ΧΡΗΣΤΟΣ Β. ΤΑΤΣΙΟΠΟΥΛΟΣ

ΑΝΙΧΝΕΥΣΗ ΚΑΙ ΕΞΟΡΥΞΗ ΟΙΚΟΝΟΜΙΚΩΝ ΚΑΙ ΕΜΠΟΡΙΚΩΝ ΠΛΗΡΟΦΟΡΙΩΝ, ΑΠΟ ΤΑ ΔΕΔΟΜΕΝΑ ΔΙΑΔΙΚΤΥΟΥ ΜΕ ΧΡΗΣΗ "ΕΥΦΥΩΝ ΜΕΣΩΝ" (INTELLIGENT AGENTS), ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΑΥΤΩΝ ΤΩΝ ΠΛΗΡΟΦΟΡΙΩΝ ΣΤΟΥΣ ΧΡΗΣΤΕΣ ΜΕ ΧΡΗΣΗ ΚΙΝΗΤΩΝ ΣΥΣΚΕΥΩΝ

Επιβλέπων Καθηγητής: Β. Βουτσινάς

ΔΙΔΑΚΤΟΡΙΚΗ ΔΙΑΤΡΙΒΗ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΑΣ

ΤΜΗΜΑ ΔΙΟΙΚΗΣΗΣ ΕΠΙΧΕΙΡΗΣΕΩΝ

ΔΕΚΕΜΒΡΙΟΣ 2009

Περίληψη

Η παρούσα Διδακτορική Διατριβή (Δ.Δ.) διαπραγματεύεται τα σύγχρονα και πολύπλοκα προβλήματα της ανταλλαγής δεδομένων και της ανάκτησης γνώσης, τα οποία προκύπτουν από την χρήση των σύγχρονων συστημάτων και μέσων, ιδιαίτερα αυτών τα οποία είναι προσβάσιμα στα πλαίσια των τεχνολογιών του Διαδικτύου. Η πρόσβαση στην πληροφορία στην εποχή μας έχει διευκολυνθεί όσο ποτέ άλλοτε και αναπάντεχα έχει δημιουργήσει την ανάγκη της οργανωμένης πρόσβασης στο περιεχόμενο, με κίνδυνο αχρήστευσής της σε αντίθετη περίπτωση. Η πληροφορία βρίσκεται πλέον παντού και είναι διαθέσιμη σχεδόν άμεσα, μετατοπίζοντας το βάρος χρήσης των υπαρχόντων συστημάτων στον κατάλληλο τρόπο χαρακτηρισμό του «περιεχομένου» με «μεταπληροφορία» (metadata) σε οργανωμένες δομές αναπαράστασης γνώσης. Ακολούθως η χρησιμότητα και η πρόσθετη αξία αυξάνει με την χρήση περιβαλλόντων και μέσων τα οποία προωθούν την διανομή της σύμφωνα με ένα καλώς ορισμένο μοντέλο παραγωγής – ζήτησης, στους κατάλληλους προς αυτό καταναλωτές – χρήστες αυτής.

Στα πλαίσια της παρούσης Δ.Δ, εξετάζονται σύγχρονες μεθοδολογίες αναπαράστασης και διαχείρισης γνώσης όπως οι οντολογίες. Επίσης, εξετάζονται σύγχρονες μεθοδολογίες ανακάλυψης και διανομής γνώσης και περιεχομένου όπως οι ευφυείς διαμεσολαβητές (intelligent agents). Μαζί με τα ανωτέρω, εξετάζονται στα πλαίσια της παρούσης Διδακτορικής Διατριβής αλγόριθμοι ανάκτησης γνώσης από πολύ μεγάλους όγκους δεδομένων, αναπαριστάμενων υπό την μορφή οντολογιών και δενδρικών δομών υπό την έννοια ταξινομημένων γράφων, βάσει πιο κλασσικών μεθοδολογιών και μοντέλων που αφορούν και αναπτύχθηκαν κυρίως για επιχειρηματικά δεδομένα, όπως είναι οι αλγόριθμοι ταξινόμησης, ομαδοποίησης και συσχέτισης.

Η παρούσα Δ.Δ. εξετάζει επίσης με βάση τις ανωτέρω μεθοδολογίες την «ομοιότητα» (similarity) κατανόησης εννοιών σε κάποιο ορισμένο πεδίο γνώσης από συγκεκριμένους χρήστες του πεδίου αυτού. Στη συνέχεια, προτείνει μία μεθοδολογία αυτόματης ανταλλαγής δομημένου περιεχομένου και εννοιών μεταξύ αυτών, ανάλογα των κοινών ή μη ενδιαφερόντων τους. Το σύστημα αναπτύσσεται σε θεωρητικό επίπεδο στην μεθοδολογία ONARM και υλοποιείται σε επίπεδο λογισμικού με βάση την προτεινόμενη μεθοδολογία στο σύστημα Concept Net.

Τέλος, θα πρέπει να σημειωθεί ότι η μεθοδολογία που προτείνεται, είναι γενική και ανεξάρτητη από την κωδικοποίηση συγκεκριμένων εννοιών σε συγκεκριμένο γνωσιακό χώρο (knowledge domain), καθιστώντας την κατ' αυτόν τον τρόπο άμεσα εφαρμόσιμη σε πολλούς τομείς οι οποίοι ενέχουν user interaction σε πολύπλοκα εννοιολογικά πεδία, όπου ο κίνδυνος μη διανομής κατάλληλου περιεχομένου αυξάνει δραματικά σε σχέση με την πολυπλοκότητα του χώρου αυτού κατά την διαδικασία μεταφοράς της γνώσης.

Πιο συγκεκριμένα, στα πλαίσια της παρούσης Δ.Δ., η μεθοδολογία ONARM και η εφαρμογή CONCEPT-NET εφαρμόστηκαν στο γνωσιακό χώρο του Τουρισμού, όπου και υλοποιήθηκε εφαρμογή για την έξυπνη διαχείριση εννοιών και περιεχομένου γενικότερα που αφορούν υπηρεσίες τουριστικού ενδιαφέροντος, με βάση αντίστοιχες προσωποποιημένες (personal) και γενικές οντολογίες αναφοράς (reference ontologies).

Ευχαριστίες

Κοιτάζοντας προς τα πίσω, όταν ξεκίναγα την προσπάθεια αυτή, μπορώ να την παρομοιάσω σαν ένα ταξίδι μακρύ, γεμάτο εμπειρίες, που κάλυψε μεγάλο και σημαντικό μέρος της ζωής μου. Ήταν μακρύ το ταξίδι αυτό και δεν μπορούσα ποτέ να φανταστώ τις δυσκολίες του. Θα ξεκίναγε από το λιμάνι της περιορισμένης γνώσης μου και θα πήγαινε στα άγνωστα, βαθιά νερά για μένα, προκειμένου να φτάσω στην Ιθάκη μου. Ήθελε προετοιμασία καλή. Ήθελε και υποστήριξη επίσης. Και όραμα. Και σκοπό. Και θα πέρναγε από πολλές συμπληγάδες, σειρήνες και απογοητεύσεις.

Κάπως έτσι ξεκίνησε το περιπετειώδες ταξίδι του διδακτορικού μου, της αναζήτησης μου για Γνώση, με εμένα στο "τιμόνι". Αλλά αλήθεια τι μπορεί να κάνει ένας μόνος τιμονιέρης, χωρίς το κατάλληλο πλήρωμα σε ένα μακρύ και δύσκολο ταξίδι;

Και θα ήταν και μεγάλο και δύσκολο, με πολλές απογοητεύσεις το ταξίδι αυτό. Δεν θα μπορούσε άλλωστε να είναι και αλλιώς προκειμένου να κατακτηθεί ο "απαγορευμένος" αλλά τόσο ποθητός καρπός της Γνώσης.

Και σε αυτό, θα ήταν πολύτιμο κάθε φως που θα όριζε την πορεία μου, από το πιο αμυδρό έως το πιο δυνατό, από κάθε φάρο στο διάβα μου, φέρνοντάς με ως το τελικό λιμάνι, τον προορισμό μου. Για να αποφύγω τους υφάλους και τις ξέρες... για να φτάσω στον προορισμό μου και να μην χάσω την πορεία μου... τον στόχο μου...

Συνέβαλλαν πολλοί στο ταξίδι μου αυτό... και με βοήθησαν άλλοι τόσοι... Γι' αυτό και θα ήθελα να τους ευχαριστήσω όλους εκ βάθους καρδίας, και θα ήθελα εκ των προτέρων να ζητήσω συγνώμη αν παραλείπω κάποιους. Αυτό δεν έγινε σκόπιμα.

Πρώτα απ' όλα, θα ήθελα να ευχαριστήσω την οικογένειά μου που με στήριξε τόσο... που με ανέχτηκε όλα αυτά τα χρόνια... τους ευχαριστώ γι' αυτό: για τον χρόνο που τους στέρησα και τον αφιέρωσα στο προσωπικό ταξίδι μου... τη σύζυγό μου Τέτα και τις κόρες μου Φωτεινούλα και Ελευθερία. Σας ευχαριστώ πάρα πολύ. Μετά την αδελφή μου και την μητέρα μου που με στήριξαν με όλες τους τις δυνάμεις... Άννα και μαμά σας ευχαριστώ.

Το ταξίδι μου όμως, δεν θα ήταν δυνατόν να πραγματοποιηθεί χωρίς την καθοδήγηση του καθηγητή μου κ. Β. Βουτσινά. Με βοήθησε πολλές φορές και μου έστεργε σαν φάρος, έτσι ώστε ούτε να χάσω, ούτε κυρίως να εγκαταλείψω την πορεία προς τον προορισμό μου. Και

στην διάρκεια αυτών των ετών, υπήρξαν πολλές τέτοιες «Σειρήνες» στο ταξίδι μου για αυτό. Σε ευχαριστώ γι' αυτό αξιότιμε κε Βασίλη Βουτσινά, αγαπητέ μου Βασίλη, εκ βάθους καρδίας, τόσο για την ακαδημαϊκή όσο και για την ανθρώπινη καθοδήγηση και υποστήριξή σου στο ταξίδι μου αυτό.

Θα ήθελα επίσης να ευχαριστήσω την τριμελή επιτροπή για την πολύτιμη συμβολή της στο να προκύψει από την εργασία μου, μια όσο το δυνατόν πιο άρτια επιστημονική διατριβή.

Επίσης ευχαριστώ, τον συνεργάτη μου Κώστα Σιδηρόπουλο που όλα αυτά τα χρόνια τραβήξαμε μαζί "κουπί", καθώς η βοήθειά του ήταν πολύτιμη στο πολύπλοκο κομμάτι του κώδικα της παρούσας Δ.Δ. Χίλια ευχαριστώ Κώστα.

Πρωτίστως όμως, η εργασία αυτή είναι αφιερωμένη στη μνήμη του πατέρα μου Βασίλη, που πρόωρα και αναπάντεχα ταξίδεψε μακριά μας, πέρσι τέτοιο περίπου καιρό, ο οποίος σε όλα τα χρόνια της ζωής μου, μου εμφύσησε και μου προσέφερε το όραμα της παιδείας, της εκπαίδευσης και της ψυχικής καλλιέργειας, σαν πρωτεύοντες στόχους της ζωή μου, ενθαρρύνοντάς με σε κάθε γνωσιακό μου άλμα στη πορεία μου και για τον οποίο η επίτευξη του παρόντος στόχου υπήρξε στόχος πολύ υψηλής προτεραιότητος και αξίας... Σε ευχαριστώ μπαμπά... Το λιμάνι του προορισμού μου ήρθε σίγουρα κοντύτερα με την συμβολή και την στήριξή σου.

Περιεχόμενα

Περίληψη 4

Ευχαριστίες 7

Περιεχόμενα 9

Εισαγωγή 13

Καινοτομία στα πλαίσια της παρούσης Διατριβής 16

Συνεισφορά της παρούσης σε τεχνολογικό υπόβαθρο 19

Οργάνωση Διατριβής 20

Κεφάλαιο 1. Θεωρητικό Υπόβαθρο του Προβλήματος 23

1.1. ΕΙΣΑΓΩΓΗ 23

1.2. Ο ΣΗΜΑΣΙΟΛΟΓΙΚΟΣ ΙΣΤΟΣ 27

1.2.1. Ορισμος 27

1.2.2. Δομες Περιγραφης Περιεχομενου Στο Σημασιολογικο Ιστο 29

1.3. ΟΝΤΟΛΟΓΙΕΣ - ΓΕΝΙΚΑ 35

1.3.1. Ορισμος 36

1.3.2. Περιβαλλοντα Απεικονισης Οντολογιων 38

1.3.2.1. PROTÉGÉ 38

1.3.2.2. KAON Reverse 42

1.3.3. Γλωσσες Περιγραφης Οντολογιων 43

1.3.3.1. RDF – RDF(S) 44

1.3.3.2. DAML + OIL 46

1.3.3.3. OWL 48

1.3.4. Γλωσσες Αναζητησης Οντολογιων 49

1.3.4.1. SPARQL 49

1.3.4.2. JENA 60

1.4. ΣΥΓΚΡΙΣΗ – ΔΙΑΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ ΟΝΤΟΛΟΓΙΩΝ ΚΑΙ Β.Δ. 71

1.4.1. Διαλειτουργικοτητα 75

1.4.2. Σημασιολογικη Διαφοροποιηση 76
1.4.3. Προτυποποιηση & Διαφοροποιηση 77
1.5. ΜΕΘΟΔΟΙ ΕΞΟΡΥΞΗΣ ΓΝΩΣΗΣ 78
1.5.1. Εξυπνοι διαμεσολαβητες 80
1.5.1.1. Εισαγωγή 80
1.5.1.2. Ολοκλήρωση Δεδομένων 81
1.5.1.3. Έξυπνοι Διαμεσολαβητές Λογισμικού 82
1.5.1.4. Αρχιτεκτονική 82
1.5.1.5. Συμπεράσματα 84
1.5.2. Κανονες Συσχετισης - Association Rules 85
1.5.2.1. Παράδειγμα Χρήσης Αλγορίθμου APRIORI 86
1.5.2.2. Apriori Αλγοριθμος 87
1.5.2.3. Παραδειγμα Εφαρμογης Αλγοριθμου APRIORI 90
1.5.3. Κανονες Ταξινομησης 91
1.5.4. Κανονες Ομαδοποιησης – Clustering Rules 92
1.5.5. Έρευνες Σε Γραφους Δενδρικης Μορφης 94
1.5.5.1. Εισαγωγή στις Δενδρικές Δομές 94
1.5.5.2. Breadth First Search (BFS) 100
1.5.5.3. Depth First Search (DFS) 101
1.5.5.4. Αποτελεσματα BFS - DFS 101
1.5.5.5. Σύγκριση BFS & DFS Αλγορίθμων 102
1.5.5.6. Γραφήματα Δενδρικών Δομών και Κανόνες Συσχέτισης 104
Κεφάλαιο 2. Θεωρητικό Υπόβαθρο Προτεινόμενης Μεθοδολογίας 108
2.1. ΑΝΤΙΣΤΟΙΧΗΣΗ ΟΝΤΟΛΟΓΙΩΝ 109
2.1.1. Μορφισμοι 110
2.1.2. Εμπλουτισμενες Οντολογιες 110
2.1.3. Αντιστοιχηση, Ευθυγραμμιση, Επικοινωνια και Ενοποιηση Οντολογιων 111

2.2. ΠΡΟΗΓΟΥΜΕΝΕΣ ΠΡΟΣΕΓΓΙΣΕΙΣ ΑΝΤΙΣΤΟΙΧΗΣΗΣ ΟΝΤΟΛΟΓΙΩΝ 111
2.3. ΑΞΙΟΛΟΓΗΣΗ - ΑΠΟΔΟΤΙΚΟΤΗΤΑ ΣΤΗΝ ΑΝΤΙΣΤΟΙΧΗΣΗ ΟΝΤΟΛΟΓΙΩΝ 117
Κεφάλαιο 3. Προτεινόμενη Μεθοδολογία 121
3.1. ΕΡΕΥΝΗΤΙΚΑ ΘΕΩΡΗΤΙΚΑ ΘΕΜΑΤΑ 121
3.1.1. Γενικη Παρουσιαση 121
3.1.2. Η προτεινομενη τεχνικη ONARM 124
3.1.2.1. Γενικά 124
3.1.2.2. Αναλυτική Παρουσίαση της Προτεινόμενης Μεθοδολογίας 126
3.1.3. Αξιολογηση - Αποδοτικοτητα μεθοδολογιας ONARM 132
3.1.4. Ευθυγραμμιση ενδιαφερόντων μερών Οντολογιων 132
3.1.4.1. Το πρόβλημα της Ευθυγράμμισης Οντολογιών 133
3.1.4.2. Σχετιζομενες Τεχνικές Αντιστοίχησης - Ευθυγράμμισης 134
3.1.4.3. Προτεινόμενα Μέτρα Ενδιαφέροντος στη Ευθυγράμμιση Οντολογιών 137
3.2. ΠΕΙΡΑΜΑΤΙΚΑ ΔΕΔΟΜΕΝΑ & ΑΠΟΤΕΛΕΣΜΑΤΑ 140
3.2.1. Αναγνωριση Σχετιζομενων Κομβων 140
3.2.2. αυτοματη συσχετιση σε σχεση με ανθρωπινη αξιολογηση 144
3.2.3. Συμπεριφορα Βαρων Σε Διαφορετικου Τυπου Οντολογιες 147
3.3. ΣΥΜΠΕΡΑΣΜΑΤΑ 155
Κεφάλαιο 4. Μελέτη Εφαρμογής της Προτεινόμενης Μεθοδολογίας στον Τουρισμό 156
4.1. ΠΑΡΟΥΣΙΑΣΗ ΕΦΑΡΜΟΓΗΣ 156
4.1.1. Σκοπος - Γενικη περιγραφη 156
4.1.2. Σεναρια Χρησης - UML Διαγραμματα 157
4.1.2.1. Σενάριο 1: USER - USER 158
4.1.2.2. Σενάριο 2: USER - SYSTEM 160
4.1.3. Περιβαλλον Υλοποιησης 162
4.1.4. Γενικη Αρχιτεκτονικη Συστηματος 163
4.1.4.1. Εισαγωγή 163
4.1.4.2. Αρχιτεκτονικη Backend 165

4.1.4.3. Αρχιτεκτονική Backend Με Βάση CLASS – UML Διαγράμματα 167
4.1.4.4. Αρχιτεκτονικη FrontEnd 178
4.1.4.5. Αρχιτεκτονική Frontend Με Βάση CLASS – UML Διαγράμματα 180
4.1.5. USER INTERFACES - SCREENSHOTS..... 182
4.1.6. ΈΛΕΓΧΟΣ ΟΡΘΗΣ ΛΕΙΤΟΥΡΓΙΑΣ ΣΥΣΤΗΜΑΤΟΣ (EVALUATION) 189
4.2. ΠΑΡΟΥΣΙΑΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ & ΔΙΑΔΙΚΑΣΙΩΝ ΕΦΑΡΜΟΓΗΣ 192
Κεφάλαιο 5. Διερεύνηση Πεδίων Εφαρμογής.....196
5.1. ΠΡΟΣΩΠΟΠΟΙΗΜΕΝΕΣ ΔΙΑΔΙΚΑΣΙΕΣ ΜΑΘΗΣΗΣ.....196
5.2. ΣΤΟΧΕΥΜΕΝΗ ΔΙΑΦΗΜΙΣΗ 197
Κεφάλαιο 6. Συμπεράσματα & Προτάσεις για Μελλοντική Έρευνα 198
ΓΛΩΣΣΑΡΙ 202
ΒΙΒΛΙΟΓΡΑΦΙΑ 204

Εισαγωγή

Στις μέρες μας, όπου κυρίαρχο στοιχείο στην τεχνολογία αποτελεί το Διαδίκτυο, έχουμε βρεθεί αντιμέτωποι με τεράστιο όγκο καταγεγραμμένης πληροφορίας, σε διάφορες δομές καταγραφής, από διάφορα πεδία, κατανεμημένης σε διάφορα γεωγραφικά σημεία. Έτσι τις πιο πολλές φορές, ερωτήματα που τίθενται έχουν ήδη απαντηθεί, με παραπάνω από ένα τρόπους, οι οποίοι όμως παραμένουν άγνωστοι και μη προσβάσιμοι στον τελικό χρήστη του διαδικτύου. Επιπρόσθετα, η ανακάλυψη αυτής της ουδετεροποιημένης πληροφορίας, επαφίεται στην δημιουργικότητα – φαντασία του χρήστη να την εξεύρει και να την χρησιμοποιήσει με ένα εύχρηστο τρόπο. Αυτά όμως μοιάζουν στην σύγχρονη εποχή παρωχημένα μοντέλα πρόσβασης στην πληροφορία. Καλούμεθα λοιπόν να συνεισφέρουμε – και αυτό αποτελεί αντικείμενο της παρούσης – μεθοδολογίες, προϊόντα λογισμικού και υπηρεσίες, οι οποίες τείνουν να χρησιμοποιούν την επιμέρους πληροφορία, να την γενικεύουν, να της προσδίδουν αφαιρετικές ιδιότητες και να την μετατρέπουν εν γένει σε επαναχρησιμοποιούμενη γνώση κατ' αυτόν τον τρόπο. Αφού αυτό επιτευχθεί, ο επόμενος στόχος – ο οποίος και αυτός είναι στα πλαίσια της παρούσης – είναι να προτείνει μεθοδολογίες για την ευφυή ανακάλυψη της γνώσης αυτής, η οποία μπορεί ή όχι να βρίσκεται στο ίδιο γεωγραφικό σημείο, να έχει ή μη παραχθεί από τον ίδιο ή όμοιους χρήστες ή / και να χρησιμοποιεί τις ίδιες περιγραφικές δομές για την γνωστοποίησή της. Τέλος, εξίσου σημαντικό, μετά τα πρώτα δυο στάδια, είναι η πρόταση μεθοδολογιών για την δυναμική και με ευφυή τρόπο διανομή της γνώσης αυτής, πεδίο στο οποίο η παρούσα Δ.Δ. συνεισφέρει επίσης.

Το πρόβλημα το οποίο η παρούσα Δ.Δ. αντιμετωπίζει, ορίζεται από τις παρακάτω συνοπτικά συνιστώσες, στις οποίες προσπαθεί να συνεισφέρει:

- Ύπαρξη πολύ σημαντικού προβλήματος στη διαχείριση γνώσης
- Ύπαρξη πολύ μεγάλου πλήθους αποθηκευμένης πληροφορίας
- Ύπαρξη πολύ σημαντικού προβλήματος στη συσχέτιση και ομογενοποίησης της καταγεγραμμένης πληροφορίας – γνώσης, της οποίας ο ρυθμός παραγωγής αυξάνεται καθημερινά, γεωμετρικά.

- Ύπαρξη Παραδόξου: Όσο περισσότερη πληροφορία είναι διαθέσιμη για ένα αντικείμενο, τόσο πιο δύσκολο είναι να βρεθεί αξιόλογη πληροφορία γύρω από αυτό. Χαρακτηριστικό παράδειγμα οι τρέχουσες μηχανές αναζήτησης και η προβληματική που αναπτύσσεται γύρω από την αποδοτικότητα τους.

Ειδικότερα, σχετικά με τις ευρέως γνωστές μηχανές αναζήτησης, παρουσιάζουν αυτές ένα εγγενές μειονέκτημα: Βασίζουν την λειτουργία του σε στατιστική ανάλυση και δεν λαμβάνουν υπόψιν τους δομικά και γλωσσικά στοιχεία των αναφερομένων εννοιών. Επιπρόσθετα, κατά την επιστροφή των πάρα πολλών αποτελεσμάτων από την αναζήτηση, δεν γίνεται εννοιολογική συσχέτιση αυτών, παρά μόνο απλή παράθεσή τους. Έτσι, προήλθε η ανάγκη εξεύρεσης τέτοιων σχέσεων – αλληλοσυσχετίσεων, η οποία επιτυγχάνεται με την εισαγωγή των μεταδεδομένων, και την διασύνδεση αυτών στο σημασιολογικό ιστό, προσδίδοντας πρόσθετη αξία στο διαδίκτυο και στην άναρχη δομή που αυτό αναπαριστά, σχετικά με την αναπαράσταση γνώσης.

Η παρούσα Δ.Δ. θεωρεί ένα πιο γενικευμένο σενάριο, το ότι δηλαδή, η γνώση αποτελεί ένα ακόμα «καταναλωτικό αγαθό» ή «υπηρεσία», όπου έχουμε τους γνωστούς όρους, παραγωγής και κατανάλωσης, ακόμα δε και της συσχέτισης αυτής από διαφορετικά γνωσιακά πεδία. Έτσι, δημιουργούνται επιχειρηματικά μοντέλα και σχέδια οικονομικής αξιοποίησης γνώσης, όπως για παράδειγμα δυναμικά Portal τουρισμού, όπου ανάλογα με το προφίλ του καθενός, προτείνονται ανάλογες υπηρεσίες από διαφορετικές πηγές διάθεσης, επενεργώντας ουσιαστικά σαν ένας οικονομικός ενδιάμεσος (business agent / broker), μεταξύ του παραγωγού και του καταναλωτή (χρήστη).

Τέλος η παρούσα Δ.Δ., θεωρώντας ότι, κινούμαστε καθημερινά σε ένα «πολυκατάστημα» εννοιών, όπου ο κάθε «πελάτης» έχει προσωποποιημένη λογική κατανάλωσης και κατ' αυτόν τον τρόπο πραγματοποιεί τις επιλογές του, πλησιάζει το θέμα της απόκτησης γνώσης μέσω των ακόλουθων ερωτημάτων, στα οποία δίνει απαντήσεις:

- Μπορούμε να προβλέψουμε τις ανάγκες του πελάτη (σε γνώση);
- Μπορούμε να προβλέψουμε τον τρόπο με τις οποίες θα καλύψει αυτές τις ανάγκες (διανομή γνώσης);
- Μπορούμε να του προσφέρουμε, απ' ευθείας, τα προϊόντα που εμείς (σαν σύστημα) θεωρούμε ότι καλύπτουν τις ανάγκες του (μέσον διανομής γνώσης);

Τα κέρδη από την εξεύρεση απαντήσεων και εφαρμογή των αυτών είναι φανερά.

Καινοτομία στα πλαίσια της παρούσης Διατριβής

Η παρούσα Δ.Δ. αντιμετωπίζει το πρόβλημα της ανίχνευσης, ανάκτησης και διανομής περιεχομένου και γνώσης με τρόπο ο οποίος μέχρι τώρα δεν έχει αντιμετωπιστεί από την τρέχουσα βιβλιογραφία και ερευνητική διαδικασία.

