της παρούσας εργασίας.

Η μηχανή αναζήτησης που κατασκευάστηκε πρόκειται να εγκατασταθεί σε υποδομές του Τμήματος Πληροφορικής και Τηλεματικής του Χαροκοπέιου Πανεπιστημίου ώστε να είναι διαθέσιμη προς εξωτερικούς χρήστες τόσο μέσω της γραφικής της διεπαφής όσο και ως διαδικτυακή υπηρεσία.

7.2. Μελλοντικές Κατευθύνσεις

Η μηχανή αναζήτησης που κατασκευάστηκε στο πλαίσιο της συγκεκριμένης πτυχιακής εργασίας αποτελεί ένα ολοκληρωμένο σύστημα κατά τα πρότυπα που περιγράφονται στα Κεφάλαια του σχεδιασμού και της υλοποίησης. Το γεγονός όμως ότι ολοκληρώθηκε, δεν σηματοδοτεί το τέλος της ενασχόλησης με αυτή. Σκοπός εξ αρχής ήταν το σύστημα να σχεδιαστεί και να υλοποιηθεί με τρόπο που το σύνολο ή μέρος του θα ήταν δυνατό να επεκταθούν, παρέχοντας νέα λειτουργικότητα, είτε να επαναχρησιμοποιηθούν για διαφορετικούς σκοπούς, ερευνητικούς και μη.

Μια ενδιαφέρουσα εφαρμογή θα ήταν η διερεύνηση της αποδοτικότητας του web crawler και κατ' επέκταση της μηχανής αναζήτησης σε διαφορετικά πεδία στοχευμένης αναζήτησης. Η πιθανότητα το σύστημα να λειτουργεί με αποδοτικό τρόπο για την αναζήτηση διαφορετικών ειδών αρχείων που παρουσιάζουν συγκεκριμένα χαρακτηριστικά, όπως συνέβη με τις περιγραφές διαδικτυακών υπηρεσιών, και μπορούν με κάποιον τρόπο να προσφέρουν θα ήταν η επικρατέστερη επιλογή επαναχρησιμοποίησης του διαθέσιμου κώδικα, με τις απαραίτητες προσαρμογές.

Η χρησιμότητα της προτεινόμενης προσέγγισης γίνεται καλύτερα αντιληπτή αν εστιάσουμε στο έργο της μηχανής αναζήτησης. Η λειτουργία του web crawler είχε σαν αποτέλεσμα την δημιουργία μια σημαντικής σε μέγεθος συγκέντρωσης πραγματικών-χρηστικών περιγραφών διαδικτυακών υπηρεσιών. Οι περιγραφές αυτές θα μπορούσαν να γίνουν αντικείμενα πειραμάτων συσταδοποίησης για την εξόρυξη ενδιαφέρουσας πληροφορίας από αυτές. Η χρήση αλγορίθμων συσταδοποίησης (clustering) στα δεδομένα των περιγραφών θα είχε σαν αποτέλεσμα την δημιουργία συστάδων από σχετιζόμενες ή ισοδύναμες υπηρεσίες. Κάτι τέτοιο θα μπορούσε να φανεί χρήσιμο:

- για την ίδια την μηχανή αναζήτησης: η συγκεκριμένη τεχνική θα καθιστούσε εφικτή την πρόταση εναλλακτικών υπηρεσιών σε περιπτώσεις όπου για παράδειγμα μια υπηρεσία δεν είναι διαθέσιμη ή η χρήση της δεν είναι επιθυμητή από τον χρήστη λόγω κόστους κλπ.
- για τη διεξαγωγή μελέτης σύγκρισης των διαθέσιμων υπηρεσιών με βάση τα μη λειτουργικά τους χαρακτηριστικά, ώστε να βρεθούν κλάσεις ισοδύναμων ή στενά σχετιζόμενων υπηρεσιών.