Πιο συγκεκριμένα, θεωρούμε ότι ο κόσμος στον οποίο ζούμε, περιγράφεται από έννοιες (concepts), τις οποίες εμείς σαν ανθρώπινα όντα αλλά ταυτόχρονα και σαν «χρήστες» του κόσμου μας, προσπαθούμε κάθε στιγμή να κατανοήσουμε, δηλαδή να τους δώσουμε, υπάρχουσα ή νέα, τη σωστή έννοια ανάμεσα από πολλές υποψήφιες, ανάλογα με το περιεχόμενο (context) μέσα στο οποίο συναντώνται.

Ακολούθως και γύρω από την «ορθή» κατανόηση των εννοιών αυτών, δημιουργούμε αυτό για το οποίο το ανθρώπινο όν είναι δημιουργημένο, την κοινωνικότητα, με βάση κοινά γνωρίσματα και ενδιαφέροντα μεταξύ των οντοτήτων μίας ή περισσοτέρων κοινωνικών ομάδων, αυτά που στις μέρες μας ονομάζονται κοινωνική δικτύωση (social networking) και ηλεκτρονικές κοινότητες (e-communities), οι οποίες είναι βασισμένες στα κοινά ενδιαφέροντα και εμπειρίες που αποκτώνται από κάθε ένα από τα μέλη τους αλλά και από την αλληλεπίδραση μεταξύ τους (interaction & interactivity) σαν σύνολα.

Στα πλαίσια αυτά η παρούσα Δ.Δ. προσφέρει τόσο μέσω της προτεινόμενης μεθοδολογίας ONARM όσο και του συστήματος λογισμικού Concept Net, σημαντική καινοτομική συνεισφορά σε αυτό που αποτελεί έναν από τους πιο σημαντικούς στόχους και πεδίο έρευνας σήμερα (EU FP7 - ICT 2009 – Intelligent Information Management – Extending the Personal Sphere), το οποίο ορίζεται σαν το πεδίο έρευνας εκείνο το οποίο ερευνά νέους τρόπους που θα επιτρέπουν στους χρήστες να διασφαλίζουν, διαχειρίζονται, οπτικοποιούν και να διερμηνεύουν τον προσωπικό τους γνωσιακό χώρο.

Επιχειρεί η παρούσα Δ.Δ., να εντοπίσει τη λύση στο πρόβλημα της ύπαρξης κοινών γνωσιακών ενδιαφερόντων, μεταξύ ατόμων και ατόμων, ατόμων και ομάδων αλλά και ομάδων και ομάδων ανθρώπων, έτσι ώστε να είναι δυνατή η προώθηση γνώσης από τον ένα στον άλλο, με αυτοματοποιημένο τρόπο, προσθέτοντας συνεχώς γνώση στο γνωστικό πεδίο (knowledge domain) του καθενός ανωτέρω.

Για την επίλυση του ανωτέρω, έχουν υλοποιηθεί στα πλαίσια της παρούσης, τεχνικές από τις παρακάτω αναφερόμενες γνωστικές περιοχές, ολοκληρωμένες σε μία καινοτόμο ενοποιημένη μεθοδολογία, τόσο με τη μορφή θεωρητικού υποβάθρου, όσο και μέσω λειτουργικής τεχνικής λύσης, τεχνικές που ανήκουν στα παρακάτω γνωστικά πεδία - περιοχές:

- Τη γνωστική περιοχή της Διαχείρισης Γνώσης, και πιο συγκεκριμένα της απεικόνισης και της ανάκτησης γνώσης
- Τη γνωστική περιοχή της εξόρυξης δεδομένων (data mining), έχοντας σαν επίκεντρο τον αλγόριθμο Apriori.
- Τη γνωστική περιοχή της Τεχνητής Νοημοσύνης και πιο συγκεκριμένα, την περιοχή της έξυπνης διερεύνησης γράφων και δένδρων τα οποία απεικονίζουν οντολογίες.
- Επίσης, στα πλαίσια της Τεχνητής Νοημοσύνης, η περιοχή του έξυπνου λογισμικού, με υλοποίηση έξυπνων αλγόριθμων, για την ανακάλυψη ενδιαφερόντων τμημάτων οντολογιών.

Συνολικά, η προτεινόμενη μεθοδολογία προτείνει σαν βασική έννοια – μοντέλο, την έννοια της υπεραγοράς, μέσα στην οποία οι καταναλωτές, καταναλώνουν - χρησιμοποιούν – διαδίδουν, έννοιες αντί προϊόντων και υπηρεσιών. Οι αλγόριθμοι που υλοποιήθηκαν είναι:

- Αλγόριθμος για την ανάγνωση εννοιών από οντολογίες περιγεγραμμένων σε μορφή RDF / RDF(s). (Νέο)
- Αλγόριθμος για την δημιουργία των δομικών στοιχείων δύο οντολογιών (Νέος αλγόριθμος και υλοποίηση, βασισμένος στη χρήση του πακέτου Jena / Java)
- Αλγόριθμος Apriori, για την εξόρυξη πιθανών υπό εξέταση μονοπατιών οντολογιών, για την εξεύρεση ενδιαφερόντων εννοιών (Νέα υλοποίηση, βασισμένος στον ήδη υπάρχοντα)
- Αλγόριθμος για την εξεύρεση κοινών κόμβων – εννοιών, τόσο από δομικής όσο και από λεκτικής απόψεως, μεταξύ δύο οντολογιών (Νέος).

- Αλγόριθμος για την ευθυγράμμιση, με αυτόματο τρόπο, οντολογιών, κοινού, παραπλήσιου η πλήρως διαφοροποιημένου λεξιλογίου, με βάση τους κόμβους αμοιβαίου ενδιαφέροντος (Νέος).
- Ορισμός μέτρων ενδιαφέροντος, για τους υπό εξέταση γράφους – οντολογίες και τους κόμβους – έννοιες τους. (Νέα)
- Ανακάλυψη σχετικών βαρών για τα παραπάνω μέτρα. Ανάλογα με την αντίστοιχη περίσταση (που απεικονίζεται στη δομή του γράφου), να υπάρχει εκ των προτέρων προγνωσιακό μοντέλο των βαρών των μέτρων, έτσι ώστε να προκύπτει το δυνατόν καλύτερο αποτέλεσμα εκ μέρους του συτήματος, ως προς την επιλογή των εννοιών, πιο κοντινό σε αυτό που θα επέλεγε και θα εκπροσωπούσε τον ανθρώπινο χρήστη και τις επιλογές του και σε συντομώτερο χρόνο (Νέα μεθοδολογία και υλοποίηση).

Όλα τα ανωτέρω, μαζί με την συνθετική υλοποίησή τους, σε ένα σύστημα, με μορφή εφαρμογής, η οποία εκτελείται σε φορητές συσκευές χρηστών, που έχουν προσωπικούς γράφους – οντολογίες, κοινών ή παραπλησίων ενδιαφερόντων, αποτελούν μία καινοτόμο μεθοδολογία η οποία ονομάστηκε ONARM.

Συνεισφορά της παρούσης σε τεχνολογικό υπόβαθρο

Η παρούσα Δ.Δ. συνεισφέρει στα παρακάτω επιστημονικά θέματα σχετικά με υλοποίηση και παρουσίαση νέων τεχνικών και τεχνολογιών:

1. Αλγόριθμος για αντιστοίχιση (mapping) οντολογιών
2. Νέα μεθοδολογία για εύρεση των σημαντικών μερών μιας οντολογίας
3. Ολοκληρωμένη εφαρμογή που υλοποιεί τα ανωτέρω στον τομέα του Τουρισμού.

1. Αλγόριθμος για αντιστοίχιση (mapping) οντολογιών

α. Χρησιμοποιεί κανόνες συσχέτισης (association rules) για την εξαγωγή των σημαντικών μονοπατιών κάθε επιμέρους οντολογίας, βάσει του αλγορίθμου Apriori.

β. Ακολούθως αναπτύχθηκε ο τρόπος με τον οποίο έγινε η βαθμονόμηση της σημαντικότητας του κάθε κόμβου στην ιεραρχία των 2 οντολογιών.

γ. Βαθμονόμηση ανά δύο όλων των κόμβων που προέκυψαν από το Apriori στο προηγούμενο βήμα (μεταξύ των 2 οντολογιών διαδικασίας) για την αξιολόγησή τους

2. Μεθοδολογία για εύρεση των σημαντικών μερών μιας οντολογίας

Εδώ ορίστηκαν 8 νέα μέτρα σημαντικότητας των κόμβων μίας οντολογίας. Αυτά προτάθηκαν με βάση τη δομή της οντολογίας και τη θέση μέσα σε αυτή του καθενός συγκεκριμένου κόμβου. Επίσης προτάθηκαν 8 αντίστοιχα μέρη με σκοπό την πρόταση ενός γενικευμένου συναρτησιακού μοντέλου F(w,n) το οποίο θα προτείνει ενδιαφέρουσες έννοιες της οντολογίας του κάθε χρήστη.

3. Εφαρμογή στον Τουρισμό

Αναπτύχθηκε εφαρμογή με οντολογίες τουριστικών εννοιών, η οποία λειτουργεί σε φορητές συσκευές για την επαλήθευση με πραγματικά δεδομένα των ανωτέρω.

Οργάνωση Διατριβής

Η παρούσα Δ.Δ. για την ευκολώτερη κατανόησή της έχει διαμορφωθεί ως είναι χωρισμένη σε έξι κύρια κεφάλαια και ένα παράρτημα, στα οποία παρουσιάζεται αναλυτικά το πρόβλημα της ανάκτησης και της έξυπνης διανομής γνώσης και περιεχομένου. Αυτών, προηγούνται τα ανεξάρτητα τμήματα της περίληψης, περιεχομένων, εισαγωγής, καινοτομίας και συνεισφοράς της παρούσης Δ.Δ σε επιστημονικό επίπεδο καθώς επίσης και του τμήματος οργάνωσης της διατριβής.

Στο πρώτο κεφάλαιο εξετάζεται αναλυτικά το θεωρητικό υπόβαθρο του προβλήματος σε συσχέτιση με την ανάπτυξη του διαδικτύου και την μετεξέλιξή του ουσιαστικά σε ένα δίκτυο εννοιών, τον αποκαλούμενο Σημασιολογικό Ιστό (Semantic WEB). Ουσιαστικά στο κεφάλαιο αυτό γίνεται παρουσίαση των δομών, τω εργαλείων και των περιβαλλόντων απεικόνισης και διαχείρισης γνώσης, τα οποία σε επόμενα κεφάλαια θα χρειαστούν για την κατανόησή τους.

Στο δεύτερο κεφάλαιο παρουσιάζεται το θεωρητικό υπόβαθρο της προτεινόμενης μεθοδολογίας, τα κυριότερα προβλήματα που αντιμετωπίστηκαν καθώς και προηγούμενες προσεγγίσεις στην επίλυση του προβλήματος αυτού. Επίσης γίνεται αξιολόγηση ως προς την αποδοτικότητα των προτεινομένων από την βιβλιογραφία προσεγγίσεων προκειμένου να αξιολογηθούν σε κατοπινό κεφάλαιο σε σχέση με τη προτεινόμενη μεθοδολογία ONARM.

Στο τρίτο κεφάλαιο παρουσιάζεται η προτεινόμενη μεθοδολογία ONARM και τα αποτελέσματά της, η οποία προτάθηκε και υλοποιήθηκε στα πλαίσια της παρούσης Δ.Δ. Στα πλαίσια αυτά στο κεφάλαιο αυτό, παρουσιάζονται εκτός από το σύνολο της ανωτέρω μεθοδολογίας και εμπειρικά δεδομένα και αποτελέσματα τα οποία χρησίμευσαν σαν επαληθεύσεις του συνόλου αλλά και των επιμέρους αλγορίθμων της μεθοδολογίας ONARM και τα οποία αποτελούν τμήματα δημοσιεύσεων.

Στο τέταρτο κεφάλαιο παρουσιάζεται η μελέτη εφαρμογής λογισμικού, η οποία προέκυψε από το θεωρητικό υπόβαθρο της ONARM μεθοδολογίας. Πιο συγκεκριμένα, παρουσιάζεται η ανάλυσή της η αρχιτεκτονική της, το περιβάλλον υλοποίησής της, τα σενάρια χρήσης της, τα user interfaces της καθώς και η τελική υλοποίησή της. Επίσης, στο ίδιο κεφάλαιο

παρουσιάζονται και διάφοροι τομείς στους οποίους τόσο η μεθοδολογία όσο και η αντίστοιχη εφαρμογή θα μπορούσαν να εφαρμοστούν με επιτυχία. Επίσης παρουσιάζονται και τα αντίστοιχα συμπεράσματα του κεφαλαίου αυτού.

Στο πέμπτο κεφάλαιο παρουσιάζονται και διερευνώνται πρόσθετα πεδία εφαρμογή της μεθοδολογίας ONARM και της εφαρμογής CONCEPT-NET, σαν μετεξέλιξη της παρούσης Δ.Δ.

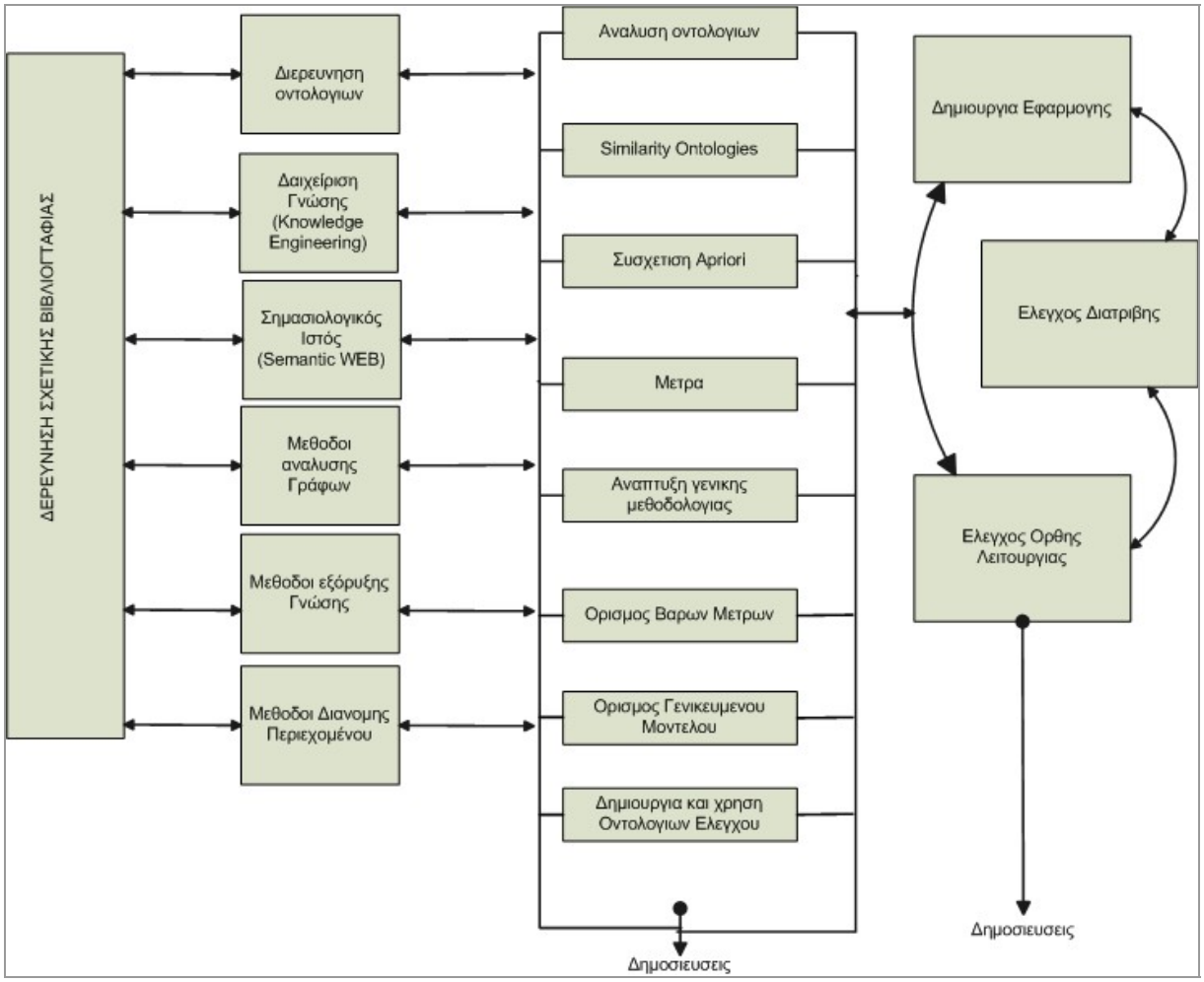
Στο έκτο κεφάλαιο παρουσιάζονται τα γενικά συμπεράσματα του συνόλου της διδακτορικής διατριβής καθώς και προτάσεις για περαιτέρω βελτιώσεις τόσο σε μεθοδολογικό όσο και σε τεχνικό επίπεδο.

Επίσης παρουσιάζονται τα κυριότερα κομμάτια κώδικα του αλγοριθμικού τμήματος της μεθοδολογίας, όπως υλοποιήθηκαν στην εφαρμογή CONCEPT-NET, σε ξεχωριστό ηλεκτρονικό παράρτημα της παρούσης, στο CD που τη συνοδεύει.

Ακολουθούν το γλωσσάριο, το οποίο περιλαμβάνει σύντομες επεξηγήσεις σημαντικών εννοιών, η κατανόηση των οποίων κρίθηκε σκόπιμη και απαραίτητη για την περαιτέρω επεξήγηση και κατανόηση της παρούσης Δ.Δ.

Τέλος, σαν ανεξάρτητο τμήμα της παρούσης ακολουθεί η βιβλιογραφία στην οποία βασίστηκε η εκπόνηση της παρούσης.

Στο παρακάτω διάγραμμα παρουσιάζεται η μεθοδολογία εκπόνησης της διατριβής και τα διάφορα στάδια που ακολουθήθηκαν κατά τη διάρκεια αυτής.



Εικόνα - Μεθοδολογία Ανάπτυξης Διδακτορικής Διατριβής

Αρχικά, έγινε βιβλιογραφική έρευνα στους τομείς των οντολογιών, της διαχείρισης γνώσης, του σημασιολογικού ιστού, των μεθόδων ανάλυσης γράφων, των υπαρχόντων μεθόδων εξόρυξης γνώσης και των τρόπων διανομής περιεχομένου. Από αυτά προέκυψαν οι ερευνητικοί τομείς της παρούσης Δ.Δ., της ανάλυσης οντολογιών με βάση την ομοιότητα τους, μέσω του αλγορίθμου Apriori και νέων μέτρων σημαντικότητος ενδιαφερόντων μερών οντολογιών. Έγινε ακολούθως ανάπτυξη της γενικής μεθοδολογίας ONARM και ορισμός γενικευμένου μοντέλου πρόγνωσης ομοιότητος οντολογιών, με βάση ευρέως αποδεκτές οντολογίες (benchmarking) αλλά ειδικών οντολογιών ελέγχου που δημιουργήθηκαν για το σκοπό αυτό στα πλαίσια της παρούσης Δ.Δ. Από τα ανωτέρω προέκυψαν οι σχετικές δημοσιεύσεις και η υλοποίηση της εφαρμογής CONCEPT-NET στην περιοχή του τουρισμού.

Κεφάλαιο 1. Θεωρητικό Υπόβαθρο του Προβλήματος

1.1. Εισαγωγη

Η ανάκτηση γνώσης είναι ένα είδος ανάκτησης πληροφοριών από μη δομημένα ή ημιδομημένα δεδομένα (unstructured or semi-structured data – τυπικά κείμενα γραμμένα σε φυσική γλώσσα, ιστοσελίδες κ.α.) και αποτελεί κλάδο της σύγχρονης επιστήμης των υπολογιστών.

Η ανάκτηση γνώσης υπάγεται στο γενικότερο κλάδο της τεχνητής νοημοσύνης (Artificial Intelligence – AI) και ειδικότερα στον τομέα διαχείρισης γνώσης (Knowledge Engineering – KE). Εκεί συναντά όλους τους τομείς και τα πεδία γνώσης καθώς και τις σχετικές τεχνικές με το χτίσιμο, τη συντήρηση και τη χρήση συστημάτων που αφορούν τη διαχείριση της γνώσης. Η διαχείριση γνώσης αντιμετωπίζει σύνθετα προβλήματα απεικόνισης, καταγραφής και χρήσης εννοιών και ορίζεται σαν ερευνητικός τομέας στο κοινό υπόβαθρο που ορίζεται από την επιστήμη των υπολογιστών (computer science) σε διάφορες εκφάνσεις αυτής, όπως η τεχνική νοημοσύνη, τα έμπειρα συστήματα (expert systems), την τεχνική ανάπτυξη λογισμικού (software engineering) και τα συστήματα λήψης αποφάσεων (decision support systems). Επίσης, ο τομέας της διαχείρισης γνώσης, βρίσκεται στενά συνδεδεμένος με κοινωνικά και οικονομικά πεδία, δεδομένου ότι τα δεδομένα στα οποία αναφέρεται, επεξεργάζεται και εν τέλει από τα οποία βγάζει αποτελέσματα, αφορούν εφαρμογές τέτοιου τύπου, με έμφαση στα επιχειρηματικα δεδομένα. Τέλος, ο τομέας της διαχείρισης γνώσης βρίσκεται στενά συνδεδεμένος και χρησιμοποιεί τομείς της μαθηματικής λογικής και ψυχολογίας, έτσι ώστε να είναι δυνατόν να δημιουργούνται υπολογιστικές δομές και πλαίσια ικανές να αναπαραστήσουν την ανθρώπινη γνώση που προκύπτει και παράγεται για διάφορους τομείς δραστηριότητας, που βασίζονται σε απεικονίσεις οντολογιών, έτσι ώστε να είναι δυνατή η διαμοίραση γνώσης σε πολλούς ταυτόχρονους χρήστες του συστήματος.

Η ανάκτηση γνώσης πρακτικά, συνίσταται στην αναγνώριση συγκεκριμένου είδους πληροφοριών, όπως κύρια ονόματα (ονόματα ανθρώπων, τοπωνύμια, ονόματα εταιρειών, ημερών, μηνών, κτλ.), χρονικές πληροφορίες (ημερομηνίες), σχέσεις και γεγονότα από

(συνήθως) ηλεκτρονικά κείμενα που με πρώτη ματιά φαίνονται ασύνδετα μεταξύ τους. Απώτερος στόχος της ανάκτησης γνώσης είναι η «κατανόηση» των βασικών συστατικών του υπό ανάλυση κειμένου, τα οποία αργότερα μπορούν να χρησιμοποιηθούν από εφαρμογές όπως αυτόματη εξαγωγή περιλήψεων κειμένων, αυτόματη απάντηση ερωτήσεων, αυτόματη μετάφραση, προώθηση πωλήσεων και γενικότερα δραστηριότητες (processes) διανομής γνώσης.

Τα βασικά στάδια της ανάκτηση γνώσης είναι συνήθως πέντε, σύμφωνα με τον οδηγό των συνεδρίων MUC (Message Understanding Conference) (Chinchor N., Robinson P., 1997):

- Αναγνώριση κυρίων ονομάτων,
- Αναγνώριση αναφορικών σχέσεων μεταξύ ονομάτων,
- Αναγνώριση γεγονότων,
- Αναγνώριση σχέσεων μεταξύ κυρίων ονομάτων και
- Αναγνώριση σχέσεων μεταξύ γεγονότων.

Η έρευνα στην περιοχή ανάκτησης γνώσης επηρεάστηκε σε μεγάλο βαθμό από τα MUC συνέδρια, τα οποία δημοσίευαν τα αποτελέσματα διαγωνισμών μεταξύ συστημάτων που επιχειρούσαν ανάκτηση γνώσης σε κείμενα που περιγράφουν καθημερινές ειδήσεις. Η παρούσα Δ.Δ. έχει σαν αντικείμενο την ανάκτηση γνώσης, η οποία είναι εκφρασμένη σε έννοιες, περιγεγραμμένες σε οντολογίες.

Σημασία της ανάκτησης γνώσης

Ο ρόλος της ανάκτησης γνώσης (knowledge extraction), στα πλαίσια της διαχείρισης γνώσης (knowledge engineering - managmnet), είναι η αναγνώριση εξειδικευμένης πληροφορίας από μη δομημένη πληροφορία (unstructured data), με μηχανικό (αυτόματο) τρόπο, και κατόπιν, ο συσχετισμός και ο μετασχηματισμός της σε ωφέλιμη γνώση. Αντίθετα με την κλασσική ανάκτηση πληροφοριών (Information Retrieval), σύμφωνα με την οποία η αναζήτηση γίνεται με βάση συγκεκριμένες λέξεις-κλειδιά και το αποτέλεσμα περιλαμβάνει μόνο κείμενα στα οποία βρίσκεται (ενδεχομένως) η χρήσιμη πληροφορία, η ανάκτηση γνώσης στοχεύει ακριβώς στην αναγνώριση της χρήσιμης μόνο πληροφορίας και το περιβάλλον (context) στο οποίο αυτή εμφανίζεται.

Δεδομένου του μεγάλου όγκου πληροφοριών που παράγονται και διακινούνται σήμερα (κύριο χαρακτηριστικό του διαδικτύου), το ζητούμενο στις μέρες μας είναι όχι απλώς η κατοχή της πληροφορίας – ο οποιοσδήποτε σήμερα μπορεί να έχει πρόσβαση σε σχετικά οποιαδήποτε πληροφορία – αλλά η διαχείριση και συσχέτιση της πληροφορίας καθώς επίσης και ο εντοπισμός περαιτέρω «σχετικής» πληροφορίας και διανομή αντίστοιχου περιεχομένου. Έτσι, ενώ με μια κλασσική μηχανή αναζήτησης ο ενδιαφερόμενος θα λάβει ως απάντηση ένα σύνολο κειμένων που ενδεχομένως περιέχουν την απάντηση που περιμένει, η ανάκτηση γνώσης στοχεύει στην απάντηση και μόνο σε αυτή σε περιεχόμενο το οποίο εμφανώς να μην είναι άμεσα συνδεδεμένο αλλά παραμένει συσχετίσιμο και χρήσιμο στον τελικό χρήστη.

Τα ανωτέρω μπορούν να βρουν εφαρμογή σε πάρα πολλούς τομείς της καθημερινής δραστηριότητος, ιδιαίτερα σε πολύπλοκα γνωσιακά πεδία τα οποία παράγουν πολυπληθή πληροφορία συνεχώς.

Στα πλαίσια αυτά, εκτενής έρευνα διεξάγεται, με σκοπό την ανακάλυψη και παρακολούθηση της πληροφορίας, η οποία παράγεται σε πραγματικό χρόνο, μαζικά, έτσι ώστε να προταθούν μεθοδολογίες και προϊόντα λογισμικού τα οποία θα είναι σε θέση, σε πραγματικό χρόνο, να κατηγοριοποιούν αυτή ακόμα και σε περιπτώσεις μη παροχής πλήρους πληροφορίας η/και αντικρουόμενης πληροφορίας, με σκοπό να παράγουν κατηγοριοποιημένη γνώση από αυτή περιγράφοντάς και οργανώνοντάς την με μεταπληροφορία, άμεσα επαναχρησιμοποιήσιμη, από ένα πλήθος ενδιαφερομένων χρηστών.

Αποτελέσματα – Επιδόσεις

Οι επιδόσεις αναλόγων συστημάτων που έχουν παρουσιαστεί κατά καιρούς σε συνέδρια και ερευνητικές εργασίες αναφέρονται σε έρευνα πάνω σε συγκεκριμένους τομείς (κείμενα γραμμένα για συγκεκριμένους θεματικούς χώρους) και διαφέρουν ανάλογα με το είδος της πληροφορίας που στοχεύουν να αναγνωρίσουν. Έτσι, αναφορικά με της πέντε προαναφερθείσες κατηγορίες της ανάκτησης γνώσης, έχουν παρατηρηθεί τα κάτωθι:

Όσο προχωράμε από την αναγνώριση κυρίων ονομάτων προς την αναγνώριση σχέσεων μεταξύ γεγονότων η δυσκολία αυξάνεται εκθετικά, αφού τα μετέπειτα στάδια ανάλυσης

εξαρτώνται σε μεγάλο βαθμό από τα προγενέστερα. Έχοντας ως δεδομένο το γεγονός αυτό, η αναγνώριση κυρίων ονομάτων επιτυγχάνεται κατά 93% περίπου κατά τον Cunningham (Cunningham H., 2000) και λαμβάνοντας υπ' όψιν το γεγονός ότι ένας άνθρωπος δε μπορεί να επιτύχει το 100%, μπορούμε να ισχυριστούμε πως τα συστήματα αναγνώρισης ονομάτων τα καταφέρνουν το ίδιο ή σχεδόν το ίδιο καλά με τον άνθρωπο.

Όμως, τα επόμενα στάδια ανάλυσης δεν εφαρμόζονται με την ίδια επιτυχία. Και τελικά, η αναγνώριση γεγονότων (που είναι ο κύριος στόχος της ανάκτησης γνώσης) δεν εφαρμόζεται με μεγάλη επιτυχία. Τα καλύτερα συστήματα αναφέρουν επιδόσεις της τάξεως του 60%, σύμφωνα με τα πρακτικά των συνεδρίων MUC (τα νούμερα που αναφέρονται εδώ δηλώνουν κατά προσέγγιση την επίδοση των συστημάτων και αποτελούν το F-measure των αποτελεσμάτων).

Στο σημείο αυτό πρέπει να σημειώσουμε δύο πράγματα. Πρώτον, τα αναφερόμενα μεγέθη αναφέρονται σε πειράματα πάνω σε συγκεκριμένους τομείς (domains) και συγκεκριμένες γλώσσες (Αγγλικά συνήθως). Το γεγονός αυτό είναι ιδιαίτερα σημαντικό, αφού κάθε τομέας και κάθε γλώσσα έχουν ιδιαιτερότητες οι οποίες πρέπει να ενσωματωθούν στα συστήματα ανάκτησης γνώσης. Από τη φύση της λοιπόν η ανάκτηση γνώσης είναι άρρηκτα συνδεδεμένη με τη γλώσσα και το είδος των δεδομένων (κειμένων) που πρόκειται να αναλυθούν.

Δεύτερον, η σχετική έρευνα έχει γίνει πιο εντατική τα τελευταία χρόνια. Συνεχώς προτείνονται νέες μέθοδοι κατασκευής αναλόγων αρχιτεκτονικών και συστημάτων οι οποίες στοχεύουν σε καλύτερες επιδόσεις και στην απλοποίηση της επέκτασης της λειτουργίας υπαρχόντων συστημάτων σε νέους τομείς και νέες γλώσσες. Έμφαση δίνεται κυρίως στην ανάπτυξη ειδικών συστημάτων μέσω αυτόματης εκμάθησης της γλωσσικής πληροφορίας και των ιδιαιτεροτήτων του χώρου από το ίδιο το σύστημα. Στο μέλλον, λοιπόν, είναι βέβαιο ότι θα έχουμε καλύτερα συστήματα και σαφώς καλύτερες επιδόσεις.

1.2. Ο Σημασιολογικος Ιστος

Στα πλαίσια της παρούσης Δ.Δ., η αναζήτηση γνώσης συνδέεται άρρηκτα με τον τρόπο με τον οποίο αυτή απεικονίζεται και περιγράφεται. Πιο συγκεκριμένα, θεωρείται ότι τα δεδομένα τα οποία μέχρι σήμερα απλά είναι καταγεγραμμένα σε διάφορες δομές δεδομένων (π.χ. βάσεις δεδομένων), είναι αναγκαίο πλέον να περιγράφονται επιπρόσθετα με μεταπληροφορία, η οποία να δίνει την αίσθηση της σημασίας του συγκεκριμένων δεδομένων. Κατ' αυτόν τον τρόπο, η υπάρχουσα πληροφορία αποκτά πρόσθετη αξία με τη σημασιολογία και είναι έτοιμη να διαλειτουργήσει με άλλη παρόμοια πληροφορία, αποθηκευμένη επίσης κατά τρόπο σημασιολογικό. Οι δομές που χρησιμοποιούνται κατά κύριο λόγο για την αποθήκευση της μεταπληροφορίας, είναι οι οντολογίες, πράγμα το οποίο πραγματεύεται η παρούσα Δ.Δ.

1.2.1. Ορισμος

Η λέξη semantic (σημασιολογία) σημαίνει «έννοια του...». Semantic (σημασιολογία) κάποιου είναι η σημασία ή έννοια αυτού. Το semantic web είναι η μορφή του web, έτσι ώστε να περιγράφονται τα στοιχεία που βρίσκονται σε αυτό, με τέτοιο τρόπο ώστε να είναι άμεσα κατανοητά από τους υπολογιστές σε εννοιολογικό επίπεδο. Αυτό είναι πολύ κρίσιμο, δεδομένου ότι μέσω αυτού, επιτυγχάνεται η συνδεσιμότητα του περιεχόμενου και από εκεί και πέρα η εξόρυξη του πλησιέστερα σχετιζομένου περιεχομένου με αυτό, που βρίσκεται σε διάφορους άλλους ιστοτόπους (sites) και υπό διαφορετική μορφή (κείμενο, φωτογραφίες, video κλπ), μέσω της μεταπληροφορίας για αυτό.

Για να γίνει κατανοητό το πόσο πολύπλοκο μπορεί να γίνει το πρόβλημα της εννοιολογικής προσέγγισης του περιεχομένου ας θεωρήσουμε το παρακάτω παράδειγμα:

- Τα «Σκαθάρια» ήταν ένα Pop συγκρότημα από το Liverpool
- Ο John Lennon ήταν ένα μέλος των «Σκαθαριών»
- Το "Hey Jude" τραγουδήθηκε από τα «Σκαθάρια»
- Τα «Σκαθάρια πετούν» (το συγκρότημα)
- Τα «Σκαθάρια δεν πετούν (το γνωστό έντομο)

Στο παραπάνω παράδειγμα, προκύπτει αδυναμία, λόγω μεγάλης διαφοροποίησης του εκάστοτε περιεχομένου, να εξευρεθεί με τις τρέχουσες τεχνικές, σχετιζόμενο περιεχόμενο

ανάλογα με το προφίλ του εκάστοτε χρήστη, που να προκύπτει από την άμεση και ορθή κατανόηση των εννοιών, για τις οποίες ενδιαφέρεται.

Παραδείγματος χάριν:

- Ενδιαφέρεται ο χρήστης μόνον για τα «Σκαθάρια» επιφανειακά, σαν συγκρότημα;
- Ενδιαφέρεται ο χρήστης, σε ποιο αναλυτικό επίπεδο, για τη μελέτη του συγκροτήματος «Σκαθάρια» ;
- Ενδιαφέρεται ο χρήστης σε μία μουσικολογική έρευνα της δεκαετίας του 60-70; Υπάρχουν στοιχεία για live εμφανίσεις του συγκροτήματος «Σκαθάρια»;
- Μήπως εν τέλει, ο χρήστης ανήκει στο χώρο της βιολογίας, ο οποίος απλά έχει συγκεκριμένες μουσικές προτιμήσεις, οπότε προσεγγίζει την έννοια «Σκαθάρια» από τελείως διαφορετικές σκοπιές ταυτόχρονα;
- Τα «Σκαθάρια» έχουν πόδια και φτερά ή ρόδες;

Τέτοιου είδους γεγονότα, πολύ συχνά αντικρουόμενα μεταξύ τους, αποτελούν την γνώση η οποία είναι καταγεγραμμένη στον παγκόσμιο ιστό και είναι άμεσα αναγνωρίσιμη από τους ανθρώπους, σε κάποιο πεδίο εφαρμογής. Με βάση αυτή τη γνώση, ο ανθρώπινος νους είναι σε θέση να συσχετίσει γεγονότα και να εξάγει συμπεράσματα προκειμένου να βγάλει νέα συμπεράσματα τα οποία δεν βγαίνουν από πρωτογενή ανάλυση και δίνουν πρόσθετη αξία στην ήδη καταγεγραμμένη γνώση. Το πρόβλημα το οποίο αντιμετωπίζεται στα πλαίσια του semantic web, είναι να αποδοθεί νόημα και σημασιολογία στους συντακτικούς κανόνες που ορίζουν την δομή του περιεχομένου και από τους οποίους προκύπτει η κατανόησή και η συσχέτισή των περιεχομένων. Έτσι, ο σημασιολογικός ιστός εξελίσσεται από αυτό το οποίο ήταν το πρότυπο και ονομάστηκε WEB 1.0 και επρόκειτο γιά την προώθηση του περιεχομένου με βάση τις συνδέσεις (links) μεταξύ σελίδων περιεχομένου στατικά. Αντί αυτού η μετεξέλιξή του, το WEB 2.0 καλείται να υλοποιήσει τη διασύνδεση του περιεχομένου με βάση την εννοιολογική συσχέτιση και την κατανόηση των περιγραφομένων εννοιών. Κατ αυτόν τον τρόπο, το περιεχόμενο περιγράφεται όχι μόνο ακριβέστερα, αλλά και η προώθησή του και διανομή του, στα πλαίσια της οποίας έγκειται και η εξόρυξη σχετικών πληροφοριών, γίνεται αποτελεσματικότερα. Για να επιτευχθεί

αυτό, ο σημασιολογικός ιστός χρησιμοποιεί δομές όπως οι σχέσεις μεταξύ «πραγμάτων» και εννοιών καθώς και ιδιοτήτων και χαρακτηριστικών που διέπουν αυτές.

1.2.2. Δομες Περιγραφης Περιεχομενου Στο Σημασιολογικο Ιστο

Στο σημείο αυτό θα γίνει μία σύντομη της κύρια γλώσσας περιγραφής περιεχομένου σήμερα στο σημασιολογικό ιστό τη γλώσσα RDF, η οποία και θα περιγραφεί λεπτομερώς παρακάτω σε ειδικό κεφάλαιο της παρούσης διατριβής, λόγω της τεράστιας δυναμικής που αναπτύσσεται γύρω από αυτήν κατά την ανάπτυξη περιεχομένου και δεδομένων στα πλαίσια του WEB 2.0 και την εκτενή χρήση της στην ανάπτυξη του προτεινόμενου λογισμικού Concept Net.

Σε γενικές γραμμές, η RDF (Resource Description Language), είναι μία γλώσσα για τον χαρακτηρισμό της πληροφορίας σε επίπεδο meta (metalevel markup language). Κατ' αυτόν τον τρόπο είναι δυνατή η εισαγωγή της μεταπληροφορίας (metadata) η οποία ορίζεται σαν data about data. Έτσι, γίνεται εφικτή η αυτοματοποιημένη ανάκτηση γνώσης από ενδιάμεσους ευφυείς διαμεσολαβητές (intelligent agents) όπως αυτοί χρησιμοποιούνται σήμερα για την ανάκτηση γνώσης με την μορφή των Web spiders, Web robots κλπ, τα οποία κάνουν χρήση αυτής της μεταπληροφορίας και αυτοεκπαιδεύονται για την έρευνα, την ανακάλυψη, την εξεύρεση, την ανάκτηση, την ανάλυση και την επεξεργασία της πληροφορίας με σκοπό την οργάνωση της γνώσης. Ακόμα δε περισσότερο, τα αρχεία τύπου RDF εκτός από (δείκτες) pointers σε δομές δεδομένων χρησιμοποιούνται και σαν δείκτες σε άλλα αρχεία RDF τα οποία με τη σειρά τους περιγράφουν δομές δεδομένων και ούτω καθ' εξής.

Η τρέχουσα έρευνα επικεντρώνεται αυτή την στιγμή σε προσπάθειες ενοποίησης των RDF αρχείων, δηλαδή των δομών που υποστηρίζουν και περιγράφουν εννοιολογικό περιεχόμενο, όπως οι οντολογίες σε ανώτερο επίπεδο. Ερευνητικοί χώροι που άπτονται άμεσα των ανωτέρω, είναι οι συγκρίσεις και οι αποστάσεις μεταξύ εννοιών στην ίδια ή σε διαφορετικές τέτοιες δομές (οντολογίες), η εξεύρεση μετρικών δομών metrics) οι οποίες να είναι σε θέση υπολογίσουν τη εννοιολογική γειτνίαση μεταξύ ολοκλήρων συνόλων ή υποσυνόλων οντολογιών (similarity – dissimilarity measures) για την εξεύρεση κοινών

κόμβων και μονοπατιών στην περιγραφή με σκοπό την ανταλλαγή γνώσης μεταξύ υπολογιστικών συστημάτων κλπ.

Η εκμετάλλευση των πλήρων δυνατοτήτων του σημασιολογικού ιστού, όπως αυτές υπάρχουν και έχουν γίνει standard στα πλαίσια του World Wide Web Consortium - W3C, ενέχουν μεγάλη δυσκολία χρήσης κυρίως λόγω της μεγάλης εκπαιδευτικής καμπύλης που απαιτείται κατά τη διαδικασία όσο και των συνεχών εξελίξεων που συμβαίνουν λόγω των διαρκών αυξανομένων σημασιολογικών αναγκών και την ανάγκη ύπαρξης εφαρμογών που υποστηρίζουν δυνατότητας σημασιολογίας (semantic enabled εφαρμογών). Στα πλαίσια της παρούσας διατριβής, θα γίνει ανάπτυξη εφαρμογής, η οποία θα υποστηρίζει την εφαρμογή μεταπληροφορίας καθώς και έξυπνων μηχανισμών για την διασύνδεση περιεχομένου και εννοιών σύμφωνα με τα τρέχοντα πρότυπα που ισχύουν στον σημασιολογικό ιστό.

Μία από τις πιο συχνά χρησιμοποιούμενες εφαρμογές του σημασιολογικού για τη εξεύρεση και δυναμική διανομή περιεχομένου (push), είναι αυτή της χρήσης του πρωτοκόλλου RSS.

Στο σημασιολογικό ιστό η γλώσσα που χρησιμοποιείται για να περιγράφονται πόροι του web (web resources) είναι η Resource Description Language ή RDF.

Προϊστορία

Λόγω των διαρκώς αυξανόμενων αναγκών, δημιουργήθηκε μία ομάδα εργασίας (Task Force) στα πλαίσια του οργανισμού W3C, ο οποίος είναι υπεύθυνος για τις διαδικασίες δημιουργίας standards για οτιδήποτε αφορά το WEB. Το 1997 ορίστηκε το πρώτο πλαίσιο περιγραφής RDF και το 1999 οριστικοποιήθηκε σαν πρόταση για standard. Αυτό που ορίστηκε σαν σχήμα RDF είναι μία γλώσσα περιγραφής, η οποία βασίζεται σε ομάδες των τριών οντοτήτων (triples) οι οποίες είχαν σκοπό να περιγράφουν τις δομές URI στον σημασιολογικό ιστό. Κατ' αυτόν τον τρόπο και με την αποδοχή αυτών των standards, αυξήθηκε η διαλειτουργικότητα μεταξύ τέτοιων συστημάτων περιγραφής εννοιών. Έτσι από το 2001 και μετά για παράδειγμα (Shadbolt N., et. al., 2006), μία οντολογία που κατασκευάστηκε και υπάρχει στα πλαίσια της Ιατρικής Επιστήμης προσφέρει διαλειτουργικότητα και μπορούν να αντλούν από εκεί πληροφορίες ή να συσχετίζουν ακόμα περισσότερο αυτές, επιστήμονες από διαφορετικού ερευνητικούς τομείς και

επιστημών, όπως για παράδειγμα γονιδιακή τεχνολογία (genomics), proteomics, από το τομέα δοκιμής νέων φαρμάκων, της επιδημιολογίας και ούτω κάθε εξής.

Αυτό έχει καταστεί δυνατόν με την υποστήριξη των εννοιών δομημένων σε οντολογίες οι οποίες είναι διαθέσιμες προς χρήση στο διαδίκτυο (http://obo.sourceforge.net) με καθεστώς ελεύθερης χρήσης. Άλλη μία τέτοια ομάδα επιστημών οι οποίες χρησιμοποιούν αντίστοιχες δομές αναπαράστασης γνώσης, είναι αυτές της υδρολογίας, οικολογίας, κλιματολογίας και ωκεανογραφίας οι οποίες ανέπτυξαν την οντολογία http://marinemetadata.org/examples/mmihostedwork/ontologieswork προσδίδοντας έτσι την δυνατότητα κατανόησης φαινομένων και λειτουργιών πέρα από τα στενά όρια της κάθε μίας επιστήμης ξεχωριστά, ολοκληρώνοντας κατ' αυτόν τον τρόπο αποθήκευσης και διαχείρισης των δεδομένων τους (data integration)

Για να γίνει εύκολα κατανοητή η δραματική αλλαγή που θα επιφέρει μία τέτοια μετεξέλιξη του διαδικτύου σε ενοποιημένο σημασιολογικό ιστό (WEB to Semantic WEB Transformation), ας υποθέσουμε ότι στα πλαίσια του WEB 2.0 έχουμε το ενδεχόμενο ένας χρήστης του διαδικτύου να θέλει να αγοράσει ένα αυτοκίνητο το οποίο να ικανοποιεί συγκεκριμένα κριτήρια και περιορισμούς. Μερικά από αυτά θα μπορούσε να είναι ο προϋπολογισμός, το έτος κατασκευής, το μοντέλο, η εταιρεία κατασκευής κλπ.

Ας εξετάσουμε τα δύο σενάρια ξεχωριστά:

Σενάριο βασισμένο στο WEB 1.0

Στα πλαίσια αυτού του σεναρίου ο χρήστης και υποψήφιος αγοραστής θα έπρεπε να κάνει στην αρχή μία εκτεταμένη έρευνα για το αν υπάρχουν συγκεκριμένα WEB sites τα οποία να υποστηρίζουν την αγοραπωλησία μέσω διαδικτύου. Κατόπιν, και εφ' όσον εντόπιζε αυτά θα έπρεπε να καταφύγει σε καθένα από αυτά και να εκτελέσει έρευνες για το συγκεκριμένο μοντέλο αυτοκινήτου που επιθυμούσε. Ακολούθως και σε περίπτωση επιτυχίας των ερωτημάτων που υπέβαλλε θα έπρεπε να κάνει refine αυτά περιορίζοντας τα αποτελέσματα της έρευνας εμπλουτίζοντας την με καθένα από τα κριτήρια – περιορισμούς που είχε στην αρχή θέσει. Πρέπει να σημειωθεί ότι αυτή η διαδικασία πρέπει να εκτελεσθεί για καθένα από τα sites που είναι υποψήφια για αυτό. Επίσης, πρέπει να σημειωθεί ότι ο χρήστης πρέπει να μάθει και να υιοθετήσει το «συντακτικό» και τον τρόπο

χρήσης του καθενός τέτοιου περιβάλλοντος, δεδομένου ότι κατά κόρον αυτά θα είναι διαφορετικά και ως προς την δομή και ως προς την περιγραφή του περιεχομένου. Τέλος, θα πρέπει να τονιστεί ότι σε ένα τέτοιο σενάριο αναζήτησης λύσης στο πρόβλημα αναζήτησης αυτοκινήτου το εργαλείο το οποίο θα χρησιμοποιήσει ο υποψήφιος χρήστης είναι έρευνες με λέξεις κλειδιά και τελεστές (operators) μεταξύ τους. Είναι φανερό ότι και αν όλα επιτύχουν σε μία τέτοιου είδους έρευνα, τα αποτελέσματα θα επιστρέφουν σε διαφορετική, μη ενοποιημένη μορφή μεταξύ των sites και άρα δεν θα έχουν καμία πρόσθετη αξία για τον χρήστη.

Σενάριο βασισμένο στο WEB 2.0

Στην περίπτωση του υποψήφιου αγοραστή ο οποίος θα λειτουργούσε στα πλαίσια του WEB 2.0, το σενάριο χρήσης γίνεται πιο δομημένο, πιο απλό και εν τέλει πιο αποτελεσματικό. Ας υποθέσουμε για αυτό ότι ο χρήστης Α είναι ο υποψήφιος αγοραστής ενώ ο χρήστης Π είναι κάποιος υποψήφιος πωλητής. Με βάση όσα έχουν περιγραφεί ανωτέρω οι δύο χρήστες δημιουργούν μέσω γραφικών διεπαφών (Graphical User Interfaces - GUIs), τα αντίστοιχα RDF αρχεία που περιέχουν το περιεχόμενο τους.

Ακολούθως, την χρήση τους την αναλαμβάνουν αυτόνομα προγράμματα τα οποία ενσωματώνουν επιχειρηματική εξυπνάδα, η ανακύπτει από την εξαγωγή των εννοιών που περιγράφονται σαν metadata στα αρχεία RDF των Α και Π ανωτέρω.

Ακολούθως, τα δύο αυτά «προγράμματα», τύπου agent, με τις μορφές ISA (I Sell Agent) και IBA (I Buy Agent), κινούνται συνεχώς μέσα στο διαδίκτυο εκπέμποντας μηνύματα. Υπάρχουν διαδικτυακοί τόποι οι οποίοι είναι δηλωμένοι σε αυτούς τους agents (marketplaces) όπου τέτοιου είδους ευφυή προγράμματα μπορούν με επιτυχία και ανταλλάξουν τα μηνύματα τους και να οδηγηθούν σε περαιτέρω πράξεις (actions) και αποφάσεις (decisions). Αποτέλεσμα αυτής της διαδικασίας είναι η ενημέρωση των χρηστών Α και Π ότι υπάρχει ένας ικανός αριθμός λύσεων στα προβλήματα “I Buy” και “I Sell” των Α και Π αντίστοιχα. Κατόπιν, οι Α και Π ενημερώνονται συγκριτικά και με βάση τους περιορισμούς που έχουν θέσει για το ποιες επιλογές μπορεί να αποτελέσουν τελικές λύσεις στα ανωτέρω προβλήματα, προσφέροντας κατ΄ αυτόν τον τρόπο βέλτιστες λύσεις στο πρόβλημα εξατομικευμένης παροχής γνώσης.

Επίσης εκτός από το καθεαυτό πρόβλημα που καλείται να επιλύσει μία τέτοιου είδους εφαρμογή, θα δύναται επίσης να επιστρέφει και άλλου είδους σχετιζόμενο περιεχόμενο, όπως οι κοντινότεροι αντιπρόσωποι του συγκεκριμένου μοντέλου, το μέσο κόστος συντήρησης, τα πιο συχνά προβλήματα όπως αναφέρονται από άλλους χρήστες, το κόστος των ανταλλακτικών, το δίκτυο σε άλλες πόλεις, πανευρωπαϊκά κλπ. Δηλαδή, με την εφαρμογή ευφυών τεχνικών διαχείρισης δεδομένων, επαυξάνεται επίσης και ο γνωσιακός χώρος – σφαίρα που αφορά την λύση του χρήστη (Personal Knowledge Sphere).

Είναι φανερό από τα ανωτέρω, ότι οι χρήστες δεν εκτελούν καθόλου αυτό το οποίο εκτελούσαν συνεχώς στο πρώτο σενάριο, δηλαδή έρευνες με βάση λέξεις κλειδιά, προκειμένου να ανιχνεύσουν δεδομένα και να τα εξορύξουν. Αντίθετα, χωρίς να γίνεται αντιληπτό από τον χρήστη το WEB 2.0 αποτελεί ένα περιβάλλον στο οποίο αλληλεπιδρούν συνεχώς ευφυείς διαμεσολαβητές (intelligent agents), προγράμματα αυτοματοποιημένης ανάκτησης δεδομένων διαδικτύου (web spiders και web robots), προκειμένου να ανευρίσκουν και ανταλλάσσουν πληροφορίες με δομημένο τρόπο όπως επί παραδείγματι σε μορφή RDF. Το πλεονέκτημα του semantic web τότε γίνεται άμεσα αντιληπτό, ειδικότερα όταν αναφερόμαστε στον τεράστιο όγκο πληροφοριών που υπάρχουν και διαρκώς αυξάνονται στο διαδίκτυο, ακόμα δε και πληροφορίας που παράγεται σε μορφή συνεχούς ροής (streaming).

Μεγάλες εταιρίες του χώρου όπως η IBM, Google, HP, Microsoft κλπ σχεδιάζουν εφαρμογές πάνω στα δεδομένα που προκύπτουν από το ερευνητικό πεδίο του semantic web. Η επιχειρηματική αγορά του αύριο διαφαίνεται να είναι με μεγάλη σιγουριά, ένα πεδίο στο παγκόσμιο ιστό, όπου εξειδικευμένες έξυπνες εφαρμογές, τύπου agent, θα ανταλλάσσουν επιχειρηματική πληροφόρηση, θα ρυθμίζουν τιμές αγοράς και πώλησης, ενσωματώνοντας επιχειρηματική νοημοσύνη. Οι έξυπνες αυτές εφαρμογές θα είναι και οι ίδιες χαρακτηρισμένες με metadata σημασιολογικά, προκειμένου να διαθέτουν τις επί συνδρομή υπηρεσίες τους μέσω σημασιολογικά χαρακτηρισμένων υπηρεσιών WEB (WEB services), σε ένα παγκοσμιοποιημένο περιβάλλον επιχειρείν.