Εναλλακτικά, θα μπορούσε να επεκταθεί ο ήδη υπάρχων κώδικας ώστε η δημιουργημένη μηχανή αναζήτησης να προσφέρει επιπλέον λειτουργικότητα. Παραδείγματα κοντινών επεκτάσεων που θα μπορούσε να επιδεχθεί το υπάρχον σύστημα είναι η παροχή δυνατότητας προοδευτικής τμηματικής ενημέρωσης του ευρετηρίου, η αυτοματοποίηση μέρους των διαδικασιών και η κλιμάκωση του συστήματος. Προς το παρόν ο μόνος τρόπος να ενημερωθεί το ευρετήριο είναι με την επαναδημιουργία του, ενώ διαδικασίες όπως η προσθήκη περιγραφών διαδικτυακών υπηρεσιών που μόλις έχουν ανακαλυφθεί στο ευρετήριο πραγματοποιούνται με μη αυτοματοποιημένο τρόπο.

Επιπλέον, η αρχιτεκτονική της μηχανή αναζήτησης προσφέρεται για την διεξαγωγή δοκιμών. Η δοκιμή διαφορετικών αλγορίθμων στην θέση του tf-idf για τον προσδιορισμό-υπολογισμό της συσχέτισης των εγγράφων με την επερώτηση ενδεχομένως να οδηγούσε σε καλύτερα αποτελέσματα. Γενικεύοντας, θα μπορούσαν να δοκιμαστούν διάφοροι εναλλακτικοί αλγόριθμοι λειτουργίας για καθένα ξεχωριστό υποσύστημα με σκοπό να προσδιοριστεί ο βέλτιστος συνδυασμός τους.

Η επέκταση της λειτουργικότητας της μηχανής αναζήτησης θα μπορούσε να επιτευχθεί χρησιμοποιώντας εξωτερικούς ως προς την μηχανή παράγοντες. Παράδειγμα τέτοιας, εξωτερικής, βοήθειας θα μπορούσε να είναι η χρήση θησαυρών ή οντολογιών. Η εκμετάλλευση των δεδομένων που προσφέρουν θα μπορούσε να βελτιώσει τα αποτελέσματα των αναζητήσεων, λαμβάνοντας υπόψη σημασιολογικές γενικεύσεις των όρων που υπάρχουν στο ευρετήριο. Σε περίπτωση που για παράδειγμα δεν υπήρχε κάποια υπηρεσία που να αντιστοιχούσε στην επερώτηση του χρήστη, η πραγματοποίηση σημασιολογικά ελαστικής (semantically relaxed) αναζήτησης ενδεχομένως να επέστρεφε κάποια υπηρεσία που ήταν παρεμφερής και εξυπηρετούσε τις ανάγκες του χρήστη αλλά δεν περιελάμβανε κάποιον από τους όρους της επερώτησης.

Στο ίδιο μοτίβο, σε περίπτωση που δεν υπήρχε κάποια υπηρεσία που να ικανοποιεί τις απαιτήσεις του χρήστη, θα ήταν αρκετά πρωτοποριακό και ωφέλιμο με την χρήση ευφυών τεχνικών όπως ο σχεδιασμός ενεργειών (planning) και οι κανόνες (rules), να εξετάζεται αν στις διαθέσιμες υπηρεσίες υπάρχουν κάποιες των οποίων η σύνθεση θα ικανοποιούσε το αίτημα του χρήστη.