Βασικό όμως στοιχείο πέραν από την περιγραφή με μεταδεδομένα των αρχικών δεδομένων που βρίσκονται ήδη αποθηκευμένα σε κατανεμημένες βάσεις δεδομένων, αποτελεί η ύπαρξη σημείων αναφορών των εννοιών που περιλαμβάνονται σε αυτές. Αυτά τα σημεία αναφορών είναι δομές οι οποίες θα γίνονται γνωστές (published) στους ευφυείς μέσους,

έτσι ώστε να υπάρχει κοινό εννοιολογικό υπόβαθρο σε επίπεδο ανταλλαγής μηνυμάτων. Στα πλαίσια της παρούσης εργασίας, εξετάζονται αναλυτικά τέτοιου είδους δομές, οι οποίες καλούνται οντολογίες σε σχέση τόσο με την απεικονιστική τους ικανότητα για τις έννοιες (concepts) όσο και για χρήση τους στην υποστήριξη εφαρμογών έξυπνου λογισμικού και δυναμικής διανομής περιεχομένου.

1.3. ΟΝΤΟΛΟΓΙΕΣ - ΓΕΝΙΚΑ

Οι οντολογίες είναι ένας μηχανισμός αναπαράστασης γνώσης, στην ορολογία των σύγχρονων πληροφοριακών συστημάτων. Οι οντολογίες πέρα από την οργάνωση μπορούν να λειτουργούν σαν μέσο επικοινωνίας σε σημασιολογικό επίπεδο (semantic level)[Error! Reference source not found.], για την αποτελεσματικότερη ανταλλαγή δεδομένων.

Το σημαντικότερο στοιχείο που πραγματικά έκανε αναγκαιότητα την ύπαρξη τέτοιων υποδομών για την περιγραφή δεδομένων, είναι η γενικευμένη χρήση του Internet από σχεδόν ολόκληρο τον κόσμο. Είναι σημαντικό να σημειωθεί, ότι η ύπαρξη των υποδομών – ιστοσελίδων, όπου περιγράφεται και καταχωρείται όλη η υπάρχουσα σήμερα γνώση, καθώς και σχετιζόμενη πληροφορία η οποία παρουσιάζεται σε άλλους τύπους, όπως για παράδειγμα κείμενα σε άλλες μορφές, αποσπάσματα ηλεκτρονικού ταχυδρομείου, εικόνες στατικές αλλά και κινούμενες (video), δημιουργούν ένα πολύπλοκο περιβάλλον απεικόνισης, καταγραφής, συσχέτισης και διανομής αυτής της πληροφορίας. Το διαδίκτυο πλέον, έχει κάνει επιτακτική την ανάγκη να ειδωθεί σαν ένα ενιαίο αλλά και ενοποιημένο πληροφοριακό σύστημα, όπου κάθε αντικείμενο πληροφορίας, θα είναι σε θέση να περιγράφεται και να οργανώνεται κατά τέτοιο τρόπο έτσι ώστε να είναι επαναχρησιμοποιήσιμο. Είναι πολύ σημαντικό να τονιστεί στο σημείο αυτό ότι τα αντικείμενα αυτά της πληροφορίας, τις πιο πολλές φορές έχουν την δυνατότητα να υπόκεινται σε πολλαπλές εξ' ίσου ορθές ερμηνείες – επεξηγήσεις, ανάλογα με το περιεχόμενο που εκάστοτε αναφέρονται. Επιπρόσθετα, η οργάνωση αυτή που επιβάλλεται να ολοκληρωθεί οφείλει, εκτός των κατάλληλων εννοιολογικών υποδομών να διαθέτει και υποδομές οι οποίες να συνεισφέρουν στην περιγραφή κατανεμημένης πληροφορίας (ομοειδούς – σχετιζόμενης) έτσι ώστε να διατίθεται στον χρήστη πληροφόρηση από πολλαπλές πηγές, αποθηκευμένη σε διαφορετικά συστήματα σε διαφορετικά επίπεδα (file systems, κτηρίων, πόλεων, κρατών κλπ) αλλά και παρομοίων οργανισμών που βρίσκονται σε διαφορετικά γεωγραφικά σημεία και διαθέτουν τέτοιου είδους πληροφορία. Η παραπάνω ανάγκη λοιπόν, οδηγεί στην οργάνωση των εννοιών μέσω εξειδικευμένων εργαλείων αναπαράστασης και καταγραφής αυτών και μέσω συγκεκριμένων γλωσσών – πρωτοκόλλων που θέτουν διεθνείς οργανισμοί (W3C), έτσι ώστε κάποια στιγμή να

οδηγηθούμε σε μία γενικευμένη απεικόνιση όλων των υπαρκτών εννοιών οι οποίες θα διαλειτουργούν ανεξάρτητα από τα περιβάλλοντα που χρησιμοποιήθηκαν για την καταγραφή, την απεικόνιση, την αποθήκευση και την διάθεσή τους. Η διάθεση της πληροφορίας υπό αυτές τις γενικευμένες συνθήκες δίνει πρόσθετη αξία στην πληροφορία και την περνά στο επόμενο επίπεδο χρησιμότητος, τη γνώση.

Στα πλαίσια, της παρούσης Δ.Δ., εξετάζονται αναλυτικά οι οντολογίες, οι τρόποι με τους οποίους αυτές μπορούν να συγκριθούν, να αξιολογηθούν σημασιολογικά και εν τέλει να διαλειτουργήσουν και να ανταλλάξουν παρόμοια στοιχεία μεταδεδομένων, με έξυπνο τρόπο.

1.3.1. Ορισμος

Τί είναι όμως οντολογία; Ο ορισμός προέρχεται από την κατηγοριοποίηση που πρότεινε ο Αριστοτέλης περί του Όντος, δηλαδή του καθενός υπάρχοντος πράγματος (αρχ. όν) και του ρήματος λέγω. Στα σύγχρονα πληροφοριακά συστήματα και στην επιστήμη των υπολογιστών, είναι μία έννοια η οποία χρησιμοποιείται για να περιγράψει την δομή και τις διαδικασίες ενός συγκεκριμένου χώρου με την ευρεία έννοια. Υπάρχουν τρεις ορισμοί οι οποίοι έχουν γίνει ευρέως αποδεκτοί:

(α) Οντολογία είναι μία επίσημη και επακριβής εξειδίκευση μίας έννοιας την οποία χρησιμοποιούν πέραν του ενός χρήστη (κοινή – shared έννοια) (Gruber T., 1995)

(β) Οντολογία είναι μία λογική θεωρία για μία έννοια ενός επίσημου λεξιλογίου (N. Guarino, 1998)

(γ) Οντολογία είναι μία επίσημη γλώσσα για την αναπαράσταση ενός χώρου γνώσης (knowledge domain) (G. L. Zuniga, 2001)

Γενικότερα, προσπαθώντας να συνδυαστούν τα παραπάνω, οντολογία είναι μία προσπάθεια για προτυποποίηση μίας έννοιας γενικότερα (standardization of conceptualization).

Μία εννοιοποίηση – έννοια (conceptualization) είναι η δημιουργία μίας επίσημης δομής η οποία απεικονίζει την πραγματικότητα, όπως αυτή γίνεται αντιληπτή και οργανώνεται από έναν ενδιάμεσο (agent), ο οποίος είναι ανεξάρτητος από το λεξιλόγιο και από την

συγκεκριμένη κατάσταση εφαρμογής. Υπάρχουν πέντε (5) κύρια τμήματα – συστατικά μίας οντολογίας:

- Η κλάση (class), αποκαλούμενη επίσης και έννοια
- Η ιδιότητα (–ες), (properties) επίσης καλούμενες slots ή roles που περιγράφουν διάφορες ιδιότητες και χαρακτηριστικά των κλάσεων
- Οι περιορισμοί (restrictions ή facets), που περιγράφουν τους περιορισμούς οι οποίοι ισχύουν για τις ιδιότητες των κλάσεων.
- Η εξειδίκευση – ύπαρξη (instance) μίας κλάσεως και
- Η σχέση (-εις) μεταξύ των κλάσεων.

1.3.2. ΠΕΡΙΒΑΛΛΟΝΤΑ ΑΠΕΙΚΟΝΙΣΗΣ ΟΝΤΟΛΟΓΙΩΝ

Για τον ορθό σχεδιασμό και απεικόνιση των οντολογιών, απαιτείται η χρήση εξειδικευμένων περιβαλλόντων. Για την πλήρη αξιοποίηση των δυνατοτήτων ανάκτησης γνώσης από αυτές, χρησιμοποιούνται εξειδικευμένες γλώσσες περιγραφής και ανάκτησης των χαρακτηριστικών τους (κόμβων, μονοπατιών, σχέσεων κλπ)

Παρακάτω παρουσιάζονται εν συντομία, τα δύο επικρατέστερα εν χρήσει περιβάλλοντα αυτή τη στιγμή απεικόνισης οντολογιών, το Protégé και το KAON Reverse. Επίσης, παρουσιάζονται εν συντομία και οι επικρατέστερες γλώσσες – τεχνολογίες περιγραφής και ανάκτησης δεδομένων (εννοιών) σε οντολογίες.

1.3.2.1. PROTÉGÉ

Το Protégé 2000 είναι ένας πολύ διαδεδομένος, λειτουργικός επεξεργαστής και συντάκτης οντολογιών που δημιουργήθηκε από το Πανεπιστήμιο Standford των Ηνωμένων Πολιτειών Αμερικής. Το εργαλείο αυτό χρησιμοποιείται από χιλιάδες χρήστες σε όλο τον κόσμο και εξελίσσεται συνεχώς. Παρέχει πολύ φιλικές γραφικές διεπαφές για τη συγγραφή και επεξεργασία οντολογιών σε OWL (Ontology WEB Language - http://www.w3.org/TR/owl-features/) και κανόνων σε SWRL (SWRL: A Semantic Web Rule Language on OWL - http://www.w3.org/Submission/SWRL/). Επίσης συνδέεται αυτόματα με reasoners (π.χ. Racer - http://protege.stanford.edu/plugins/owl/api/ReasonerAPIExamples.html). Γενικά έχουν αναπτυχθεί πολλά plug-ins που επεκτείνουν τη λειτουργία του. Μερικά από αυτά παρέχουν ευκολίες και βοηθήματα για την ανάπτυξη των οντολογιών (π.χ. οπτικοποίησή τους) ενώ άλλα παρέχουν διασύνδεση με εξωτερικά συστήματα (π.χ. Sesame, Jena2).

Ειδικότερα, το Protégé 2000 δίνει τη δυνατότητα στους χρήστες του να κατασκευάζουν οντολογίες για θεματικές περιοχές, να προσαρμόζουν στις δικές τους απαιτήσεις τις φόρμες εισαγωγής δεδομένων και να εισάγουν τα δεδομένα. Επίσης ξεχωρίζει, για ακόμα δύο χαρακτηριστικά του, σε σχέση με τα περισσότερα περιβάλλοντα επεξεργασίας οντολογιών: την εξελιξιμότητα και την επεκτασιμότητα του. Η βασισμένη σε συστατικά λογισμικού αρχιτεκτονική (component-based architecture) επιτρέπει στους προγραμματιστές να προσθέτουν λειτουργικότητα στο σύστημα απλά προσθέτοντας νέα κατάλληλα, αυτόνομα

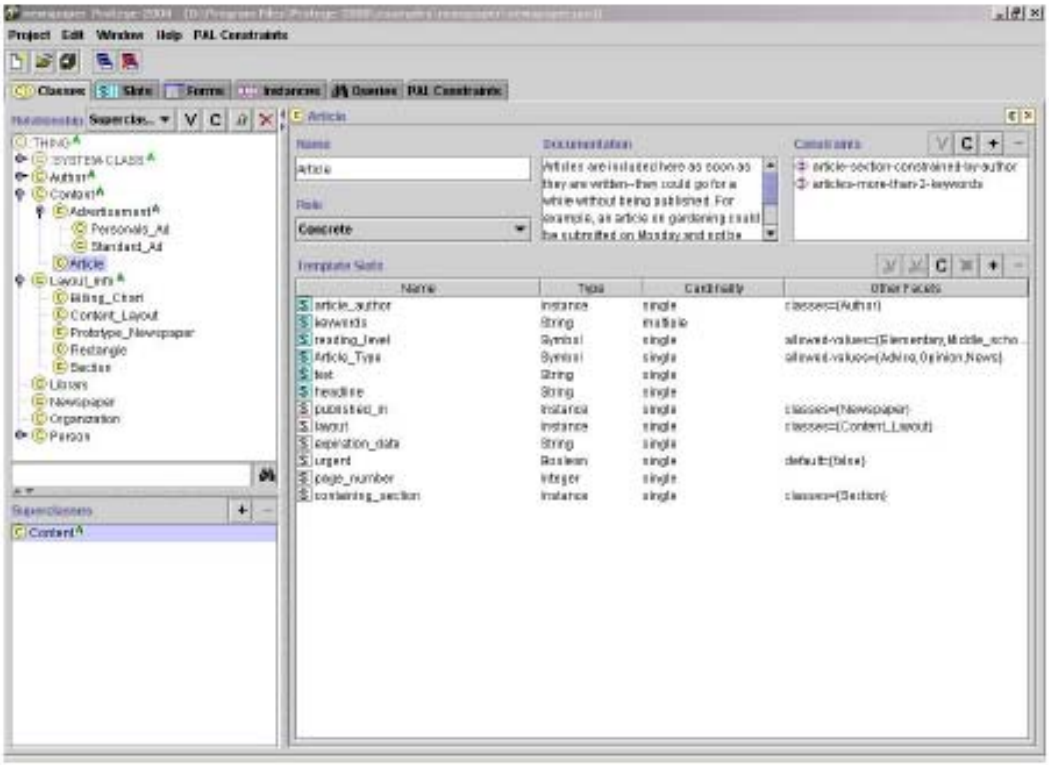
τμήματα κώδικα (plugins). Συνεπώς, μπορεί εύκολα να επεκταθεί ώστε να περιλαμβάνει γραφικά συστατικά όπως γραφικές παραστάσεις και πίνακες, όπως ήχο, εικόνες και video, καθώς και πολλαπλές μορφές αποθήκευσης όπως OWL, RDF, XML και HTML. Η αντίστοιχη βιβλιοθήκη του (Protégé Plugin Library) περιέχει συνεισφορές προγραμματιστών από ολόκληρο τον κόσμο.

Με το Protégé 2000 είναι δυνατή η ταυτόχρονη εργασία με κλάσεις οντολογιών και στιγμιότυπα (instances) αυτών των κλάσεων. Μια οντολογία είναι ένα μοντέλο ενός συγκεκριμένου πεδίου γνώσης, το οποίο στο Protégé αναπαριστάται ως ένα σύνολο κλάσεων με τις αντίστοιχες ιδιότητες (slots). Το Protégé υποστηρίζει δομή κλάσεων, με ιεραρχία αυτών και πολλαπλή κληρονομικότητα, μετα-κλάσεις και ιεραρχία τους κ.λ.π.

Τέλος, διευκολύνει την συμφωνία με το πρωτόκολλο Open Knowledge Base Connectivity (OKBC) για πρόσβαση σε συστήματα βάσεων γνώσεων και δίνει τη δυνατότητα να εκτελούνται εφαρμογές βασισμένες στα συστατικά του Protégé 2000 και ενσωματωμένες στο περιβάλλον του.

Περιβάλλον εργασίας – Βασικές λειτουργίες

Στην εικόνα 1 παρουσιάζεται το περιβάλλον εργασίας του Protégé-2000. Στο αριστερό τμήμα του κυρίου παραθύρου φαίνονται οι κατηγορίες και οι υποκατηγορίες (classes και subclasses) μιας οντολογίας και η ιεραρχική δομή τους, ενώ στο δεξί τμήμα μπορούμε να δούμε την περιγραφή της επιλεγμένης κατηγορίας (article), με το όνομά της, κάποιες πληροφορίες (documentation), ορισμένους περιορισμούς που πιθανόν να υπάρχουν, και τέλος τις ιδιότητες (slots) της κατηγορίας στο κάτω δεξί τμήμα.



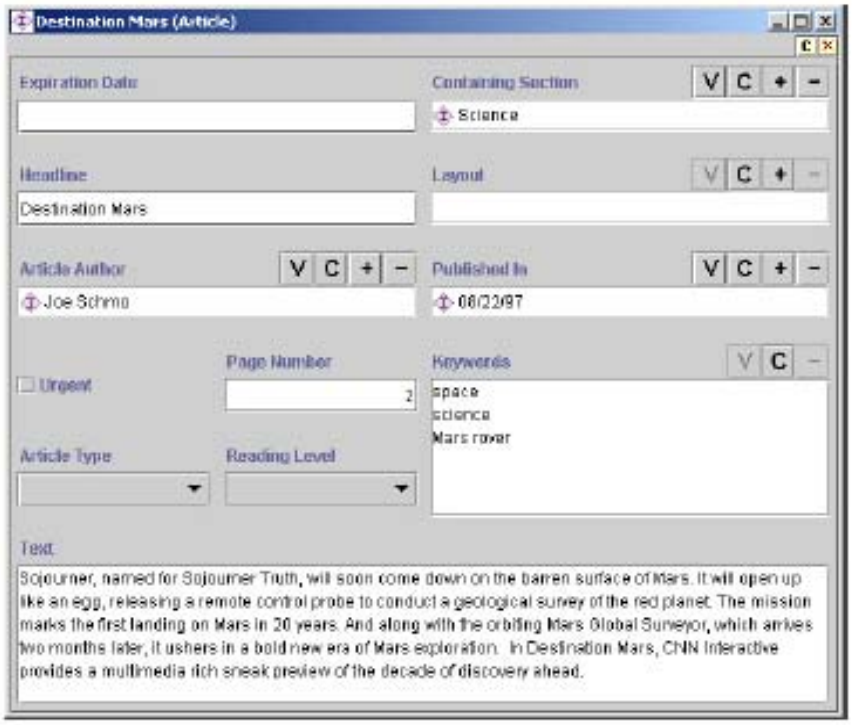
Εικόνα 1 - Το περιβάλλον εργασίας του Protégé-2000

Στην εικόνα 2 βλέπουμε την περιγραφή μιας εκ των ιδιοτήτων της συγκεκριμένης κατηγορίας, με στοιχεία όπως το είδος των τιμών που μπορεί να πάρει (ακέραιος αριθμός, δεκαδικός, σύμβολο κλπ) αλλά και το εύρος τους, και τις κατηγορίες με τις οποίες συνδέεται.



Εικόνα 2 - Περιγραφή μιας ιδιότητας της κατηγορίας article

Αυτές οι κατηγορίες περιέχουν αντικείμενα (instances), τα οποία ουσιαστικά αποτελούν τις εγγραφές στη συγκεκριμένη βάση δεδομένων. Ένα παράδειγμα μιας τέτοιας εγγραφής της κατηγορίας article βλέπουμε στην εικόνα 3.



Εικόνα 3 - Αντικείμενο (instance) της κατηγορίας article

Στο παράθυρο αυτό παρατηρούνται οι τιμές όλων των ιδιοτήτων της κατηγορίας article για το συγκεκριμένο αντικείμενο. Επίσης η εισαγωγή πληροφορίας στο Protégé όσον αφορά τις εγγραφές πραγματοποιείται με μια σειρά από φόρμες όπως αυτή που φαίνεται στην παραπάνω εικόνα, οι οποίες μπορούν να σχεδιαστούν και να διαφοροποιηθούν κατά βούληση του χρήστη.

Πλεονεκτήματα του Protégé-2000

Ο συγκεκριμένος ontology editor έχει ορισμένα χαρακτηριστικά τα οποία τον κατατάσσουν ανάμεσα στις καλύτερες εφαρμογές του είδους του σήμερα. Το πιο βασικό από αυτά και αυτό που κάνει τη συγκεκριμένη εφαρμογή να ξεχωρίζει σε σχέση με άλλες είναι η μεγάλη επεκτασιμότητα που έχει, η οποία απορρέει από την αρχιτεκτονική με την οποία έχει κατασκευαστεί. Λέγοντας επεκτασιμότητα εννοούμε τη δημιουργία επιπρόσθετων λειτουργιών οι οποίες αναπτύσσονται ξεχωριστά από την εφαρμογή, είτε από την ίδια ομάδα προγραμματιστών είτε – κυρίως – από εξειδικευμένους χρήστες της. Αυτές οι επιπλέον λειτουργίες κυκλοφορούν υπό μορφή μικρών προγραμμάτων που ονομάζονται

plug-ins, και ο κάθε χρήστης του Protégé μπορεί να εγκαταστήσει οποιεσδήποτε από αυτές θεωρεί ότι συμβάλλουν στην επίτευξη του σκοπού της οντολογίας του.

Η ποικιλία των plug-ins που κυκλοφορούν είναι πολύ μεγάλη: υπάρχουν πολλά που αφορούν την οπτική αναπαράσταση της οντολογίας με διαφορετικές μορφές, άλλα που αφορούν χρήση λειτουργιών βάσης δεδομένων στην οντολογία (ερωτήσεις, σύνθετοι περιορισμοί κλπ), και άλλα που χρησιμεύουν στην αυτόματη εισαγωγή γνώσης στη βάση από ήδη υπάρχουσες οντολογίες ή ακόμα και άλλες μορφές δεδομένων, όπως για παράδειγμα απλά αρχεία κειμένου. Επίσης υπάρχουν ορισμένα plug-ins που επιτρέπουν την εισαγωγή πολυμέσων σε μια οντολογία, όπως αρχείων εικόνας, ήχου, ακόμα και βίντεο. Τέλος, όλο και περισσότερα γίνονται τα plug-ins που επιτρέπουν την αλληλεπίδραση του Protégé και των οντολογιών που αυτό δημιουργεί με παρόμοιες εφαρμογές και όχι μόνο, όπως για παράδειγμα η γλώσσα XML, αλλά και η CLIPS που χρησιμοποιείται στην ανάπτυξη έμπειρων συστημάτων.

Τέλος, μια από τις πιο χρήσιμες λειτουργίες του Protégé αποτελεί και η σύνδεση που μπορεί να παρέχει με το διαδίκτυο. Πιο συγκεκριμένα, έχει τη δυνατότητα άμεσης εξαγωγής οποιασδήποτε οντολογίας σε μορφή html η οποία μπορεί έτσι να δημοσιευθεί στο διαδίκτυο χωρίς περαιτέρω επεξεργασία. Ακόμα, υπάρχουν και αναπτύσσονται ορισμένα εργαλεία που χρησιμοποιούν το διαδίκτυο ως πηγή για εύρεση και εισαγωγή γνώσης σε μια οντολογία, χαρακτηριστικό που αποτελεί ένα ακόμα βήμα στην πορεία για μια μεγάλη, ιδανική, παγκόσμια βάση γνώσης.

1.3.2.2. KAON Reverse

Το KAON Reverse είναι ένα ημι-αυτόματο εργαλείο, με γραφικό για το χρήστη περιβάλλον, το οποίο κατασκευάστηκε από το Πανεπιστήμιο της Karlsruhe της Γερμανίας με σκοπό τη μεταφορά της πληροφορίας που είναι αποθηκευμένη σε μια βάση δεδομένων σε μια οντολογία. Η οντολογία δεν δημιουργείται αυτόματα από τη βάση δεδομένων, αλλά πρέπει να υπάρχει πριν αρχίσει η διαδικασία της αντιστοίχησης.

Η διαδικασία της αντιστοίχησης που ακολουθείται από το συγκεκριμένο εργαλείο περιλαμβάνει τα παρακάτω βήματα:

1. *Εξαγωγή των μεταδεδομένων (σχέσεις, γνωρίσματα, πρωτεύοντα κλειδιά, ξένα κλειδιά κτλ) από το σχεσιακό σχήμα με χρήση μιας τεχνικής reverse engineering.*
2. *Ανάλυση της πληροφορίας που αντλήθηκε στο προηγούμενο βήμα για την αντιστοίχηση των στοιχείων της βάσης στα στοιχεία της οντολογίας με τη βοήθεια κανόνων αντιστοίχησης.*
3. *Έλεγχος αν έχουν αντιστοιχηθεί όλα τα στοιχεία του σχεσιακού σχήματος στα στοιχεία της οντολογίας.*
4. *Μεταφορά των δεδομένων από τη βάση στην οντολογία σύμφωνα με τους κανόνες αντιστοίχησης που ορίστηκαν στα προηγούμενα βήματα.*

Η διαδικασία της αντιστοίχησης χωρίζεται σε δύο κύρια μέρη: στην αντιστοίχηση πινάκων (Table Mapping) και στην αντιστοίχιση στηλών (Column Mapping). Πιο συγκεκριμένα, οι πίνακες του σχεσιακού σχήματος της βάσης δεδομένων αντιστοιχίζονται σε κλάσεις της οντολογίας και οι στήλες των πινάκων αυτών σε datatype properties (attributes) ή σε object properties (relations) της οντολογίας. Το KAON Reverse δίνει στους χρήστες τη δυνατότητα να επιλέξουν ανάμεσα σε απλή ή σύνθετη αντιστοίχηση. Στην πρώτη περίπτωση όλες οι πλειάδες του πίνακα «μετατρέπονται» σε στιγμιότυπα (instances) της οντολογίας, ενώ στη δεύτερη περίπτωση η δημιουργία των στιγμιοτύπων περιορίζεται από SQL επερωτήσεις.

Αξίζει να σημειωθεί ότι το KAON Reverse προσφέρει μεθόδους εύρεσης της λεξικογραφικής ομοιότητας προκειμένου να ανακαλύψει πιθανή αντιστοίχηση ανάμεσα στα στοιχεία των σχημάτων. Η οντολογία που χρησιμοποιείται από το συγκεκριμένο εργαλείο είναι υλοποιημένη σε RDF(S) και ως εκ τούτου δεν προσφέρει δυνατότητα αναπαράστασης σύνθετων οντολογιών. Συγκεκριμένα, το KAON Reverse δεν υποστηρίζει οντολογίες που περιέχουν αξιώματα (axiomatized ontologies) αλλά και relations που το domain τους είναι ένα σύνολο από κλάσεις.

1.3.3. Γλωσσες Περιγραφης Οντολογιων

1.3.3.1. RDF – RDF(S)

Το *Πλαίσιο Περιγραφής Πόρων,* γνωστό ως RDF (Resource Description Framework) συνιστά μια γλώσσα, η οποία σχεδιάστηκε με στόχο την αναπαράσταση πληροφοριών για πόρους στον Παγκόσμιο Ιστό. Στην πραγματικότητα το RDF αποτελεί ένα σύνολο προδιαγραφών το οποίο σχεδιάστηκε από το World Wide Web Consortium (W3C) ως ένα μοντέλο δεδομένων για μεταδεδομένα. Στην πράξη χρησιμοποιείται σαν μια γενική μέθοδος για την περιγραφή πόρων που συναντώνται στον Παγκόσμιο Ιστό. Ως γλώσσα το RDF μπορεί να αναπαρασταθεί με ποικίλους τρόπους, εκ των οποίων ο πιο κοινός είναι η XML.

Βασικός στόχος του RDF είναι η αναπαράσταση μεταδεδομένων σχετικών με πόρους του Παγκόσμιου Ιστού, όπως για παράδειγμα ο τίτλος, ο συγγραφέας ή η ημερομηνία τροποποίησης μιας ιστοσελίδας, διάφορες πληροφορίες για ένα έγγραφο ή έναν διαμοιραζόμενο πόρο.

Ένα βασικό χαρακτηριστικό του RDF είναι ότι η δομή του, όπως θα δούμε στη συνέχεια, είναι κατάλληλη για επεξεργασία από εφαρμογές κατασκευασμένες να διαλειτουργούν με τη χρήση μεταπληροφορίας. Θα ήταν ορθότερο να πούμε ότι το RDF είναι σχεδιασμένο όχι τόσο για να μπορεί να αναγνωρισθεί, αναγνωσθεί και να κατανοηθεί από ένα χρήστη, αλλά το ίδιο, παρέχοντας ένα κοινό πλαίσιο αναπαράστασης της πληροφορίας αποτελεί ιδανικό πρότυπο για την ανταλλαγή αυτής της πληροφορίας μεταξύ εφαρμογών, που διαχειρίζονται περιεχόμενο σημασιολογικά.

Η βασική ιδέα που διατρέχει το RDF συνίσταται στην ταυτοποίηση αντικειμένων με τη χρήση διευθύνσεων URI (Uniform Resource Identifier), και την περιγραφή πόρων μέσω Ιδιοτήτων (Properties) και Τιμών Ιδιοτήτων (Property Values).

Ας πάρουμε ως παράδειγμα μια απλή δήλωση: **http://www.example.org/index.html** has a **creator** whose name is **John Smith.**

Στην παραπάνω πρόταση το URL ταυτοποιεί ένα αντικείμενο (μια ιστοσελίδα), η λέξη «creator» προσδιορίζει μιαν ιδιότητα και οι λέξεις «John Smith» αποτελούν την τιμή της ιδιότητας.

Το RDF κωδικοποιεί προτάσεις όπως η παραπάνω. Επομένως, τα πράγματα που μπορούμε να περιγράψουμε με το RDF έχουν ιδιότητες με τιμές και κάθε αντικείμενο που περιγράφουμε έχει μια ταυτότητα.

Χρησιμοποιώντας την κατάλληλη ορολογία, μπορούμε να πούμε ότι στην παραπάνω πρόταση, το μέρος που ταυτοποιεί το πράγμα για το οποίο μιλάμε, αποτελεί το **Υποκείμενο (subject)** της πρότασης. Το μέρος που χαρακτηρίζει την ιδιότητα του υποκειμένου αποτελεί το **Κατηγόρημα (predicate)**. Τέλος, το μέρος που προσδιορίζει την τιμή αυτής της ιδιότητας, αποτελεί το **Αντικείμενο (object)**.

Όπως είπαμε προηγουμένως, το RDF σχεδιάστηκε ώστε τις δηλώσεις τις οποίες κωδικοποιεί, να μπορούν να τις επεξεργαστούν εφαρμογές.

Επομένως, είναι απαραίτητο να προσδιοριστούν 2 στοιχεία που θα διευκολύνουν αυτή την κωδικοποίηση:

1. Ένα σύστημα αναγνωριστικών, τα οποία μπορεί να επεξεργαστεί ένα πρόγραμμα και χρησιμοποιούνται για την ταυτοποίηση ενός υποκειμένου, κατηγορήματος, ιδιότητας. Το web παρέχει ένα τέτοιο σύστημα: το URI (Uniform Resource Identifier).
2. Μια γλώσσα αναπαράστασης των δηλώσεων RDF και επικοινωνίας μεταξύ εφαρμογών. Το RDF χρησιμοποιεί τη γλώσσα XML για αυτό το σκοπό.

Το RDF αναπαριστά προτάσεις με τη μορφή κόμβων και συνδέσεων αυτών των κόμβων σε ένα γράφο. Ας πάρουμε ως παράδειγμα την πρόταση:

http://www.example.org/index.html has a **creator** whose name is **John Smith.**

Η παραπάνω πρόταση θα μπορούσε να αναπαρασταθεί έχοντας:

- Υποκείμενο http://www.example.org/index.html
- Κατηγόρημα http://purl.org/dc/elements/1.1/creator
- Αντικείμενο http://www.example.org/staffid/85740

Υπό μορφή γράφου θα είχαμε:

- Ένα κόμβο για το υποκείμενο
- Ένα κόμβο για το αντικείμενο
- Μια σύνδεση για το κατηγόρημα, με κατεύθυνση από το υποκείμενο στο αντικείμενο

Μια δημοφιλής και φιλική και προς το χρήστη σύνταξη για τις δηλώσεις RDF είναι οι τριπλέτα (triple). Η παραπάνω πρόταση μπορεί να αναπαρασταθεί ως τριπλέτα:

<http://www.example.org/index.html>

<http://purl.org/dc/elements/1.1/creator>

<http://www.example.org/staffid/85740>

Το συντακτικό XML που θα χρησιμοποιούσαμε για να εκφράσουμε την παραπάνω πρόταση θα έχει την εξής μορφή:

```
<?xml version="1.0"?>
 <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
          xmlns:exterms="http://www.example.org/terms/">
  <rdf:Description rdf:about="http://www.example.org/index.html">
    <exterms:creator>John Smith</exterms:creator>
  </rdf:Description>
 </rdf:RDF>
```

1.3.3.2. DAML + OIL

Η DAML+OIL είναι μια γλώσσα για την κατασκευή οντολογιών στο πλαίσιο της προσπάθειας για την ανάπτυξη του Σημασιολογικού Ιστού. Πρόκειται δηλαδή για μια γλώσσα που αποσκοπεί στην περιγραφή πόρων που βρίσκονται στον Παγκόσμιο Ιστό (Web). Ως γλώσσα βασίζεται σε πρώιμα πρότυπα του World Wide Web Consortium, όπως είναι τα RDF και RDFS, και επεκτείνει τα τελευταία με πιο εκφραστικά στοιχεία.

Η DAML+OIL με τον οργανισμό DAML.ORG υπεύθυνου για την ανάπτυξη και βελτίωση της, αποτελεί συνδυασμό των γλωσσών DAML και OIL και έχει καθαρή και καλώς ορισμένη σημασιολογία, με δυνατότητες υποστήριξης δημιουργίας οντολογιών και περιγραφής αυτών για περαιτέρω επεξεργασία από διαδικασίες μηχανικής κατανόησης (machine understandable). (http://www.daml.org/about.html)

Μια βάση γνώσης στη DAML+OIL αποτελεί μια συλλογή από τριάδες RDF. Η DAML+OIL προσδίδει συγκεκριμένο νόημα σε τριάδες που χρησιμοποιούν το λεξιλόγιο της DAML+OIL. Μια οντολογία DAML+OIL αποτελείται από πολλά μέρη, ορισμένα προαιρετικά και άλλα επαναλαμβανόμενα. Συγκεκριμένα, μια οντολογία DAML+OIL αποτελείται από >=0 headers, ακολουθούμενα από >=0 στοιχεία κλάσεων, στοιχεία ιδιοτήτων και περιπτώσεις κλάσεων.

Το παράδειγμα που παρουσιάζεται παρακάτω, αρχίζει με ένα αρχικό RDF tag το οποίο περιλαμβάνει δηλώσεις namespace:

```
<rdf:RDF
```

```
 xmlns:rdf ="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
 xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
 xmlns:xsd ="http://www.w3.org/2000/10/XMLSchema#"
 xmlns:daml="http://www.daml.org/2001/03/daml+oil#"
 xmlns:dex ="http://www.daml.org/2001/03/daml+oil-ex#"
 xmlns:exd ="http://www.daml.org/2001/03/daml+oil-ex-dt#"
 xmlns     ="http://www.daml.org/2001/03/daml+oil-ex#"
>
```

Ένα στοιχείο daml:ontology περιλαμβάνει πληροφορίες έκδοσης και στοιχεία εισαγωγής:

```
<Ontology rdf:about="">
 <versionInfo>$Id: NOTE-daml+oil-reference-20011218.html,v 1.6 2001/12/18 22:12:09 connolly Exp $</versionInfo>
 <rdfs:comment>An example ontology</rdfs:comment>
 <imports rdf:resource="http://www.w3.org/2001/10/daml+oil"/>
</Ontology>
```

Η DAML+OIL χωρίζει το οντολογικό σύμπαν σε 2 μέρη. Το ένα καλείται Datatype Domain και αποτελείται από τις τιμές που ανήκουν στους τύπους δεδομένων του XML Schema. Το άλλο μέρος καλείται Object Domain και αποτελείται από αντικείμενα που είναι μέλη των κλάσεων που περιγράφονται από την DAML+OIL.

Η DAML+OIL αφορά κυρίως τη κατασκευή κλάσεων οι οποίες περιγράφουν το Object Domain. Αυτές οι κλάσεις καλούνται κλάσεις αντικειμένων και στοιχεία τύπου daml:Class, υποκλάσης του στοιχείου rdfs:Class.

Τα στοιχεία κλάσης, daml:Class, περιέχει μέρος του ορισμού μιας κλάσης αντικειμένου.

Ένα στοιχείο ιδιότητας, rdf:Property, αναφέρεται σε ένα όνομα ιδιότητας (ένα URI).

Οι περιπτώσεις τόσο των κλάσεων, όσο και των ιδιοτήτων γράφονται με το συντακτικό RDF και RDFS:

```
<continent rdf:ID="Asia"/>
<rdf:Description rdf:ID="Asia">
 <rdf:type>
  <rdfs:Class rdf:about="#continent"/>
 </rdf:type>
</rdf:Description>
<rdf:Description rdf:ID="India">
 <is_part_of rdf:resource="#Asia"/>
</rdf:Description>
```

Οι τιμές τύπων δεδομένων γράφονται στο RDF συντακτικό, αλλά με ιδιαίτερη σημασιολογία στη DAML+OIL.

1.3.3.3. OWL

Η OWL (Web Ontology Language) αποτελεί στην ουσία, ένα σύνολο γλωσσών αναπαράστασης γνώσης με στόχο την κατασκευή και αναπαράσταση οντολογιών. Η OWL υποστηρίζεται από το World Wide Web Consortium και θεωρείται μια από τις θεμελιώδεις τεχνολογίες για την εξέλιξη του Σημασιολογικού Ιστού (Semantic Web).

Η γλώσσα OWL ως προϊόν έρευνας συνιστά μια αναθεώρηση της οντολογικής γλώσσας ιστού DAML+OIL και έχει σχεδιαστεί με στόχο τη χρήση της από εφαρμογές για τις οποίες απαιτείται επεξεργασία του περιεχομένου της πληροφορίας, παρά η απλή παρουσίαση του περιεχομένου στους χρήστες. Η OWL διευκολύνει και βελτιώνει την επεξεργασία του περιεχομένου που βρίσκεται στο Web, σε σχέση με τις τεχνολογίες XML, RDF και RDFS, καθώς παρέχει επιπρόσθετο λεξιλόγιο σε συνδυασμό με τυπική σημασιολογία. Η OWL μπορεί να χρησιμοποιηθεί για να αναπαραστήσει τη σημασία όρων μέσα σε λεξιλόγια καθώς και τις σχέσεις μεταξύ αυτών. Αυτού του είδους η αναπαράσταση όρων και σχέσεων καλείται οντολογία.

Η OWL παρέχει 3 διαφορετικές γλώσσες, με διαφορετικό βαθμό εκφραστικότητας, οι οποίες έχουν σχεδιαστεί για συγκεκριμένες ανάγκες υλοποίησης:

- Η OWL Lite υποστηρίζει τους χρήστες οι οποίοι απαιτούν κυρίως μια ιεραρχία ταξινόμησης και απλούς περιορισμούς.
- Η OWL DL υποστηρίζει εκείνους τους χρήστες οι οποίοι επιθυμούν μέγιστη εκφραστικότητα παράλληλα με υπολογιστική πληρότητα και δυνατότητα λήψης αποφάσεων. Η OWL DL αντιστοιχεί περισσότερο από τις υπόλοιπες στην περιγραφική λογική.
- Η OWL Full απευθύνεται σε χρήστες οι οποίοι επιθυμούν μέγιστη εκφραστικότητα παράλληλα με την συντακτική ευρύτητα του πλαισίου RDF, χωρίς υπολογιστικές εγγυήσεις.

Η OWL παρέχει τη δυνατότητα για τη δημιουργία κλάσεων και ιδιοτήτων και τον ορισμό περιπτώσεων κλάσεων και των λειτουργιών τους.

- Κλάσεις:

  Κλάσεις οριζόμενες από το χρήστη, οι οποίες αποτελούν υπό-κλάσεις της ρίζας owl:Thing. Μια κλάση μπορεί να περιέχει άτομα (individuals), τα οποία αποτελούν

περιπτώσεις της κλάσης και άλλων υπό-κλάσεων. Για παράδειγμα, η κλάση Πελάτης θα μπορούσε να είναι υποκλάση της owl:Thing, ενώ οι κλάσεις Γυναίκα και Άντρας υποκλάσεις της κλάσης Πελάτης

- Ιδιότητες:

Μια ιδιότητα είναι μια δυαδική σχέση η οποία ορίζει τα χαρακτηριστικά μιας κλάσης. Οι ιδιότητες χαρακτηρίζουν περιπτώσεις κλάσεων και συχνά λειτουργούν ως συνδέσεις σε άλλες περιπτώσεις κλάσεων. Υπάρχουν 2 είδη ιδιοτήτων: ιδιότητες τύπου (datatype) και ιδιότητες αντικειμένου (object properties). Οι πρώτες αποτελούν σχέσεις μεταξύ περιπτώσεων κλάσεων και ορισμών RDF ή τύπων δεδομένων ενός XML Schema. Οι δεύτερες είναι σχέσεις μεταξύ περιπτώσεων 2 κλάσεων.

- Περιπτώσεις:

Πρόκειται για ατομικές οντότητες οι οποίες ανήκουν στις κλάσεις που έχουν οριστεί. Μια κλάση μπορεί να έχει οσοδήποτε αριθμό περιπτώσεων. Οι περιπτώσεις χρησιμοποιούνται για τον ορισμό της σχέσης μεταξύ διαφορετικών κλάσεων.

- Λειτουργίες:

Η OWL υποστηρίζει ποικίλες λειτουργίες επί των κλάσεων, όπως είναι η ένωση, η τομή, το συμπλήρωμα. Επί πλέον, επιτρέπει απαρίθμηση κλάσεων, υποστήριξη πλήθους (cardinality) και αποσύνδεσης εννοιών (concept disjointness).

Η OWL βασίζεται στην «υπόθεση του ανοιχτού κόσμου», σε αντίθεση με την Prolog ή την SQL, οι οποίες υιοθετούν την «υπόθεση του κλειστού κόσμου». Υπό τους όρους της αυτής, αν μια δήλωση δεν μπορεί να αποδειχθεί αληθής στο πλαίσιο της τρέχουσας γνώσης - πεδίου, δεν είμαστε σε θέση να συμπεράνουμε ότι η δήλωση είναι ψευδής.

1.3.4. Γλωσσες Αναζητησης Οντολογιων

1.3.4.1. SPARQL

Η γλώσσα SPARQL (http://www.w3.org/TR/rdf-sparql-query/), σύμφωνα με τον οργανισμό W3C (World Wide Web Concortium), που είναι υπεύθυνος για την δημιουργία προδιαγραφών και τεχνολογιών που σχετίζονται με το Web, ορίζει

(http://www.w3.org/TR/rdf-sparql-query/), ότι είναι μία γλώσσα η οποία μπορεί να χρησιμοποιηθεί για την έκφραση ερωτημάτων με την μορφή SQL queries προς αποκλίνουσες (diverse) πηγές δεδομένων οι οποίες είτε περιγράφονται τοπικά μέσω δομών RDF, είτε μπορούν να ειδωθούν μέσω middleware με τη χρήση RDF. Η γλώσσα SPARQL υποστηρίζει εγγενώς την αναζήτηση σε δομές γράφων.

Αναζήτηση RDF δεδομένων με την χρήση της SPARQL

Όλο και περισσότερα δεδομένα αποθηκεύονται πλέον σε μορφή RDF όπως παραδείγματος χάριν τα Really Simple Syndication - RSS, και έτσι έχει δημιουργηθεί η ανάγκη για να βρεθεί ένας απλός τρόπος για να βρίσκονται συγκεκριμένες πληροφορίες.

Η SPARQL αποτελεί μια ισχυρή γλώσσα αναζήτησης και γεμίζει αυτό το κενό, κάνοντας εύκολο να βρει κανείς τα δεδομένα που χρειάζεται στην RDF δομή. Παρακάτω περιγράφονται τα χαρακτηριστικά της SPARQL και η διασύνδεσή της με τη γλώσσα RDF.

Όπως είδαμε παραπάνω, η γλώσσα RDF δίνει τη δυνατότητα στα δεδομένα να αποκεντρωθούν και να κατανεμηθούν. Τα RDF μοντέλα μπορεί να συγχωνευτούν εύκολα, και μπορούν να ανταλλαχθούν σε RDF σειρές πολύ απλά πάνω από το HTTP.

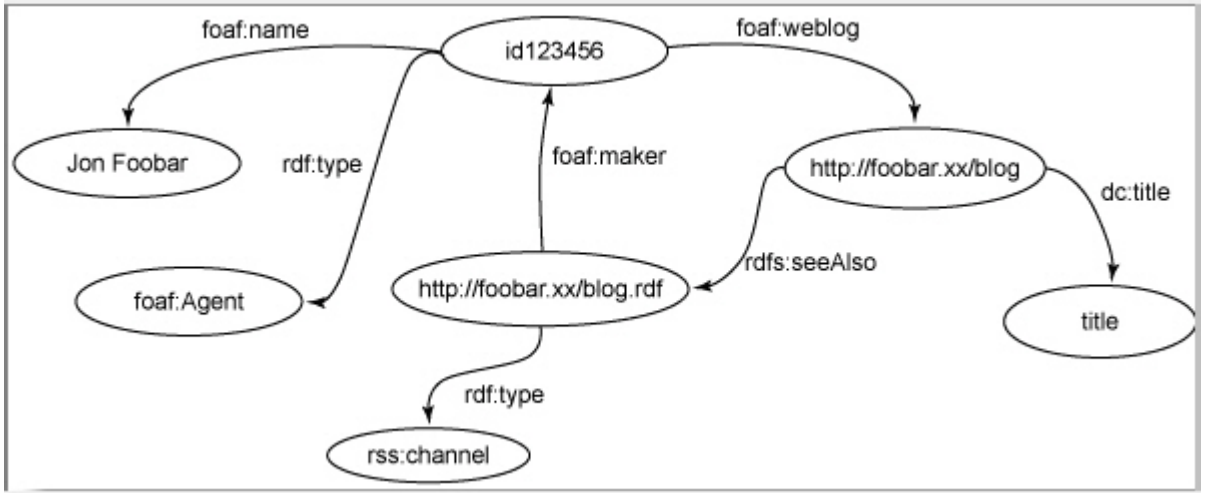
Κατ' αυτόν τον τρόπο οι εφαρμογές μπορεί να συνδεθούν με πολλαπλές RDF πηγές δεδομένων στο Web.

Πως όμως κάποιος μπορεί να βρει και να επεξεργαστεί τα δεδομένα που χρειάζεται μέσα στους RDF γράφους (οντολογίες); Το πρωτόκολλο SPARQL και η RDF γλώσσα αναζήτησης (SPARQL) είναι επί του παρόντος σε συζήτηση ως «W3C Working Draft», με σκοπό την μετεξέλιξή του σε standard W3C. Η SPARQL αναπτύσσεται πάνω σε προηγούμενες RDF γλώσσες αναζήτησης, όπως οι rdfDB, RDQL και SeRQL γλώσσες και έχει αρκετά χαρακτηριστικά δικά της. Σε αυτή την ενότητα, θα χρησιμοποιήσουμε 3 περιπτώσεις RDF γράφων που καθοδηγούν τους PlanetRDF -- FOAF (Frien of a Friend) γράφους, οι οποίοι περιγράφουν συνεργάτες, τα RSS feeds, και τον γράφο των blogger – για να αναπαραστήσουμε κάποια από τα ενδιαφέροντα πράγματα που μπορούν να κάνουν οι ερωτήσεις αναζήτησης της SPARQL με τα δεδομένα. Υπάρχουν υλοποιήσεις της SPARQL για μια ποικιλία πλατφορμών και γλωσσών.

Με βάση ότι στην ανάπτυξη της εφαρμογής χρησιμοποιήθηκε η γλώσσα αναζήτησης Jena, θα εστιαστεί το ενδιαφέρον στο Jena Semantic Web Toolkit για πλατφόρμα Java.

Δομή SPARQL ερώτησης αναζήτησης

Στο παράδειγμα χρησιμοποιείται από το απλό μοντέλο του bloggers.rdf που επιτρέπει σε κάποιον με βάση τα λεξικά FOAF και Dublin Core, να δώσει όνομα, τίτλο στο blog, μια URL, και μια περιγραφή για το RSS feed για κάθε ένα συνεργάτη του blog. Το σχήμα δείχνει τη βασική δομή του γράφου για ένα απλό συνεργάτη. Το πλήρες μοντέλο απλά επαναλαμβάνει αυτή τη δομή για κάθε ένα blog που συγκεντρώνουμε.



Εικόνα 1: Βασική δομή γράφου για ένα απλό συνεργάτη στο bloggers.rdf

Παρατηρούμε μια αρχική ερώτηση SPARQL στο μοντέλο των blogger. Η ερώτηση εκφράζει, «Βρες την URL του blog που με το πρόσωπο που ονομάζεται Jon Foobar» και η υλοποίηση της φαίνεται στην παρακάτω λίστα κώδικα:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?url
FROM     <bloggers.rdf>
WHERE  {
    ?contributor foaf:name "Jon Foobar" .
```

```
    ?contributor foaf:weblog ?url .
}
```

Η ερώτηση SPARQL για να βρούμε την URL ενός blog συνεργάτη

Χρησιμοποιώντας την SPARQL με το εργαλείο Jena

Η υποστήριξη της SPARQL στο εργαλείο Jena είναι επί του παρόντος διαθέσιμη μέσω ενός συστατικού λογισμικού που λέγεται ARQ. Εκτός από την υποστήριξη της υλοποίησης της SPARQL, η μηχανή αναζήτησης ARQ μπορεί επιπλέον να αναλύσει ερωτήσεις που εκφράζονται σε γλώσσα RDQL ή στην δική της εσωτερική γλώσσα αναζήτησης. Η ARQ είναι υπό ανάπτυξη, και δεν είναι ακόμα κομμάτι της βασικής διανομής του εργαλείου Jena. Όμως, είναι διαθέσιμη είτε από το CVS repository της Jena ή σαν μια αυτόνομη εφαρμογή.

```
$ export ARQROOT=~/ARQ-0.9.5
$ chmod +rx $ARQROOT/bin/*
$ export PATH=$PATH:$ARQROOT/bin
$ sparql
         Usage: [--data URL] [exprString | --query file]
```

Παράμετροι περιβάλλοντος για την χρήση της Jena ARQ

Εκτέλεση SPARQL ερωτήσεων αναζήτησης με το Jena API

Οι εφαρμογές Java μπορούν να καλέσουν απευθείας τις δυνατότητες της Jena SPARQL. Οι ερωτήσεις SPARQL δημιουργούνται και εκτελούνται με την Jena μέσω κλάσεων στο πακέτο com.hp.hpl.jena.query. Η χρήση της κλάσης QueryFactory είναι η πιο απλή προσέγγιση.

Το επόμενο βήμα είναι η δημιουργία ενός instance της κλάσης QueryExecution, μια κλάση που αναπαριστά μια απλή εκτέλεση ερώτησης αναζήτησης. Υπάρχουν αρκετές διαφορετικές μέθοδοι στην QueryExecution, η κάθε μια εκτελεί διαφορετικό τύπο ερώτησης.

Η παρακάτω λίστα κώδικα δείχνει ένα απλό τρόπο για να βάλει κανείς όλα τα βήματα μαζί. Εκτελεί μια ερώτηση στο bloggers.rdf, και βγάζει τα αποτελέσματα στην κονσόλα.