Βιβλιογραφία

- [1] Alexa - Google.com, <http://www.alexa.com/siteinfo/google.com> [τελευταία πρόσβαση: 13/09/11]
- [2] Alonso, G., Casati, F., Kuno, H., Machiraju. V., *Web Services, Concepts, Architectures and Applications*, Springer-Verlag Berlin Heidelberg, 2004.
- [3] AltaVista, <http://www.altavista.com/> [τελευταία πρόσβαση: 21/09/11]
- [4] Aol, <http://search.aol.com/aol/webhome> [τελευταία πρόσβαση: 11/09/11]
- [5] Apache HttpClient, <http://hc.apache.org/httpclient-3.x/> [τελευταία πρόσβαση: 10/09/2011]
- [6] Ask, <http://www.ask.com/> [τελευταία πρόσβαση: 11/09/11]
- [7] Baeza-Yates, R., Ribeiro-Neto, B., *Modern Information Retrieval*. New York: ACM Press, Addison-Wesley, ISBN 0-201-39829-X, 1999.
- [8] Baidu, <http://www.baidu.com/> [τελευταία πρόσβαση: 10/09/11]
- [9] Berners-Lee, T., Cailliau, R., *WorldWideWeb: Proposal for a hypertexts Project*, 1990.
- [10] Berners-Lee, T., Hendler, J., Lassila, O., *The Semantic Web*, Scientific American, 284(5), pp. 34-43, 2001.
- [11] BFS, Breadth First Search, http://en.wikipedia.org/wiki/Breadth-first_search [τελευταία πρόσβαση: 12/10/11]
- [12] Bing, <http://www.bing.com/> [τελευταία πρόσβαση: 10/09/11]
- [13] Blekko, <http://blekko.com/> [τελευταία πρόσβαση: 10/09/11]
- [14] Booth, D., et al., *Web Services Architecture*, W3C Working Draft, <http://www.w3.org/TR/ws-arch/>, 2003.
- [15] Camel Case, http://en.wikipedia.org/wiki/CamelCase#cite_note-0 [τελευταία πρόσβαση: 17/09/11]
- [16] Castillo, C., *Effective Web Crawling*, University of Chile, 2004.
- [17] CSS, Cascading Style Sheets, <http://www.w3.org/Style/CSS/Overview.en.html> [τελευταία πρόσβαση: 10/09/2011]
- [18] Cutting, D., Pedersen, J., *Optimizations for dynamic inverted index maintenance*, Proceedings of SIGIR, 405-411, 1990.
- [19] Dan, G., *New ideas for Web service discovery-ontology-based prototype system of service search engine*, 2nd International Conference on Software Technology and Engineering, 2010.
- [20] Desktop Search Engine Market Share, <http://www.netmarketshare.com/search-engine-market-share.aspx?qprid=4&qpcustomd=0> [τελευταία πρόσβαση: 13/09/11]
- [21] Desktop Search Engine Market Share, <http://www.netmarketshare.com/search-engine-market-share.aspx?qprid=4&qpcustomd=0> [τελευταία πρόσβαση: 13/09/11]

- [22] Dong, X., Halevy, A., Madhavan, J., Nemes, E., Zhang, J., *Similarity search for web services*, Proceedings of VLDB, 2004.
- [23] Doulas, A., Cho, J., Olston, C., *What's new on the web?: the evolution of the web from a search engine perspective*, Proceedings of the 13th international conference on World Wide Web, ACM New York, NY, USA, 2004.
- [24] DuckDuckGo, <http://duckduckgo.com/> [τελευταία πρόσβαση: 10/09/11]
- [25] Edwards, J., McCurley, K. S., and Tomlin, J. A., *An adaptive model for optimizing performance of an incremental web crawler*, In Proceedings of the Tenth Conference on World Wide Web (Hong Kong: Elsevier Science), 2001.
- [26] Extensible Markup Language (XML) 1.0 (Fifth Edition), <http://www.w3.org/TR/xml/> [τελευταία πρόσβαση: 12/20/11]
- [27] FTP, File Transfer Protocol Specification, <http://tools.ietf.org/html/rfc959> [τελευταία πρόσβαση: 21/09/11]
- [28] Goetz et al, *Java Concurrency in Practice*, Addison Wesley Professional, Section 3.4. Immutability, 2006.
- [29] Google Alexa Rank, <http://www.alexa.com/siteinfo/google.com> [τελευταία πρόσβαση: 21/09/11]
- [30] Google, Corporate Information: Technology Overview, <http://www.google.com/about/corporate/company/tech.html> [τελευταία πρόσβαση: 10/09/11]
- [31] Google, <http://www.google.com/> [τελευταία πρόσβαση: 10/09/11]
- [32] Halavais, A., *Search Engine Society*, Polity Press, 2008, ISBN:0745642152 9780745642154, 2008.
- [33] Hatzi, O., *Σύνθεση Διαδικτυακών Υπηρεσιών μέσω Σχεδιασμού Ενεργειών*, Διδακτορική Διατριβή, Τμήμα Πληροφορικής, Χαροκόπειο Πανεπιστήμιο, 2009.
- [34] Hawking, D., *How things work: Web Search Engines (part 1)*, IEEE Computer, 2006.
- [35] Heritrix Web Crawler, <http://crawler.archive.org/> [τελευταία πρόσβαση: 11/09/11]
- [36] History of the Internet, <http://www.leidenuniv.nl/letteren/internethistory/?c=7> [τελευταία πρόσβαση: 12/10/11]
- [37] HTTP, HyperText Transfer Protocol 1.1, <http://tools.ietf.org/html/rfc2616> [τελευταία πρόσβαση: 12/10/11]
- [38] Inktomi, <http://en.wikipedia.org/wiki/Inktomi> [τελευταία πρόσβαση: 21/09/11]
- [39] Internet Media Type registration, consistency of use, <http://www.w3.org/2001/tag/2002/0129-mime> [τελευταία πρόσβαση: 10/09/2011]
- [40] Inverted Index, http://en.wikipedia.org/wiki/Inverted_index [τελευταία πρόσβαση: 10/09/11]
- [41] Jansena, B. J., Spin, A. *How are we searching the World Wide Web? A comparison of nine search engine transaction logs*. Information Processing & Management Volume 42, Issue 1, January 2006, Pages 248-263, 2006.
- [42] Java, <http://www.java.com/> [τελευταία πρόσβαση: 12/09/2011]