```
// Open the bloggers RDF graph from the filesystem
InputStream in = new FileInputStream(new File("bloggers.rdf"));

// Create an empty in-memory model and populate it from the graph
Model model = ModelFactory.createMemModelMaker().createModel();
model.read(in,null); // null base URI, since model URIs are absolute
in.close();

// Create a new query
String queryString =
          "PREFIX foaf: <http://xmlns.com/foaf/0.1/> " +
          "SELECT ?url " +
          "WHERE {" +
          "      ?contributor foaf:name \"Jon Foobar\" . " +
          "      ?contributor foaf:weblog ?url . " +
          "      }";

Query query = QueryFactory.create(queryString);

// Execute the query and obtain results
QueryExecution qe = QueryExecutionFactory.create(query, model);
ResultSet results = qe.execSelect();

// Output query results
ResultSetFormatter.out(System.out, results, query);

// Important - free up resources used running the query
qe.close();
```

Εκτελώντας μια ερώτηση χρησιμοποιώντας το Jena API

Γράφοντας πιο πολύπλοκες ερωτήσεις

Το RDF μοντέλο χρησιμοποιείται συχνά για να αναπαραστήσει ημι-δομημένα δεδομένα. Αυτό σημαίνει ότι δύο κόμβοι του ίδιου τύπου σε ένα μοντέλο μπορεί να έχει διαφορετικά είδη ιδιοτήτων. Για παράδειγμα, μια περιγραφή ενός προσώπου σε ένα FOAF μοντέλο μπορεί να αποτελείται από μόνο μια e-mail διεύθυνση. Εναλλακτικά, θα μπορούσε να

συμπεριλαμβάνει ένα πραγματικό όνομα, IRC παρώνυμα, URL φωτογραφιών που απεικονίζουν το συγκεκριμένο πρόσωπο, κ.λ.π.

Η παρακάτω λίστα κώδικα δείχνει ένα πολύ μικρό FOAF γράφο, που εκφράζεται σε σύνταξη Turtle. Περιέχει περιγραφές τεσσάρων μυθοπλαστικών προσώπων, αλλά οι περιγραφές έχουν η κάθε μία διαφορετικά είδη ιδιοτήτων.

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

_:a foaf:name         "Jon Foobar" ;
    foaf:mbox         <mailto:jon@foobar.xx> ;
    foaf:depiction    <http://foobar.xx/2005/04/jon.jpg> .

_:b foaf:name         "A. N. O'Ther" ;
    foaf:mbox         <mailto:a.n.other@example.net> ;
    foaf:depiction    <http://example.net/photos/an-2005.jpg> .

_:c foaf:name         "Liz Somebody" ;
    foaf:mbox_sha1sum "3f01fa9929df769aff173f57dec2fe0c2290aeea"

_:d foaf:name         "M Benn" ;
    foaf:depiction    <http://mbe.nn/pics/me.jpeg> .
```

Ένας μικρός FOAF γράφος που περιγράφει 4 ανθρώπους

Προαιρετικά ταιριάσματα

Ας υποθέσουμε ότι θέλουμε να γράψουμε μια ερώτηση που επιστρέφει το όνομα κάθε προσώπου που περιγράφεται στη προηγούμενη λίστα κώδικα, μαζί με ένα σύνδεσμο σε μια φωτογραφία για τον καθένα, αν υπάρχει διαθέσιμη. Μία SELECT ερώτηση που το σχήμα γράφου περιλαμβάνει μια ιδιότητα foaf:depiction θα έβρισκε 3 μόνο απαντήσεις. Η «Liz Somebody» δεν θα σχημάτιζε μια λύση, γιατί έχει μια ιδιότητα foaf:name αλλά καμία ιδιότητα foaf:depiction, και χρειαζόμαστε και τις δυο ιδιότητες για να μας φέρει αποτέλεσμα η ερώτηση.

Σε αυτή την περίπτωση χρησιμοποιούμε την λέξη κλειδί της SPARQL, την OPTIONAL. Αυτά τα μπλοκ προσδιορίζουν επιπρόσθετα σχήματα γράφων που δεν προκαλούν λύσεις που

απορρίπτονται αν δεν ταιριάξουν μεταξύ τους, αλλά προσδένονται στο γράφο όταν ταιριάξουν. Η επόμενη λίστα κώδικα αναπαριστά μια ερώτηση που βρίσκει το foaf:name κάθε προσώπου στα FOAF δεδομένα, και έπειτα προαιρετικά βρίσκει την ιδιότητα foaf:depiction.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?depiction
WHERE {
        ?person foaf:name ?name .
        OPTIONAL {
            ?person foaf:depiction ?depiction .
        } .
      }
```

Querying FOAF data with an optional block

Ο παρακάτω κώδικας, δείχνει τα αποτελέσματα της ερώτησης στα ανωτέρω. Με αυτή την άποψη, η ερώτηση είναι παρόμοια με ένα left outer join στην SQL γλώσσα βάσεων δεδομένων.

```
-----------------------------------------------------------
| name           | depiction                              |
===========================================================
| "A. N. O'Ther" | <http://example.net/photos/an-2005.jpg> |
| "Jon Foobar"   | <http://foobar.xx/2005/04/jon.jpg>      |
| "Liz Somebody" |                                         |
| "M Benn"       | <http://mbe.nn/pics/me.jpeg>            |
-----------------------------------------------------------
```

Αποτελέσματα της ερώτησης

Εναλλακτικά ταιριάσματα

Τα εναλλακτικά ταιριάσματα προσδιορίζονται δηλώνοντας πολλαπλά σχήματα γράφων, με τη λέξη-κλειδί UNION ανάμεσά τους. Η ερώτηση που φαίνεται στη παρακάτω λίστα κώδικα βρίσκει το όνομα κάθε προσώπου στον FOAF γράφο, μαζί με τις ιδιότητές τους *foaf:mbox* ή *foaf:mbox_sha1sum*. Ο «M Benn» δεν είναι απάντηση της ερώτησης, γιατί δεν έχει ούτε την ιδιότητα *foaf:mbox* ή την ιδιότητα *foaf:mbox_sha1sum*. Σε αντίθεση με τα OPTIONAL

σχήματα γράφων, κάθε μια από τις εναλλακτικές ιδιότητες πρέπει να ταιριάζει σε κάθε αποτέλεσμα της ερώτησης.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name ?mbox
WHERE {
  ?person foaf:name ?name .
  {
      { ?person foaf:mbox ?mbox } UNION { ?person foaf:mbox_sha1sum ?mbox }
  }
}

-----------------------------------------------------------------------
| name                | mbox                                          |
=======================================================================
| "Jon Foobar"        | <mailto:jon@foobar.xx>                        |
| "A. N. O'Ther"      | <mailto:a.n.other@example.net>                |
| "Liz Somebody"      | "3f01fa9929df769aff173f57dec2fe0c2290aeea" |
-----------------------------------------------------------------------
```

Μια ερώτηση με εναλλακτικά ταιριάσματα και τα αποτελέσματά της

Περιορισμοί τιμών

Η λέξη-κλειδί FILTER στην SPARQL περιορίζει τα αποτελέσματα μιας ερώτησης θέτοντας περιορισμούς στις τιμές των μεταβλητών. Αυτοί οι περιορισμοί στις τιμές των μεταβλητών είναι λογικές εκφράσεις που αποτιμώνται σε boolean τιμές, και μπορούν να συνδυαστούν με λογικούς τελεστές (&& και ||). Για παράδειγμα, μια ερώτηση που επιστρέφει μια λίστα ονομάτων θα μπορούσε να αλλάξει με ένα φίλτρο για να επιστρέφει μόνο ονόματα που ταιριάζουν με μία δεδομένη συνηθισμένη έκφραση. Ή όπως φαίνεται στη παρακάτω λίστα κώδικα, μπορούν να βρεθούν αντικείμενα σε ένα RSS feed που δημοσιεύτηκαν ανάμεσα σε δύο ημερομηνίες με ένα φίλτρο που ορίζει όρια στις ημερομηνίες που δημοσιεύτηκαν τα αντικείμενα.

```
PREFIX rss:   <http://purl.org/rss/1.0/>
PREFIX xsd:   <http://www.w3.org/2001/XMLSchema#>
PREFIX  dc:   <http://purl.org/dc/elements/1.1/>
SELECT ?item_title ?pub_date
WHERE {
      ?item rss:title ?item_title .
      ?item dc:date ?pub_date .
      FILTER xsd:dateTime(?pub_date) >= "2005-04-01T00:00:00Z"^^xsd:dateTime &&
             xsd:dateTime(?pub_date) < "2005-05-01T00:00:00Z"^^xsd:dateTime
      }
```

Χρησιμοποιώντας ένα φίλτρο για να πάρουμε RSS feed αντικείμενα που δημοσιεύτηκαν τον Απρίλιο του 2005

Δουλεύοντας με πολλαπλούς γράφους

Έως τώρα όλες οι ερωτήσεις που έχουν παρουσιαστεί περιείχαν δεδομένα που αποτελούνται μόνο από ένα απλό RDF γράφο. Στην SPARQL ορολογία, αυτές οι ερωτήσεις έχουν γίνει στον «background» γράφο. Ο «background» γράφος είναι ό,τι προσδιορίζεται από την πρόταση του FROM της ερώτησης.

Η SPARQL μπορεί να κάνει αναζήτηση σε οποιοδήποτε αριθμό γράφων. Αυτοί οι πρόσθετοι γράφοι προσδιορίζονται από τα URI τους, και είναι ο καθένας ξεχωριστός μέσα σε μια ερώτηση (query). Όπως με τον background γράφο, οι πολλαπλοί γράφοι μπορούν να προσδιοριστούν μέσα στην ίδια την ερώτηση, χρησιμοποιώντας το «FROM NAMED <URI>», όπου το URI προσδιορίζει το γράφο.

Οι πολλαπλοί γράφοι χρησιμοποιούνται μέσα σε μια SPARQL ερώτηση με τη λέξη-κλειδί GRAPH, σε συνδυασμό με ένα URI γράφου ή ένα όνομα μεταβλητής. Αυτή η λέξη-κλειδί ακολουθείται από ένα σχήμα γράφου για να ταιριάζει με το γράφο στην ερώτηση.

Βρίσκοντας απαντήσεις μέσα σε ένα συγκεκριμένο γράφο

Όταν η λέξη-κλειδί GRAPH χρησιμοποιείται με το URI ενός γράφου, το σχήμα του γράφου εφαρμόζεται σε όποιο γράφο προσδιορίζεται με αυτό το URI. Αν βρεθούν ταιριάσματα στο συγκεκριμένο γράφο, αποτελούν μέρος μιας απάντησης της ερώτησης αναζήτησης. Στη παρακάτω λίστα κώδικα, δυο συγκεκριμένοι FOAF γράφοι περνάνε μέσα στην ερώτηση αναζήτησης. Οι απαντήσεις είναι τα ονόματα εκείνων των ανθρώπων που βρίσκονται και

στους δύο γράφους. Να σημειωθεί ότι οι κόμβοι που αναπαριστούν ανθρώπους σε καθένα από τους 2 FOAF γράφους είναι κενοί κόμβοι, και έχουν εμβέλεια μόνο στο γράφο που τους περιέχει. Αυτό σημαίνει ότι ο ίδιος κόμβος ενός προσώπου δεν μπορεί να υπάρχει και στους δύο συγκεκριμένους γράφους στην ερώτηση, και έτσι πρέπει να χρησιμοποιηθούν διαφορετικές μεταβλητές (x και y) για να τους αναπαραστήσουν

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?name
FROM NAMED <jon-foaf.rdf>
FROM NAMED <liz-foaf.rdf>
WHERE {
  GRAPH <jon-foaf.rdf> {
    ?x rdf:type foaf:Person .
    ?x foaf:name ?name .
  } .
  GRAPH <liz-foaf.rdf> {
    ?y rdf:type foaf:Person .
    ?y foaf:name ?name .
  } .
}
```

Ερώτηση για να βρούμε ανθρώπους που περιγράφονται σε 2 συγκεκριμένους FOAF γράφους

Συνδυάζοντας background δεδομένα και καθορισμένους γράφους

Οι ερωτήσεις αναζήτησης μπορούν επίσης να χρησιμοποιήσουν background δεδομένα σε συνδυασμό με καθορισμένους γράφους. Η παρακάτω λίστα κώδικα, χρησιμοποιεί το live RSS feed από την PlanetRDF.com ως background δεδομένα, και κάνει ερώτηση αναζήτησης σε συνδυασμό με ένα συγκεκριμένο γράφο που περιέχει ένα καθορισμένο FOAF προφίλ. Η ιδέα είναι να δημιουργηθεί ένα προσωποποιημένο RSS feed βρίσκοντας τα 10 τελευταία posts από μπλόγκερς που ξέρει κάποιο συγκεκριμένο πρόσωπο. Το πρώτο μέρος της ερώτησης αναζήτησης βρίσκει τον κόμβο που αναπαριστά το συγκεκριμένο στο αρχείο FOAF, και έπειτα βρίσκει τα ονόματα των ανθρώπων που ο γράφος δείχνει ότι ξέρει το συγκεκριμένο πρόσωπο. Το δεύτερο μέρος ψάχνει για αντικείμενα στο RSS feed που δημιουργούνται από αυτούς τους ανθρώπους. Τελικά, το σύνολο του αποτελέσματος

ταξινομείται με την ημερομηνία δημιουργίας του αντικειμένου, και περιορίζεται στο να επιστρέψει μόνο 10 αποτελέσματα.

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rss:  <http://purl.org/rss/1.0/>
PREFIX dc:   <http://purl.org/dc/elements/1.1/>

SELECT ?title ?known_name ?link

FROM <http://planetrdf.com/index.rdf>
FROM NAMED <phil-foaf.rdf>

WHERE {
        GRAPH <phil-foaf.rdf> {
          ?me foaf:name "Phil McCarthy" .
          ?me foaf:knows ?known_person .
          ?known_person foaf:name ?known_name .
        } .

        ?item dc:creator ?known_name .
        ?item rss:title ?title .
        ?item rss:link ?link .
        ?item dc:date ?date.
      }

ORDER BY DESC[?date] LIMIT 10
```

Ανακτώντας ένα προσωποποιημένο live PlanetRDF feed

Συμπέρασμα

Τα παραδείγματα σε αυτή την ενότητα βοηθούν στην κατανόηση των βασικών χαρακτηριστικών και της σύνταξης της SPARQL, μαζί με τα οφέλη που φέρνει αυτή η γλώσσα αναζήτησης στις RDF εφαρμογές.

1.3.4.2. JENA

Ο Σημασιολογικός Ιστός είναι ουσιαστικά ένα σύστημα διαχείρισης μεταδεδομένων. Μπορούμε να χρησιμοποιήσουμε μεταδεδομένα για να περιγράψουμε οντότητες που έχουν σύνολα ιδιοτήτων και σχέσεων σε πολλές άλλες κλάσεις. Γλώσσες όπως η RDF, η RDFS και η DAML παρέχουν τα markup tags που μπορούν να χρησιμοποιηθούν για να εκφράσουν (ή να βεβαιώσουν) αυτές τις σχέσεις. Σε άλλα σημεία της παρούσης διατριβής, έχουμε τα βασικά στοιχεία αυτών των γλωσσών και έχουμε κατασκευάσει μικρές οντολογίες για παραδείγματα.

Η δημιουργία και η διαχείριση μεγάλης κλίμακας οντολογιών είναι μια διαδικασία που είναι καλύτερο και ασφαλέστερο να την υλοποιούμε με διάφορα εργαλεία οργάνωσης γνώσης και αυτό το λόγο είναι φυσικά απαραίτητο η ύπαρξη πρότυπων γλωσσών μεταδεδομένων και περιγραφικών μεθοδολογιών. Μπορούμε να ταξινομήσουμε αυτά τα εργαλεία σε δύο σύνολα: ετοιμοπαράδοτα προϊόντα και υλοποιήσεις τύπου βιβλιοθήκης / API. Οι ετοιμοπαράδοτες λύσεις συμπεριλαμβάνουν προϊόντα όπως το OntoMat και το Protégé, τα οποία μπορούν να χρησιμοποιηθούν άμεσα για να γραφτούν οπτικά τα μεταδεδομένα, με εύχρηστο τρόπο.

Πολλές όμως φορές, είναι συχνά επιθυμητό να χτίζει κανείς εξολοκλήρου νέες εφαρμογές για την διαχείριση μεταδεδομένων του Σημασιολογικού Ιστού. Αντί για ετοιμοπαράδοτα προϊόντα, σε αυτή την περίπτωση, επιθυμούμε γενικού σκοπού API και βιβλιοθήκες που μπορούν να συμπεριλάβουν τα προγράμματά μας τα οποία θα χρησιμοποιήσουμε για να κατασκευάσουμε τις δικές μας εφαρμογές. Η Jena αποτελεί ένα τέτοιο API για την Java, μέσα από εφαρμογές της οποίας ολοκληρώνεται και χρησιμοποιείται ενοποιημένα σαν μία ολοκληρωμένη εφαρμογή.

Δημιουργία Περιγραφικών Αρχείων RDF

Το RDF (Resource Description Framework) είναι ένα πρότυπο όπως αναφέρθηκε, (τεχνικά μια σύσταση του W3C) για την περιγραφή πόρων, βάσει των οποίων μπορούμε να σκεφτούμε οτιδήποτε, έτσι ώστε να προσδιορίσουμε κάποια συγκεκριμένη ή αφηρημένη έννοια, σε επίπεδο μεταπληροφορίας.

Σε αυτά τα διαγράμματα ο πόρος φαίνεται ως έλλειψη και προσδιορίζεται από ένα URI (Uniform Resource Identifier). Οι πόροι έχουν ιδιότητες. Μια ιδιότητα αναπαρίσταται από ένα τόξο, που έχει ως ετικέτα το όνομα μιας ιδιότητας. Το όνομα της ιδιότητας είναι επίσης ένα URI, αλλά καθώς το URI είναι σχετικά μεγάλο σε μήκος και βαρύ, το διάγραμμα το δείχνει σε μια μορφή XML qname. Το κομμάτι πριν το «:» ονομάζεται πρόθεμα του namespace. Το κομμάτι μετά το «:» ονομάζεται τοπικό όνομα (local name) και αναπαριστά ένα όνομα σε αυτό το namespace. Οι ιδιότητες αναπαριστούνται συνήθως σε αυτή την qname μορφή όταν γράφονται σαν RDF XML και είναι βολική σύντομη γραφή για την αναπαράστασή τους σε διαγράμματα και κείμενο. Αυστηρά, όμως, οι ιδιότητες προσδιορίζονται από ένα URI. Η μορφή nsprefix:localname είναι σύντομη γραφή για το URI του namespace συνδεόμενη με το localname. Η κάθε ιδιότητα έχει μια τιμή. Εδώ, η τιμή είναι ένα literal, που προς το παρόν μπορούμε να τη θεωρήσουμε ως string χαρακτήρων. Τα literals φαίνονται ως τετράγωνα.

Η JENA είναι ένα Java API που μπορεί να χρησιμοποιηθεί για να δημιουργεί και να διαχειρίζεται RDF γράφους. Η JENA έχει κλάσεις αντικειμένων για να αναπαριστά πόρους, ιδιότητες και literal και ονομάζονται αντίστοιχα Resource, Property, Literal. Στην JENA, ένας γράφος ονομάζεται μοντέλο και αναπαρίσταται από την διεπαφή Model.

Στο σημείο αυτό μας ενδιαφέρει η διαλειτουργικότητα της Jena και της γλώσσας RDF, λόγω της εκτεταμένης χρήσης στην εφαρμογή Concept Net, όπου η Jena χρησιμοποιείται για την ανάγνωση αρχείων RDF και κατόπιν για την δημιουργία σε πραγματικό χρόνο (instantiation) του μοντέλου της οντολογίας. Κατόπιν χρησιμοποιείται στην εξαγωγή χρήσιμων δομικών χαρακτηριστικών της οντολογίας μέσω Jena queries.

Παρακάτω αναφερόμαστε σε παράδειγμα που αφορά οργάνωσης μεταδεδομένων (metadata) σε μορφή RDF, των VCARDS, όπως αυτές ορίζονται στον W3C(1). Το RDF πρότυπο αποτελεί σήμερα την καλύτερη επιλογή για την απεικόνιση διαγραμμάτων κόμβων και τόξων που συνδέουν έννοιες μεταξύ τους.

Ο κώδικας για τη δημιουργία ενός μοντέλου έχει ως ακολούθως:

```
// some definitions
   static String personURI    = "http://foo.com/staff/Michael";
   static String fullName     = "Michael Fox";
```

```
// create an empty graph
 Model model = new ModelMem();

// create the resource
   Resource michaelFox= model.createResource(personURI);

 // add the property
  michaelFox.addProperty(VCARD.FN, fullName);
```

Το αποτέλεσμα σε μορφή RDF είναι της ακόλουθης μορφής:

```
<rdf:RDF
 xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
 xmlns:vcard='http://www.w3.org/2001/vcard-rdf/3.0#'
>
 <rdf:Description rdf:about='http://foo.com/staff/Michael'>
   <vcard:FN>Michael Fox</vcard:FN>
 </rdf:Description>
</rdf:RDF>
```

Αρχίζει με κάποιους σταθερούς ορισμούς και έπειτα δημιουργεί ένα άδειο μοντέλο. Η ModemMem είναι μια κλάση που υλοποιεί την διεπαφή Model και κρατά όλα τα δεδομένα στην κύρια μνήμη.

Ο πόρος Michael Fox δημιουργείται και μια ιδιότητα προστίθεται σε αυτόν. Η ιδιότητα παρέχεται από μια σταθερή κλάση VCARD, που κρατά αντικείμενα που αναπαριστούν όλους τους ορισμούς στο σχήμα VCARD. Η JENA παρέχει σταθερές κλάσεις για άλλα γνωστά σχήματα, όπως τα RDF, Dublin Core, DAML. Ακολούθως το μοντέλο εμπλουτίζεται με περισσότερους και λεπτομερέστερους πόρους γνώσεις., χρησιμοποιώντας εξειδικευμένα χαρακτηριστικά RDF/Jena.

```
    String personURI    = "http://foo.com/staff/Michael";
    String givenName    = "Michael";
    String familyName   = "Fox";
    String fullName     = givenName + " " + familyName;
    String other ="Mike";
    String prefix ="Dr";
    // create an empty graph
    Model model = new ModelMem();
```

```
    // create the resource
    //   and add the properties cascading style
    Resource michaelFox = model.createResource(personURI)
              .addProperty(VCARD.FN, fullName)
              .addProperty(VCARD.N,
                        model.createResource()
                             .addProperty(VCARD.Given, givenName)
                             .addProperty(VCARD.Family, familyName)
                             .addProperty(VCARD.Other, other)
                             .addProperty(VCARD.Prefix, prefix));
```

Το αποτέλεσμα σε μορφή RDF προκύπτει ως ακολούθως:

```
<rdf:RDF
 xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
 xmlns:vcard='http://www.w3.org/2001/vcard-rdf/3.0#'
>
 <rdf:Description rdf:nodeID='A0'>
   <vcard:Given>Michael</vcard:Given>
   <vcard:Family>Fox</vcard:Family>
   <vcard:Other>Mike</vcard:Other>
   <vcard:Prefix>Dr</vcard:Prefix>
 </rdf:Description>
 <rdf:Description rdf:about='http://foo.com/staff/Michael'>
   <vcard:FN>Michael Fox</vcard:FN>
   <vcard:N rdf:nodeID='A0'/>
 </rdf:Description>
</rdf:RDF>
```

Εδώ έχουμε προσθέσει μια νέα ιδιότητα, vcard:N για να αναπαραστήσουμε τη δομή του ονόματος του Michael Fox. Ας σημειωθεί ότι η ιδιότητα vcard:N παίρνει ως τιμή έναν πόρο. Εδώ, είναι ένας κενός κόμβος.

Jena - Δηλώσεις (statements)

Μια δήλωση ονομάζεται μερικές φορές τριπλή (triple), εξαιτίας των τριών μερών που συμμετέχουν σε αυτή. Ένας RDF γράφος αναπαρίσταται ως ένα σύνολο δηλώσεων. Κάθε κλήση της addProperty πρόσθετε μια άλλη δήλωση. Η διεπαφή του μοντέλου της JENA ορίζει μια μέθοδο listStatement(), η οποία επιστρέφει έναν επαναλαμβανόμενο στοιχείο iterator. Κάθε φορά που καλείται η επόμενη μέθοδος του iterator, επιστρέφει ένα

αντικείμενο JENA τύπου Statement. Η διεπαφή Statement παρέχει συμπληρωματικές μεθόδους για το αντικείμενο, για το κατηγορούμενο, και τα αντικείμενα μιας δήλωσης, έτσι ώστε αυτή να γίνεται κατανοητή.

```
// list the statements in the graph
StmtIterator iter = model.listStatements();
// print out the predicate, subject and object of each statement
while (iter.hasNext()) {
   Statement stmt      = iter.next();        // get next statement
   Resource  subject   = stmt.getSubject();  // get the subject
   Property  predicate = stmt.getPredicate(); // get the predicate
   RDFNode   object    = stmt.getObject();   // get the object
   System.out.print(subject.toString());
   System.out.print(" " + predicate.toString() + " ");
   if (object instanceof Resource) {
      System.out.print(object.toString());
   } else {
       // object is a literal
       System.out.print(" \"" + object.toString() + "\");
   }
   System.out.println(" .");
}
```

Αφού το αντικείμενο μιας δήλωσης μπορεί να είναι είτε ένας πόρος είτε ένα literal, η μέθοδος getObject() επιστρέφει ένα αντικείμενο, το RDFNode, που είναι μια συνηθισμένη υπερκλάση και του Resource και του Literal. Το βασικό αντικείμενο είναι του κατάλληλου τύπου, έτσι ο κώδικας χρησιμοποιεί το **instanceof** για να το προσδιορίσει και να το επεξεργαστεί κατάλληλα. Όταν εκτελέσουμε το πρόγραμμα, η έξοδος θα είναι η ακόλουθη.

```
 anon:f62373:f6436ca2a6:-7fff http://www.w3.org/2001/vcard-rdf/3.0#Given  "Michael" .
http://foo.com/staff/Michael http://www.w3.org/2001/vcard-rdf/3.0#N anon:f62373:f6436ca2a6:-7fff .
anon:f62373:f6436ca2a6:-7fff http://www.w3.org/2001/vcard-rdf/3.0#Family  "Fox" .
anon:f62373:f6436ca2a6:-7fff http://www.w3.org/2001/vcard-rdf/3.0#Prefix  "Dr" .
anon:f62373:f6436ca2a6:-7fff http://www.w3.org/2001/vcard-rdf/3.0#Other  "Mike" .
http://foo.com/staff/Michael http://www.w3.org/2001/vcard-rdf/3.0#FN  "Michael Fox" .
```

Αναλύοντας τα ανωτέρω, θα παρατηρήσουμε ότι κάθε γραμμή αποτελείται από 3 πεδία που αναπαριστούν το υποκείμενο, το κατηγορούμενο και το αντικείμενο κάθε δήλωσης. Υπάρχουν 4 δηλώσεις. Η «anon:f62373:f6436ca2a6:-7fff» είναι ένας εσωτερικός identifier

που παράγεται από την JENA. Δεν είναι ένα URI και δεν θα έπρεπε να συγχέεται με αυτό. Είναι απλά μια εσωτερική ετικέτα που χρησιμοποιείται από την υλοποίηση της JENA.

Jena - Γράφοντας Δομές RDF

Η JENA έχει μεθόδους για την ανάγνωση και εγγραφή RDF ως XML. Αυτές μπορούν να χρησιμοποιηθούν για να αποθηκεύσουν ένα μοντέλο RDF σε αρχείο, και αργότερα να το ξαναδιαβάσει.

```
// now write the model in XML form to a file
model.write(new PrintWriter(System.out));
```

Οι προδιαγραφές RDF περιγράφουν πώς να αναπαραστήσουμε το RDF σε XML. Η σύνταξη RDF XML είναι αρκετά πολύπλοκη. Η RDF συνήθως εμπεριέχεται σε ένα <rdf: RDF> στοιχείο. Το στοιχείο είναι προαιρετικό αν υπάρχουν άλλοι τρόποι να γνωρίζουμε ότι κάποιο XML είναι RDF, αλλά συνήθως είναι παρόν. Το RDF στοιχείο προσδιορίζει τα δύο namespaces που χρησιμοποιούνται στο κείμενο. Υπάρχει ένα στοιχείο <rdf:Description> που περιγράφει τον πόρο του οποίου το URI είναι «http://foo.com/staff/Michael». Αν έλειπε το rdf:about χαρακτηριστικό, αυτό το στοιχείο θα αναπαριστούσε ένα κενό κόμβο.

Το στοιχείο <vcard:FN> περιγράφει μια ιδιότητα του πόρου. Το όνομα της ιδιότητας είναι «FN» στο vcard namespace. Το RDF το μετατρέπει σε μια URI αναφορά ενώνοντας την URI αναφορά για το πρόθεμα του namespace και «FN», το τοπικό μέρος του ονόματος. Αυτό δίνει μια URI αναφορά «http://www.w3.org/2001/vcard-rdf/3.0#FN». Η τιμή της ιδιότητας είναι το literal «Michael Fox».

Το στοιχείο <vcard:N> είναι ένας πόρος. Σε αυτή την περίπτωση αναπαρίσταται με μία σχετική URI αναφορά. Το RDF το μετατρέπει σε μια απόλυτη URI αναφορά ενώνοντας την με το βασικό URI του τρέχοντος αρχείου.

Υπάρχει όμως μία ατέλεια σε αυτή την RDF XML μορφή. Η σύνταξη RDF XML δεν μπορεί να αναπαριστά όλους τους RDF γράφους. Για παράδειγμα, δεν μπορεί να αναπαραστήσει ένα κενό κόμβο, που είναι το αντικείμενο δύο δηλώσεων. Ο κειμενογράφος που χρησιμοποιήσαμε (dumb writer) για να γράψουμε αυτό το RDF XML δεν προσπαθεί να γράψει σωστά το υποσύνολο των γράφων που μπορεί να γραφεί σωστά. Δίνει ένα URI σε κάθε κενό κόμβο, και έτσι δεν είναι άλλο πια κενός.

Η JENA έχει μια επεκτάσιμη διεπαφή (extensible interface) που επιτρέπει σε νέους κειμενογράφους για διαφορετικές γλώσσες για το RDF, να ολοκληρώνονται εύκολα. Η JENA επίσης συμπεριλαμβάνει ένα πιο εξεζητημένο κειμενογράφο, με περισσότερες δυνατότητες, RDF XML που μπορούμε να τον καλέσουμε ορίζοντας ένα άλλο πεδίο (argument)στην κλήση της μεθόδου write().

```
// now write the model in XML form to a file
model.write(new PrintWriter(System.out), "RDF/XML-ABBREV");

<rdf:RDF
   xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
   xmlns:RDFNsId1='http://www.w3.org/2001/vcard-rdf/3.0#'>
   <rdf:Description rdf:about='http://foo.com/staff/Michael'>
      <RDFNsId1:FN>Michael Fox</RDFNsId1:FN>
      <RDFNsId1:N
         RDFNsId1:Given='Michael'
         RDFNsId1:Family='Fox'
         RDFNsId1:Other='Mike'
         RDFNsId1:Prefix='Dr'/>
   </rdf:Description>
</rdf:RDF>
```

Αυτός ο κειμενογράφος, που ονομάζεται PrettyWriter, εκμεταλλεύεται τα χαρακτηριστικά της σύντομης RDF XML σύνταξης για να γράφει ένα γράφο πιο συμπαγή. Μπορεί επίσης να διατηρεί τους κενούς κόμβους όπου είναι δυνατόν. Όμως δεν είναι κατάλληλος για πολύ μεγάλους γράφους, καθώς η απόδοσή του δεν είναι αποδεκτή. Για να γράφει κάποιος μεγάλους γράφους και να διατηρεί τους κενούς κόμβους, γράφει σε φορμά N-triple

```
// now write the model in XML form to a file
model.write(new PrintWriter(System.out), "N-TRIPLE");


_:Af62373X3aXf64377bd20X3aXX2dX7fff <http://www.w3.org/2001/vcard-rdf/3.0#Family> "Fox" .
_:Af62373X3aXf64377bd20X3aXX2dX7fff <http://www.w3.org/2001/vcard-rdf/3.0#Given> "Michael" .
<http://foo.com/staff/Michael> <http://www.w3.org/2001/vcard-rdf/3.0#N>
_:Af62373X3aXf64377bd20X3aXX2dX7fff .
_:Af62373X3aXf64377bd20X3aXX2dX7fff <http://www.w3.org/2001/vcard-rdf/3.0#Prefix> "Dr" .
_:Af62373X3aXf64377bd20X3aXX2dX7fff <http://www.w3.org/2001/vcard-rdf/3.0#Other> "Mike" .
<http://foo.com/staff/Michael> <http://www.w3.org/2001/vcard-rdf/3.0#FN> "Michael Fox" .
```

Το ακόλουθο απόσπασμα κώδικα θα διαβάσει και θα γράψει τον γράφο:

```
Model model = new ModelMem();

    // Read from an input stream
    InputStream is = new BufferedInputStream(
       new FileInputStream(
       "/home/byildiz/Research/jbproject2/jena/computeResource.rdf"));

    // read the RDF/XML file
    model.read(new InputStreamReader(is), "");

    Resource NS1 = model.getResource("http://protege.stanford.edu/computeResource");
    System.out.println("NS1 " + NS1);
       /*  // retrieve the value of the N property
         Resource name = (Resource) NS1.getProperty()
                                   .getObject();
         // retrieve the given name property
         String fullName = NS1.getProperty(contact)
                           .getString();
    // write it to standard out
    model.write(new PrintWriter(System.out));
```

Το δεύτερο όρισμα στην κλήση της μεθόδου read() είναι το URI που θα χρησιμοποιηθεί για την ανάλυση σχετικών URI. Καθώς δεν υπάρχουν σχετικές URI αναφορές στο τεστ αρχείο, μπορεί να είναι κενό:

Το input αρχείο, που δημιουργήθηκε από το Protégé έχει ως ακολούθως:

```
<?xml version='1.0' encoding='ISO-8859-1'?>
<!DOCTYPE rdf:RDF [
          <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
          <!ENTITY a 'http://protege.stanford.edu/system#'>
          <!ENTITY computeResource 'http://protege.stanford.edu/computeResource#'>
          <!ENTITY rdfs 'http://www.w3.org/TR/1999/PR-rdf-schema-19990303#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
          xmlns:computeResource="&computeResource;"
          xmlns:a="&a;"
```

```
        xmlns:rdfs="&rdfs;">
<a:_pal_constraint rdf:about="&computeResource;KB_9895_00019"
        rdfs:label="KB_9895_00019"/>
<a:_pal_constraint rdf:about="&computeResource;computeResource_00020"
        a:_pal_name="compres1"
        rdfs:label="compres1"/>
<a:_pal_constraint rdf:about="&computeResource;computeResource_00021"
        a:_pal_name="compres2"
        rdfs:label="compres2"/>
<computeResource:computeResourceStatic rdf:about="&computeResource;computeResource_00022"
        computeResource:contact="byildiz"
        computeResource:cpuSpeedGHz="1.7"
        computeResource:cpusPerNode="1"
        computeResource:description="linux machine"
        computeResource:hostName="machine1.ucs.indiana.edu"
        computeResource:infoURL="http://machine1.ucs.indiana.edu"
        computeResource:ipAddress="156.56.*.*"
        computeResource:machineKey="machine1"
        computeResource:machineName="machine1"
        computeResource:manufacturer="dell"
        computeResource:memoryGB="1"
        computeResource:model="8100"
        computeResource:numNodes="1"
        computeResource:organization="ptl"
        computeResource:os="linux"
        computeResource:peakPerformanceGFLOPS="1000"
        computeResource:scratchDiskGB="100"
        computeResource:site="machine1.ucs.indiana.edu"
        rdfs:label="1.7"/>
<computeResource:computeResourceStatic rdf:about="&computeResource;computeResource_00023"
        computeResource:contact="byildiz"
        computeResource:cpuSpeedGHz="2"
        computeResource:cpusPerNode="2"
        computeResource:description="server"
        computeResource:hostName="machine2.ucs.indiana.edu"
        computeResource:infoURL="machine2.ucs.indiana.edu"
        computeResource:ipAddress="156.56.*.*"
        computeResource:machineKey="darya"
        computeResource:machineName="machine2"
        computeResource:manufacturer="dell"
        computeResource:memoryGB="1"
```

```
        computeResource:model="1000"
        computeResource:organization="ptl"
        computeResource:os="sun sparc"
        computeResource:peakPerformanceGFLOPS="1500"
        computeResource:scratchDiskGB="200"
        computeResource:site="machine1.ucs.indiana.edu"
        rdfs:label="2"/>
</rdf:RDF>
```

Η έξοδος του αρχείου RDF, που διαβάστηκε και τυπώθηκε στην οθόνη:

```
NS1 http://protege.stanford.edu/computeResource
<rdf:RDF
 xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
 xmlns:NS0='http://www.w3.org/TR/1999/PR-rdf-schema-19990303#'
 xmlns:NS1='http://protege.stanford.edu/computeResource#'
 xmlns:NS2='http://protege.stanford.edu/system#'
>
 <rdf:Description rdf:about='http://protege.stanford.edu/computeResource#computeResource_00023'>
  <rdf:type rdf:resource='http://protege.stanford.edu/computeResource#computeResourceStatic'/>
  <NS1:contact>byildiz</NS1:contact>
  <NS1:cpuSpeedGHz>2</NS1:cpuSpeedGHz>
  <NS1:cpusPerNode>2</NS1:cpusPerNode>
  <NS1:description>server</NS1:description>
  <NS1:hostName>machine2.ucs.indiana.edu</NS1:hostName>
  <NS1:infoURL>machine2.ucs.indiana.edu</NS1:infoURL>
  <NS1:ipAddress>156.56.*.*</NS1:ipAddress>
  <NS1:machineKey>darya</NS1:machineKey>
  <NS1:machineName>machine2</NS1:machineName>
  <NS1:manufacturer>dell</NS1:manufacturer>
  <NS1:memoryGB>1</NS1:memoryGB>
  <NS1:model>1000</NS1:model>
  <NS1:organization>ptl</NS1:organization>
  <NS1:os>sun sparc</NS1:os>
  <NS1:peakPerformanceGFLOPS>1500</NS1:peakPerformanceGFLOPS>
  <NS1:scratchDiskGB>200</NS1:scratchDiskGB>
  <NS1:site>machine1.ucs.indiana.edu</NS1:site>
  <NS0:label>2</NS0:label>
 </rdf:Description>
 <rdf:Description rdf:about='http://protege.stanford.edu/computeResource#computeResource_00021'>
  <rdf:type rdf:resource='http://protege.stanford.edu/system#_pal_constraint'/>
  <NS2:_pal_name>compres2</NS2:_pal_name>
```

```
  <NS0:label>compres2</NS0:label>
</rdf:Description>
<rdf:Description rdf:about='http://protege.stanford.edu/computeResource#KB_9895_00019'>
  <rdf:type rdf:resource='http://protege.stanford.edu/system#_pal_constraint'/>
  <NS0:label>KB_9895_00019</NS0:label>
</rdf:Description>
<rdf:Description rdf:about='http://protege.stanford.edu/computeResource#computeResource_00020'>
  <rdf:type rdf:resource='http://protege.stanford.edu/system#_pal_constraint'/>
  <NS2:_pal_name>compres1</NS2:_pal_name>
  <NS0:label>compres1</NS0:label>
</rdf:Description>
<rdf:Description rdf:about='http://protege.stanford.edu/computeResource#computeResource_00022'>
  <rdf:type rdf:resource='http://protege.stanford.edu/computeResource#computeResourceStatic'/>
  <NS1:contact>byildiz</NS1:contact>
  <NS1:cpuSpeedGHz>1.7</NS1:cpuSpeedGHz>
  <NS1:cpusPerNode>1</NS1:cpusPerNode>
  <NS1:description>linux machine</NS1:description>
  <NS1:hostName>machine1.ucs.indiana.edu</NS1:hostName>
  <NS1:infoURL>http://machine1.ucs.indiana.edu</NS1:infoURL>
  <NS1:ipAddress>156.56.*.*</NS1:ipAddress>
  <NS1:machineKey>machine1</NS1:machineKey>
  <NS1:machineName>machine1</NS1:machineName>
  <NS1:manufacturer>dell</NS1:manufacturer>
```

Αναφορές: http://www.w3.org/TR/vcard-rdf , http://www.w3.org/TR/vcard-rdf

1.4. Συγκριση – Διαλειτουργικοτητα Οντολογιων και Β.Δ.

Η παρούσα Δ.Δ., αναφέρεται στην προσπάθεια αυτοματοποιημένης σύγκρισης οντολογιών, με σκοπό κατά την επίτευξή της να ανταλλαγούν μεταδεδομένα από την μια οντολογία στην άλλη. Η παρούσα Δ.Δ., ορίζει μεταξύ άλλων, στα πλαίσια της προτεινόμενης μεθοδολογίας της και μέτρα ενδιαφέροντος πιθανών παρομοίων κόμβων – εννοιών της, έτσι ώστε χρησιμοποιώντας αυτά, να βγαίνουν αυτοματοποιημένα συμπεράσματα για την ομοιότητά τους.

Για να λειτουργήσουν σωστά στο επίπεδο ανταλλαγής εννοιών θεωρούνται οι οντολογίες σαν δομές εννοιών, δηλαδή δομές οι οποίες θα πρέπει να υποστηρίζουν κριτήρια με βάση τα οποία θα μπορούν να αξιολογηθούν οι έννοιες τους από αυτοματοποιημένα υπολογιστικά συστήματα. Είναι πολύ σημαντικό σε αυτά τα πλαίσια να ορισθούν και να αναπτυχθούν μέτρα ομοιότητος (similarity measures), τα οποία θα είναι ικανά να συγκρίνουν τέτοιες δομές, για ίδια ή περίπου ίδια γνωστικά πεδία περιγεγραμμένα σε οντολογικές δομές. Ο σκοπός του παρόντος, είναι να παρουσιάσει μεθοδολογικά την τρέχουσα (state – of the art) κατάσταση στον τομέα αυτόν, στο πρόβλημα που ορίζεται σαν «πρόβλημα διαλειτουργικότητος σε σημασιολογικό επίπεδο» (semantic interoperability problem) τα υπέρ καθώς και τα χαρακτηριστικά των μεθόδων αποθήκευσης σημασιολογικού περιεχομένου σε οντολογίες και βάσεις δεδομένων και την επίπτωση τους σε προσπάθειες ενοποίησης και επαναχρησιμοποίησης των περιεχομένων τους με σκοπό την σημασιολογική επαναχρησιμοποίησή της (semantic reusability).

Στο παρελθόν έχουν αναπτυχθεί πολλές προσεγγίσεις σχετικά με την ολοκλήρωση των ετερογενών πηγών πληροφοριών. Τα τελευταία χρόνια η σημασιολογία (semantics), διαδραματίζει έναν σημαντικό ρόλο κατά τη διάρκεια του στόχου ολοκλήρωσης (integration) της πληροφορίας (H. Wache, T. V¨ogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hubner, 2001), εισέρχεται δε ολοένα και εστιάζει περισσότερο σε προσεγγίσεις, οι οποίες είναι βασισμένες οντολογικά κατά την ανωτέρω διαδικασία.

Τα βασικά θέματα που άπτονται στην επίλυση του προβλήματος της διαλειτουργικότητος είναι τα κατωτέρω:

Στο πλαίσιο της επίλυσης του προβλήματος της διαλειτουργικότητος των δεδομένων, ειδικότερα αυτών που είναι αποθηκευμένα σε συστήματα τα οποία είναι μέχρι σήμερα εν χρήσει σε κλασικές σχεσιακές βάσεις δεδομένων, το πρόβλημα κατά κύριο λόγο ανάγεται στην ομογενοποίηση και ενοποίηση των σχημάτων των βάσεων αυτών (database schema integration). Η διαδικασία αυτή, ενώ μπορεί να είναι επιτυχής για μικρές βάσεις δεδομένων και να δώσει αποτελέσματα, καταλήγει να είναι μια επίπονη διαδικασία, η οποία δεν μπορεί ούτε να αυτοματοποιηθεί αλλά και η οποία απαιτεί τη συνεχή ανθρώπινη εμπλοκή και παρέμβαση σε κάθε πιθανότητα επεκτάσεών της.

Πιο συγκεκριμένα, σύμφωνα με τους Sheth and Larson (A. Steth and S. Larson, 1990), υπάρχει στις βάσεις δεδομένων η ευθύνη και εναπόκειται στον χρήστη να κατανοήσει τη σημασιολογική έννοια των οντοτήτων των σχημάτων των βάσεων, οι οποίες καλούνται να διαλειτουργήσουν ως προς τα αποθηκευμένα στοιχεία τους (πίνακες, πεδία, τύποι αυτών).

Η εμπλοκή του χρήστη στο πρόβλημα της αντιστοίχισης σχημάτων (schema mapping) φτάνει σε τέτοιο επίπεδο, έτσι ώστε ανάλογα με την κατανόηση που έχει αυτός (υποκειμενικότητα) να προτείνει και το σχήμα της ενδιάμεσης διαδικασίας (federation schema) προκειμένου να υλοποιήσει τη διαλειτουργικότητα αυτή.

Στην καλύτερη περίπτωση η διαδικασία αυτή μπορεί να προχωρήσει γράφοντας μικρούς κανόνες που θα προσπαθούν να προτείνουν τμήματα του target schema από το source schema, πάλι όμως σε μια ημιαυτοποιημένη διαδικασία, με έντονη και υποκειμενική ανθρώπινη παρέμβαση.

Προκειμένου λοιπόν να προχωρήσει η ανάκτηση γνώσης μέσα από ένα πολύ μεγάλο πλήθος δεδομένων, τα οποία προέρχονται από ετερογενείς πηγές, χωρίς προσυμφωνημένο τρόπο περιγραφής, είναι υποχρεωτικό πρώτα να γίνει μια ευρέως αποδεκτή κατανόηση του προβλήματος αυτού, δηλαδή της περιγραφής με σημασιολογικά μεταδεδομένα (semantics) των προς διαλειτουργία δεδομένων και αντιστοίχων δομών τους.

Οι κλασικές βάσεις δεδομένων δεν διαθέτουν δομές οι οποίες να είναι σε θέση να υποδεχθούν σημασιολογικά μεταδεδομένα και την συσχέτισή τους, με έναν ομοιόμορφο τρόπο. Επίσης, οι κλασικοί μηχανισμοί DBMS δεν διαθέτουν τις υποδομές εκείνες προκειμένου να υποστηρίξουν τη λογική, η οποία θα χρησιμοποιηθεί σε περιπτώσεις αντικρουόμενων σημασιολογικών θεμάτων (conflict resolution logic). Επίσης, ακόμα και

στην περίπτωση που κάποιος κατάφερνε να απεικονίσει τα δεδομένα, να τα χαρακτηρίσει κατάλληλα, ομοιογενώς και ομοιόμορφα με σημασιολογικά μεταδεδομένα ακόμα δε και να περιγράψει μια λογική επίλυση σημασιολογικών προβλημάτων, απομένει το δυσεπίλυτο πρόβλημα της δυνατότητας εύκολου αποχωρισμού(decoupling) των δεδομένων, των μεταδεδομένων και της λογικής για λόγους συντήρησης του καθενός από τα παραπάνω, όπως για παράδειγμα προσθήκη νέων data, νέων μεταδεδομένων και διασπορά τους, προσθήκη / αφαίρεση λογικής κλπ.

Προκειμένου τα παραπάνω να επιλυθούν με αποτελεσματικότερο τρόπο, τα προβλήματα της ενοποίησης δεδομένων (data intergration) ανάκτησης και συσχέτισης αντιμετωπίζονται στα πλαίσια του σημασιολογικού (semantics) με βάση τον μηχανισμό υποστήριξης των οντολογιών.

Οι οντολογίες εδώ αντιμετωπίζονται σαν απεικόνιση ενός πεδίου γνώσεως, οι οποίες έχουν έννοιες, ιδιότητες εννοιών και σχέσεις. Στις περισσότερες των περιπτώσεων οι έννοιες οργανώνονται γύρω από μια μορφή-γράφου-δένδρου η οποία αποκαλείται ταξονομία (taxonomy), όπου η κάθε έννοια θεωρείται εξειδίκευση της προηγούμενης έννοιας (parent concept). Με αυτά τα δεδομένα η σημασιολογική συσχέτιση στα πλαίσια των οντολογιών ανάγεται στη σωστή αντιστοίχηση των ταξινομιών τους, μια διαδικασία η οποία είναι αντίστοιχη με τη διαδικασία αντιστοίχησης των σχημάτων των βάσεων δεδομένων. Σε αυτή τη διαδικασία, μεγάλη βαρύτητα έχει η ανάδειξη κατάλληλων μέτρων ομοιότητας (similarity measures) τα οποία θα μπορούν να εφαρμοστούν για την αυτοματοποίηση αυτής της διαδικασίας.

Τα μέτρα που θα πρέπει να υιοθετηθούν για τη διαδικασία αυτή θα πρέπει να ικανοποιούν τις παρακάτω συνθήκες:

1. Να είναι σωστά ορισμένα (well defined)
2. Να είναι ακριβή (exact measures)
3. Να επιτρέπουν ευελιξία στη συσχέτιση των δεδομένων δηλαδή πιθανές συσχετίσεις είτε με έννοιες ανωτέρου επιπέδου (parent) είτε με έννοιες χαμηλότερου επιπέδου (children).

4. Τα μέτρα τα οποία θα υιοθετηθούν θα πρέπει να είναι ανεξάρτητα από το περιβάλλον περιγραφής της οντολογίας καθώς και από τη γλώσσα περιγραφής της.

5. Τα μέτρα τα οποία θα υιοθετηθούν θα πρέπει να λαμβάνουν υπόψιν τόσο γλωσσικού τύπου πληροφορία η οποία μπορεί να εξαχθεί από τους γράφους, όσο και δομικού τύπου πληροφορία για αυτές. Με αυτόν τον κανόνα εξασφαλίζεται το δυνατόν περισσότερο η συσχέτιση περισσοτέρων εννοιών και από τις δύο παραπάνω κατηγορίες.

Η έλλειψη συμμόρφωσης σε τέτοιους περιορισμούς, κατά τον ορισμό των μέτρων, μπορεί να οδηγήσει σε έλλειψη υπάρχουσας πληροφορίας κατά τη διάρκεια της συσχέτισής της.

Οι οντολογίες σαν δομές, εκ κατασκευής, έχουν τη δυνατότητα να αποθηκεύσουν τη δομή (structure) για τα δεδομένα (data instances). Επιπλέον διαθέτουν την υποδομή για μεταδεδομένα (metadata) τα οποία εύκολα μπορούν να διαχειριστούν, έτσι ώστε να περιγράφουν, κατά απαίτηση ένα ευρύ φάσμα δεδομένων. Στη συνέχεια, οι οντολογίες υποστηρίζουν τη λογική μέσω μηχανισμού, με την οποία μπορούν να επιλυθούν οι σημασιολογικές διαφοροποιήσεις πάνω σε ιδέες ή παρόμοιες έννοιες (conflicts). Κατ' αυτόν τον τρόπο, ο μηχανισμός των οντολογιών χρησιμοποιείται για την απεικόνιση γνώσης και σε αυτό θα εστιαστεί στο μέλλον η εξόρυξη δεδομένων περιγεγραμμένων σημασιολογικά.

1.4.1. ΔΙΑΛΕΙΤΟΥΡΓΙΚΟΤΗΤΑ

Για να επιτευχθεί η σημασιολογική διαλειτουργικότητα σε ένα ετερογενές πληροφοριακό σύστημα ή μεταξύ διαφορετικών ετερογενών συστημάτων, η έννοια της πληροφορίας η οποία ανταλλάσσεται, πρέπει να είναι αμοιβαία κατανοητή σε όλα τα επιμέρους προς διαλειτουργία συστήματα. Σημασιολογικές διαφορές συμβαίνουν και εμφανίζονται, όταν δύο ή περισσότερα περιεχόμενα δεν χρησιμοποιούν το ίδιο σύστημα αναφοράς ερμηνείας των. Για παράδειγμα, το «beatle», σε τρία συστήματα εννοιολογικής ερμηνείας, το ζωικό βασίλειο, τον χώρο του αυτοκινήτου και τον χώρο της μουσικής, έχει με ορισμένα κοινά χαρακτηριστικά (attributes) όπως το χρώμα, την ταχύτητα, το δέρμα, κλπ αλλά διαφοροποιούνται τελείως σε σχέση με άλλα χαρακτηριστικά όπως τον τροχό ή το πέλμα, ή τα φτερά ή τον τύπο της παράστασης. Επίσης, υπάρχουν και ορισμένα τα οποία ξεκάθαρα δεν μπορούν να ενταχθούν στο ένα ή στο άλλο π.χ. «μήκος ουράς» κλπ, των οποίων το εύρη τιμών είναι ξένα μεταξύ τους, δηλαδή δεν περιλαμβάνουν κοινές τιμές.

Ο Goh ορίζει τρεις κύριες αιτίες οι οποίες είναι υπεύθυνες για τέτοιου είδους αντικρουόμενες σημασιολογικά διαφοροποιήσεις – ασυμβατότητες (conflicts) (Cheng Hian Goh, 1997).

(α) *Confounding Conflicts:* Αυτές οι συγχύσεις εννοιών συμβαίνουν όταν τμήματα της πληροφορίας εμφανίζονται να έχουν την ίδια έννοια, αλλά διαφέρουν στην πραγματικότητα σημασιολογικά, επειδή λόγου χάριν, διαφέρουν τα προσωρινά περιεχόμενα στα οποία αναφέρονται.

(β) *Scaling Conflicts:* Αυτές συμβαίνουν όταν διαφορετικά συστήματα αναφοράς χρησιμοποιούνται για να μετρήσουν μία τιμή. Παραδείγματα τέτοια είναι τα διαφορετικά νομίσματα ή οι διαφορετικού τύπου βαθμολόγηση ιδίων αξιών (τύπος ή / και κλίμακα).

(γ) *Naming Conflicts*: Αυτές συμβαίνουν όταν σχήματα ονοματολογίας (naming schemes) της ίδιας πληροφορίας, διαφέρουν σημαντικά. Ένα συχνό φαινόμενο είναι η χρήση ομώνυμων και συνώνυμων εννοιών, σε διαφορετικά συστήματα αναπαράστασης γνώσης.