- [43] JavaScript, <https://developer.mozilla.org/en/JavaScript/Reference> [τελευταία πρόσβαση: 10/09/2011]
- [44] Lausen , H., Haselwanter, T. *Finding Web Services*, Proceedings of the 1st European Semantic Technology Conference, 2007.
- [45] Lycos, <http://www.lycos.com/> [τελευταία πρόσβαση: 10/09/11] [118]
- [46] M.F. Porter, *An algorithm for suffix stripping*, *Program*, 14(3) pp 130–137, 1980.
- [47] Ma, C., Song, M., Xu, K., Zhang, X., *Web Service discovery research and implementation based on semantic search engine*, 2nd Symposium on Web Society, 2010.
- [48] Makhoul, J., Kubala, F., Schwartz, R., Weischedel, R., In Proceedings of DARPA Broadcast News Workshop, Herndon, VA, 1999.
- [49] Microsoft Expression Web, version 4, http://www.microsoft.com/expression/products/web_overview.aspx [τελευταία πρόσβαση: 10/09/2011]
- [50] MSN Search, http://en.wikipedia.org/wiki/MSN_Search [τελευταία πρόσβαση: 21/09/11]
- [51] Netbeans, version 7.0, <http://netbeans.org/> [τελευταία πρόσβαση: 10/09/2011]
- [52] OWL, Web Ontology Language, <http://www.w3.org/TR/owl-ref/> [τελευταία πρόσβαση: 12/10/11]
- [53] OWL-S, Semantic Markup for Web Services, <http://www.daml.org/services/owl-s/1.1/> [τελευταία πρόσβαση: 12/10/11]
- [54] OWLS-TC version 2.2 revision 1, <http://projects.semwebcentral.org/projects/owls-tc/> [τελευταία πρόσβαση: 10/09/2011]
- [55] Package Java.net, <http://download.oracle.com/javase/1.4.2/docs/api/java/net/package-summary.html> [τελευταία πρόσβαση: 10/09/2011]
- [56] Paolucci, M., Kawmura, M., Payne, T., Sycara, K., *Semantic Matching of Web Services Capabilities*, First International Semantic Web Conference, 2002.
- [57] Piccinelli, G., *Service Provision and Composition in Virtual Business Communities*, 1999.
- [58] Planet Lab, www.planet-lab.org/, [τελευταία πρόσβαση: 11/09/11]
- [59] Platzner, C., Dustdar, S., *A Vector Space Search Engine for Web Services*, In Proceedings of the 3rd European IEEE Conference on Web Services, 2005.
- [60] Robots Exclusion Protocol, <http://www.robotstxt.org/robotstxt.html> [τελευταία πρόσβαση: 05/09/2011]
- [61] Rose, D. E., Levinson, D., *Understanding user goals in web search*. Proceedings of the 13th international conference on World Wide Web, ACM New York, NY, USA, 2004.
- [62] Salton, G., Buckley, C., Yu, C.T., *An evaluation of term dependence models in information retrieval*, In Research and Development in Information Retrieval (LNCS 146), 1982.
- [63] Search Engine Index, http://en.wikipedia.org/wiki/Index_%28search_engine%29 [τελευταία πρόσβαση: 10/09/11]

- [64] Seekda!, Web Services Search Engine, <http://webservices.seekda.com/> [τελευταία πρόσβαση: 10/09/11]
- [65] Semantic Annotations for WSDL, <http://www.w3.org/2002/ws/sawSDL/> [τελευταία πρόσβαση: 12/10/11]
- [66] Semantic Markup for Web Service, <http://www.w3.org/Submission/OWL-s> [τελευταία πρόσβαση: 12/20/11]
- [67] Sergey, B., Page, L., *The Anatomy of a Large-Scale Hypertextual Web Search*, 1998.
- [68] Shkapenyuk, V., Suel, T., *Design and implementation of a high performance distributed web crawler*, In Proceedings of the 18th International Conference on Data Engineering (ICDE), pages 357-368, San Jose, California, IEEE CS Press, 2002.
- [69] SOAP, Simple Object Access Protocol, <http://www.w3.org/TR/soap/> [τελευταία πρόσβαση: 12/10/11]
- [70] SOASPECS, <http://www.soaspecs.com/page3.asp> [τελευταία πρόσβαση: 12/10/11]
- [71] Spider Trap, http://en.wikipedia.org/wiki/Spider_trap [τελευταία πρόσβαση: 19/08/11]
- [72] Stop Words, http://en.wikipedia.org/wiki/Stop_words [τελευταία πρόσβαση: 17/09/11]
- [73] TF-IDF, http://en.wikipedia.org/wiki/TF_IDF [τελευταία πρόσβαση: 17/09/11]
- [74] The internet Gopher protocol, <http://tools.ietf.org/html/rfc1436> [τελευταία πρόσβαση: 21/09/11]
- [75] UDDI Specification version 3.0.2, http://www.uddi.org/pubs/uddi_v3.htm [τελευταία πρόσβαση: 12/10/11]
- [76] VisualVM, version 1.3.2, <http://visualvm.java.net/> [τελευταία πρόσβαση: 10/09/2011]
- [77] Web Crawling, http://en.wikipedia.org/wiki/Web_crawling [τελευταία πρόσβαση: 10/09/11]
- [78] Web Search for a Planet: The Google Cluster Architecture, labs.google.com/papers/googlecluster-ieee.pdf [τελευταία πρόσβαση: 12/10/11]
- [79] Web Service Description Language, http://en.wikipedia.org/wiki/Web_Services_Description_Language [τελευταία πρόσβαση: 12/20/11]
- [80] Web Service, http://en.wikipedia.org/wiki/Web_service [τελευταία πρόσβαση: 12/10/11]
- [81] Web Services Activity, <http://www.w3.org/2002/ws/> [τελευταία πρόσβαση: 12/10/11]
- [82] Web Services Description Language (WSDL) Version 2.0 Part 0: Primer, <http://www.w3.org/TR/wsdl20-primer/> [τελευταία πρόσβαση: 12/20/11]
- [83] WebCrawler, <http://www.webcrawler.com/> [τελευταία πρόσβαση: 21/09/11]
- [84] WSDL 2.0, Web Services Description Language, <http://www.w3.org/TR/wsdl20/> [τελευταία πρόσβαση: 12/20/11]
- [85] WSDL Version 2.0: RDF Mapping, <http://www.w3.org/TR/wsdl20-rdf/> [τελευταία πρόσβαση: 12/20/11]

- [86] WSDL, Web Service Definition Language, <http://www.w3.org/TR/wsdl> [τελευταία πρόσβαση: 12/20/11][q1]
- [87] WSDL-S, <http://www.w3.org/Submission/WSDL-S/> [τελευταία πρόσβαση: 12/10/11]
- [88] XML Schema Part 1: Structures Second Edition, <http://www.w3.org/TR/xml-infoset/> [τελευταία πρόσβαση: 12/20/11]
- [89] XSL Transformations Version 2.0, <http://www.w3.org/TR/xslt20/> [τελευταία πρόσβαση: 12/20/11]
- [90] Yahoo!, <http://search.yahoo.com/> [τελευταία πρόσβαση: 10/09/11]
- [91] Yandex, <http://www.yandex.com/> [τελευταία πρόσβαση: 10/09/11]
- [92] Zhang, Y., Zheng, Z., Lyu, M.R., *WSExpress: A QoS-aware Search Engine for Web Services*, International Conference on Web Services (ICWS), 2010.