Ο κύριος σκοπός της χρήσης των οντολογιών στο σημείο αυτό είναι η ανταλλαγή δεδομένων να περιγράφεται όχι μόνο με βάση τη συντακτική τους δομή αλλά και με βάση μια κοινή εννοιολογική δομή αλληλοκατανόησης που να είναι σε θέση να επιλύσουν τις τρείς παραπάνω αιτίες σημασιολογικής αντικρουόμενων σημασιολογικών αντιθέσεων (semantic conflicts). Ειδικότερα, για τις τρέχουσες και μελλοντικές εφαρμογές του διαδικτύου με τη μορφή WEB 2.0 - σημασιολογικού ιστού, τα συστήματα ανακάλυψης, ανάκτησης και ανταλλαγής δεδομένων θα εγκαταλείπουν ολοένα και περισσότερο τους καθημερινούς τρόπους οι οποίοι βασίζονται σε ανταλλαγή σχημάτων βάσεων δεδομένων περιγραφικών μη δυναμικών αρχείων των πληροφοριών που είναι αποθηκευμένες σε αυτές λόγω αυτής ακριβώς της αδυναμίας τους. Την αντικατάσταση αυτών των μηχανισμών αναλαμβάνουν με την πάροδο του χρόνου “έξυπνα συστήματα” τα οποία είναι ικανά, ενσωματώνοντας λογική να αναλύουν σημασιολογικά το περιεχόμενο σε πραγματικό χρόνο, με βάση δομές-οντολογίες μεταδεδομένων, αυτοματοποιημένα. Κατ’ αυτόν τον τρόπο, θα προκύπτουν πιο αποδοτικές λύσεις ανταλλαγής δεδομένων και ανάκτησης γνώσης. Παρακάτω παρουσιάζονται δύο από τα πιο σημαντικά θέματα που μπορεί να οριοθετήσουν πιθανά προβλήματα όσον αφορά την ανταλλαγή δεδομένων και την εννοιολογική διαλειτουργικότητα συστημάτων, η σημασιολογική διαφοροποίηση και η προτυποποίηση-διαφοροποίηση.

1.4.2. ΣΗΜΑΣΙΟΛΟΓΙΚΗ ΔΙΑΦΟΡΟΠΟΙΗΣΗ

Με την εισαγωγή των οντολογιών σαν μηχανισμών αναπαραστάσεως γνώσης και ανταλλαγής δεδομένων, τόσο μεταξύ χρηστών όσο και μεταξύ έξυπνων εφαρμογών λογισμικού πρέπει να αντιμετωπίζεται και να επιλύεται κατάλληλα το θέμα της διαφοροποίησης σημασιολογικά των περιγεγραμμένων σε αυτές εννοιών.

Το ζήτημα αυτό ανακύπτει λόγω του ότι ενδιάμεσοι φορείς λογικής και κατανόησης (λογισμικού με τη μορφή agents η ανθρώπων) έχουν διαφορετική αντίληψη και αναπαράσταση των γνωσιακών περιοχών στις οποίες απευθύνονται. Αυτό είναι ένα κρίσιμο σημείο δεδομένου ότι η αντιμετώπιση του επιτρέπει την ευκολότερη γενίκευση η εξειδίκευση της γνώσης που πρόκειται να ανταλλαγεί.

Κατ' αυτόν τον τρόπο , ιδιαίτερα σε εφαρμογές που στηρίζονται σε έξυπνο λογισμικό (intelligent agents) συστηματοποιείται η αλληλοκατανόηση τέτοιων οντοτήτων μέσω της εγγύησης της ποιότητας της προσφερόμενης, σημασιολογικής υπηρεσίας (semantic quality of service).

1.4.3. ΠΡΟΤΥΠΟΠΟΙΗΣΗ & ΔΙΑΦΟΡΟΠΟΙΗΣΗ

Μια σημαντική εφαρμογή με τις μέχρι σήμερα εφαρμογές είναι η εφαρμογή της προτυποποίησης στα πλαίσια της ανταλλαγής δεδομένων. Αυτό είναι ιδιαίτερα χρήσιμο όταν θεωρούμε ότι η γνώση είναι κάτι σχετικό και περιγράφει ένα γνωσιακό χώρο από ένα συγκεκριμένο σημείο αναφοράς. Οι σύγχρονες όμως εφαρμογές ανταλλαγής γνώσης, απαιτούν τη συνεχή διαφοροποίηση του σημείου από το οποίο το εκάστοτε γνωσιακό επίπεδο θεωρείται.

Αυτό έχει σαν αποτέλεσμα τη μη δυνατή ουσιαστικά προτυποποίηση της δομής της περιγραφόμενης γνώσης, ιδιαίτερα όταν αναφερόμαστε σε κατανεμημένες πηγές από χρήστες που δεν είναι δυνατόν να συνεργάζονται στην προτυποποίηση (λόγω ανταγωνισμού κλπ) και τέλος από τη συνεχή αλλαγή του περιεχομένου δεδομένων (data). Σε τέτοιες περιπτώσεις (Alexander Maedche and Steffen Staab, 2001), η διαφοροποίηση αυξάνει και η προτυποποίηση μειώνεται και η λύση δίνεται μέσω της τεχνολογίας σύγκρισης οντολογιών με το να βρεθεί πως σχετίζονται σημασιολογικά αυτές.

1.5. ΜΕΘΟΔΟΙ ΕΞΟΡΥΞΗΣ ΓΝΩΣΗΣ

Στα πλαίσια της παρούσας Δ.Δ., σημαντικό κομμάτι ερευνητικής προσπάθειας συντελείται και στο να εξευρεθούν - εξορυχθούν στοιχεία με τη μορφή μεταδεδομένων των ανεξαρτήτων μεταξύ τους χρηστών, τα οποία εκ πρώτης όψεως δεν σχετίζονται μεταξύ τους, αλλά μετά την προτεινόμενη μεθοδολογία θα δημιουργηθούν ισχυρές συσχετίσεις μεταξύ τους. Ο τομέας αυτός έρευνας αποτελεί αντικείμενο της διαδικασίας εξόρυξης γνώσης και δεδομένων (data mining) και εξετάζεται κατωτέρω.

Η εξόρυξη δεδομένων είναι μία αναλυτική διαδικασία που έχει σχεδιαστεί και εφαρμόζεται με διαφορετικές μορφές για να εξερευνεί πολύ μεγάλο όγκο δεδομένων, κυρίως επιχειρηματικών, η άλλων που σχετίζονται με αυτά. Η εξερεύνηση αυτή γίνεται ψάχνοντας συστηματικά μεταξύ των μεταβαλλόμενων δεδομένων με σκοπό να ορίσει πρωταρχικές σχέσεις (relationships) μεταξύ αυτών, κατόπιν να κατηγοριοποιήσει και τελικά να επαληθεύσει τις συσχετίσεις και κατηγοριοποιήσεις αυτές. Στο τέλος το επαληθευμένο μοντέλο που προκύπτει από την ανωτέρω διαδικασία εφαρμόζεται σε κάθε νέα πληροφορία που εισέρχεται στο σύστημα για την κατάλληλη ένταξη και συσχέτιση σε αυτό.

Οι κύριες φάσεις είναι οι εξής :

1. Ανακάλυψη – Ελευθέρωση των δεδομένων (extracting)

   Στη φάση αυτή γίνεται όλη η διαδικασία εκκκαθάρισης των δεδομένων και άλλων σχετικών διαδικασιών όπως cleaning data, data transformation, subsets of data και ορισμός προς παρατήρηση πεδίων (fields)

2. Δημιουργία Μοντέλων και Επαλήθευση αυτών (Model Building and Validation)

   Κατά τη φάση αυτή δημιουργούνται διάφορα μοντέλα κατόπιν λεπτομερούς ανάλυσης των βάσεων δεδομένων. Στη συνέχεια, αυτά εισάγονται στο σύστημα όπου τα αποτελέσματα που επιστρέφονται αξιολογούνται και υιοθετείται η εκάστοτε καταλληλότερη μέθοδος

3. Εφαρμογή (Deployment)

   Η καταλληλότερη μέθοδος όπως προέκυψε, εφαρμόζεται σε αυτή τη φάση, σε νέα στοιχεία που εισέρχονται στο σύστημα.

Οι τρόποι και οι αλγόριθμοι με τους οποίους πραγματοποιείται η εξόρυξη γνώσης χωρίζονται σε δύο μεγάλους τομείς:

α) τρόποι οι οποίοι βασίζονται σε αυστηρά μαθηματικά και αλγοριθμικά πρότυπα και

β) τρόποι οι οποίοι αντανακλούν τις φυσικές διεργασίες και τον τρόπο με τον οποίο η γνώση και η πληροφορία μεταφέρονται και αποκτώνται στην φύση καθεαυτή.

Κρίνεται σκόπιμο εφ' όσον αναφερόμαστε σε χώρους πολύ πολύπλοκους, ως προς την αναπαράστασή τους, να εξετασθούν και οι μεθοδολογίες οι οποίες εμπνεύστηκαν από τις ανωτέρω φυσικές διεργασίες.

Στους (α) θα μπορούσε να εντάξει κανείς:

1. Τους κανόνες συσχέτισης. Σκοπός είναι ο εντοπισμός συσχετίσεων μεταξύ χαρακτηριστικών.
2. Τούς κανόνες ταξινόμησης. Σκοπός είναι η πρόβλεψη μίας διακριτής διαδικασίας.
3. Τούς κανόνες ομαδοποίησης. Σκοπός είναι η ανάδειξη ομοίων υποδειγμάτων.

Στους (β) θα μπορούσε να εντάξει κανείς την νοημοσύνη η οποία αναπτύσσεται σε όντα με ίδια ή παραπλήσια ενδιαφέροντα, διαφόρων τύπων. Σε αυτή την κατηγορία γίνεται η μοντελοποίηση των προφίλ των ανθρώπων σε σχέση με συγκεκριμένα ενδιαφέροντα ή δραστηριότητες, εκμεταλλεύσιμες ή μη επιχειρηματικά. Επίσης στους (β) εντάσσεται και η ενστικτώδης αλλά πολύ ενδιαφέρουσα εξυπνάδα (intelligence) ζώντων οργανισμών όπως εντόμων (ants), πτηνών, σμηνών και σχηματισμών τους (swarms & flocks) τα οποία διαμορφώνουν αυτό το οποίο ονομάζεται collective intelligence (CI) ή εξυπνάδα ή λογική ομάδος.

Σε τέτοιου είδους συστήματα, γίνεται μοντελοποίηση της γνώσης σαν διαδικασία ανεύρεσης τροφής επί το συνηθέστερον, σε κάποιο χώρο n-διαστάσεων. Για την ακριβέστερη μοντελοποίηση του προβλήματος, υπεισέρχονται εκτός των διαστάσεων του χώρου, σαν παράμετροι, στον οποίο καλείται να επιλυθεί το συγκεκριμένο πρόβλημα, και

άλλοι όπως ο τρόπος διάδοσης της πληροφορίας ανεύρεσης ή μη αυτής. Παρακάτω ακολουθεί παρουσίαση αυτών των μεθοδολογιών.

Μία ενδιάμεση προσέγγιση έξυπνου λογισμικού είναι και αυτή των ευφυών διαμεσολαβητών (Intelligent agents) που χρησιμοποιούνται για τη διανομή (brokerage – mediation) με έξυπνο τρόπο της σχετικής γνώσης και πληροφορίας.

1.5.1. ΕΞΥΠΝΟΙ ΔΙΑΜΕΣΟΛΑΒΗΤΕΣ

Στο σημείο αυτό παρουσιάζονται μέσω συγκεκριμένης μεθοδολογίας οι έξυπνοι διαμεσολαβητές και ο τρόπος με τον οποίο μπορούν να χρησιμοποιηθούν για την αναγνώριση και δυναμική διανομή περιεχομένου, σε κατάλληλους ρόλους – χρήστες. Το τμήμα αυτό της εργασίας εκτελέστηκε στα πλαίσια του ερευνητικού έργου QUALEG (EU-IST-2002-507767). Το QUALEG σαν ερευνητικό έργο επικεντρώθηκε στην επίλυση του προβλήματος που αντιμετωπίζεται σε πανευρωπαϊκό επίπεδο και αφορά την μείωση της συμμετοχής των πολιτών στο πολιτικό γίγνεσθαι. Για το σκοπό αυτό τέθηκε ως στόχος η αύξηση της συμμετοχής αυτής, μέσω της ανάπτυξης ενός μηχανισμού συνεχούς παρακολούθησης των διαδικασιών και της ποιότητας τους, εκφρασμένων σε δείκτες όπως η υποβολή αιτημάτων, η δυναμική απάντησή τους, η δημιουργία πολιτικών, το επίπεδο ικανοποίησης των πολιτών για τις προσφερόμενες υπηρεσίες από τις τοπικές αρχές (local governments) κλπ. Για την επίτευξη του στόχου αυτού, εκτός από την μεθοδολογία αναπτύχθηκε και καινοτόμο λογισμικό βασισμένο στην τεχνολογία των έξυπνων διαμεσολαβητών, το οποίο ήταν υπεύθυνο για την εξαγωγή γνώσης μέσα από κείμενα (κυρίως emails), την συσχέτιση της γνώσης αυτής σε ειδική οντολογία e-government καθώς και με διαδικασίες ροής εργασιών (workflow) περιγεγραμμένων στην γλώσσα WSDL. Τέλος υλοποιήθηκαν και εργαλεία απόκτησης γνώσης από documents και ακολούθως η εξαγωγή γνώσης παρομοίως από αυτά μέσω μηχανισμών text mining.

1.5.1.1. ΕΙΣΑΓΩΓΗ

Η ολοκλήρωση δεδομένων είναι η διαδικασία συνδυασμού δύο ή περισσότερων συνόλων δεδομένων για κατανομή και ανάλυση με σκοπό την υποστήριξη διαχείρισης των δεδομένων. Οι ευφυείς διαμεσολαβητές είναι αυτόνομα, ή ημι-αυτόνομα pro-active και

reactive υπολογιστικά λογισμικά. Παρόλο που υπάρχει μια τεράστια συλλογή ερευνών στην ολοκλήρωση δεδομένων, αυτή η έρευνα έχει μικρή επίδραση στα state-of-the-art συστήματα που χρησιμοποιούν ευφυείς διαμεσολαβητές. Πιστεύουμε ότι αυτό το χάσμα μπορεί να εξηγηθεί στο γεγονός ότι οι περισσότερες προσεγγίσεις βασίζονται στη σημασιολογική προσαρμογή που θα επιλυθεί πρώτη (πιθανώς με το χέρι), πριν απευθυνθούμε στα πιο «τεχνικά» θέματα της ολοκλήρωσης. Όμως, οι ερευνητές και οι επαγγελματίες ομοίως έρχονται να αντιληφθούν ότι δεν μπορεί να υπάρξει λύση στην παράδοση της ολοκληρωμένης πληροφορίας εκτός εάν το πρόβλημα της σημασιολογικής ετερογένειας αντιμετωπιστεί άμεσα (A. M. Ouksel and A.P. Sheth, 1999). Το παράδειγμα εργάζεται προς αυτό το στόχο με τη χρήση οντολογιών.

1.5.1.2. Ολοκληρωση Δεδομενων

Η ολοκλήρωση δεδομένων είναι ένα κοινό θέμα στον κόσμο της πληροφορικής. Όπως η πληροφορία όλο και περισσότερο γίνεται πιο πολύπλοκη και τεράστια και η διαχείριση της πληροφορίας περισσότερο κρίσιμη, η ανάγκη για εξακρίβωση της ολοκλήρωσης των δεδομένων και η αντιγραφή τους είναι το κλειδί στην αξιόπιστη λειτουργία του πληροφορικού συστήματος.

Πολλές τεχνικές έχουν εφαρμοστεί για την προώθηση της διαδικασίας ολοκλήρωσης των δεδομένων, όπως η ολοκλήρωση event-based λογισμικού (D. Barret, L. Clarke, P. Tarr, and A. Wise, 1996), η ολοκλήρωση σχήματος βάσης δεδομένων (C. Batini, M. Lenzerini, and S. Navathe, 1986), ολοκλήρωση πληροφορίας βασισμένης στο Διαδίκτυο (C.A. Knoblock, S. Minton, J.L. Ambite, N. Ashish, I. Muslea, A. Philpot, and S. Tejada, 2001), και η σημασιολογική ολοκλήρωση (S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano, 2001). Το πεδίο αυτό έχει δει την υλοποίηση πολλών εργαλείων όπως το DIKE (L. Palopoli, L.G. Terracina, and D. Ursino., 2000), Clio (R. J. Miller, M.A. Hern`andez, L.M. Haas, L.-L. Yan, C.T.H. Ho, R. Fagin, and L. Popa, 2001), Cupid (J. Madhavan, P.A. Bernstein, and E. Rahm., 2001), OntoBuilder (G. Modica, A. Gal, and H. Jamil., 2001) για να ονομάσουμε μερικά.

1.5.1.3. Έξυπνοι Διαμεσολαβητες Λογισμικου

Στο σημερινό κόσμο, με την εξάπλωση των υπολογιστών, οι διαμεσολαβητές λογισμικού είναι απαραίτητοι για την προώθηση της αποτελεσματικής εκμετάλλευσης από τον χρήστη των συστημάτων λογισμικού. Οι έξυπνοι διαμεσολαβητές χρησιμοποιούνται για να ξεκινήσουν επικοινωνίες, να παρακολουθούν γεγονότα, και να εκτελούν διαδικασίες που βοηθούν χρήστες για να κατανοήσουν τον τεχνικά πολύπλοκο κόσμο.

Υπάρχουν έξυπνοι διαμεσολαβητές λογισμικού για την πρόσβαση σε διαδικτυακά συστήματα πληροφοριών μέσω κινητών συσκευών (A. C. Ramos, J. Gensel, M. Villanova-Oliver, and H. Martin, 2005). Υπάρχουν επίσης συστήματα πολλαπλών πρακτόρων λογισμικού που βοηθούν τους επενδυτές με τις επενδύσεις τους στα χρηματιστήρια (S. C. Sundararajan, S. Sankarlal, and A. Kumar. , 2005).

Έχουν αναλυθεί ευφυείς διαμεσολαβητές για εξόρυξη κειμένου για ενέργειες στο δίκτυο (Y. Kusumura, Y. Hijikata, and S. Nishida, 2004). Επίσης έχουν εισαχθεί ευφυείς διαμεσολαβητές σε ψηφιακές βιβλιοθήκες, όπως στην Ψηφιακή Βιβλιοθήκη του Πανεπιστημίου του Μίτσιγκαν (W. P. Birmingham, E. H. Durfee, T. Mullen, and M. P. Wellman, 1995). Σε αυτό το δικό μας παράδειγμα προτείνουμε ένα πλαίσιο εργασίας και μια αρχιτεκτονική για ευφυείς διαμεσολαβητές για την ολοκλήρωση δεδομένων.

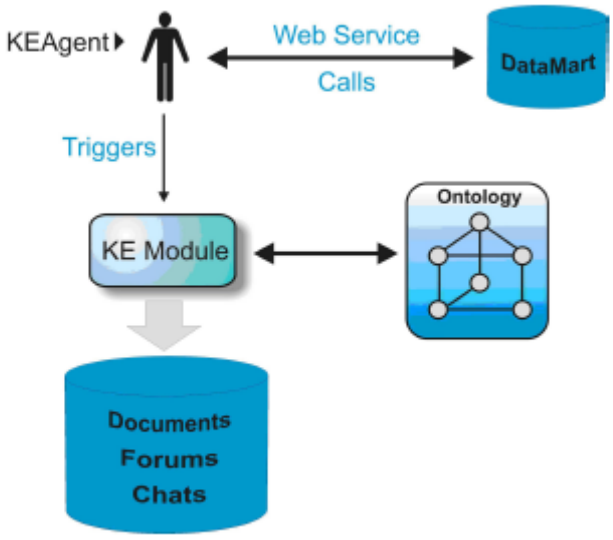
1.5.1.4. Αρχιτεκτονικη

Στο παράδειγμά μας (A. Gal, A. Segev, C. Tatsiopoulos, K. Sidiropoulos, and P. Georgiades, 2005) η ολοκλήρωση του γλωσσικού περιβάλλοντος σε μια έννοια οντολογίας εφαρμόζεται στις διαδικασίες της δρομολόγησης e-mail και στην ανάλυση γνώμης.

Δρομολόγηση e-mail: Ο χρήστης παρέχει στο σύστημα ένα κατώφλι απόστασης t_1. Οποιαδήποτε έννοια της οντολογίας που ταιριάζει με τα συμφραζόμενα, που αυτόματα αναπαράγεται από ένα e-mail, και η απόστασή του είναι μικρότερη από το κατώφλι ($d(A,B)< t_1$) θεωρείται σχετική και το e-mail δρομολογείται κατάλληλα.

Ανάλυση γνώμης: Προσδιορίζεται ένα σχετικό σύνολο εννοιών οντολογιών, παρόμοιο με τη δρομολόγηση e-mail. Τότε για κάθε έννοια της οντολογίας, υπολογίζεται η σχετική απόσταση των διαφορετικών γνωμών αυτής της έννοιας. Αν η διαφορά στην απόσταση είναι πολύ κοντινή (δεδομένου ενός πρόσθετου κατωφλίου t_2), το σύστημα αποφεύγει να

δώσει μια γνώμη (και το e-mail δρομολογείται κατάλληλα). Σε άλλη περίπτωση, το e-mail χαρακτηρίζεται από τη γνώμη με την μικρότερη απόσταση. Υπάρχουν διαφορετικοί ευφυείς διαμεσολαβητές στο σύστημα, ταξινομημένοι σύμφωνα με τις διαδικασίες τους. Έχουν υλοποιηθεί σε JADE πλατφόρμα και είναι σύμφωνοι με τις προδιαγραφές FIPA για διαλειτουργικά συστήματα πολλαπλών ευφυών πρακτόρων λογισμικού. Είναι οι ακόλουθοι:



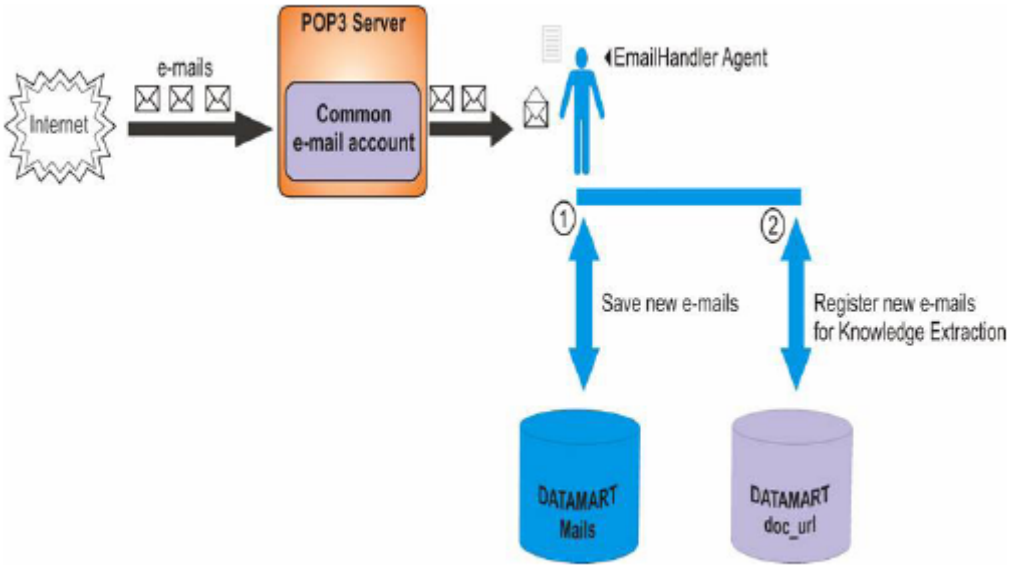
Σχ.: Αρχιτεκτονική του πράκτορα λογισμικού KEAgent

Knowledge Extraction Agent: Όπως φαίνεται στο σχήμα έχει την αρμοδιότητα να ξεκινά το Knowledge Extraction Module, έτσι ώστε να αναλύεται συχνά το περιεχόμενο της αποθηκευμένης πληροφορίας. Υπάρχουν πολλών ειδών κείμενα που θα μπορούσαν να αναλυθούν: κείμενα σε φόρουμ, σε chat, σε μηνύματα εισερχόμενων e-mail. Ειδικότερα αυτός ο πράκτορας λογισμικού εκτελεί περιοδικές έρευνες στις βάσεις δεδομένων του συστήματος για να δει αν υπάρχει νέα πληροφορία για να αναλυθεί. Κάθε συναλλαγή με τις βάσεις δεδομένων πραγματοποιείται μέσω Web Services. Αν βρεθούν νέα κείμενα, ο πράκτορας λογισμικού ξεκινά το KE Module. Αυτό εκτελεί την αντιστοίχηση με αναφορά σε μια οντολογία, που προσδιορίζει το σύνολο των εννοιών και των σχέσεών τους. Μετά από αυτή τη διαδικασία, αποθηκεύεται ένα σύνολο λέξεων-κλειδιών σε μια βάση.

Opinion Analysis Agent: Παρόμοια με τον KE Agent, ο OA Agent συχνά ψάχνει στις βάσεις δεδομένων για να βρει ποια κείμενα πρέπει να αναλυθούν από το OA Module. Όταν

βρεθούν τα κείμενα που χρειάζονται ανάλυση, ο πράκτορας λογισμικού ξεκινά τον αλγόριθμο ανάλυσης γνώμης πάνω σε αυτά και η έξοδος του είναι μια έννοια οντολογίας και μια λίστα λέξεων.

Email Handler Agent: Όλα τα εισερχόμενα e-mail που έχουν σταλθεί σε ένα κοινό λογαριασμό e-mail συλλέγονται από ένα πράκτορα λογισμικού. Ο Email Handler Agent που φαίνεται στο παρακάτω σχήμα κόβει το κάθε e-mail στα ξεχωριστά του κομμάτια και παίρνει το κείμενο που περιέχεται σε αυτό. Ειδικότερα σώζει πληροφορία που αφορά το e-mail, όπως sender, subject, body κ.λ.π.



Αρχιτεκτονική του συστήματος Βασισμένου σε Intelligent Agents

1.5.1.5. ΣΥΜΠΕΡΑΣΜΑΤΑ

Στο παράδειγμα μας υλοποιήθηκαν 2 ευφυείς διαμεσολαβητές, ξεχωρίζοντας τη διαδικασία της εξόρυξης γνώσης από κείνη της ανάλυσης γνώμης. Η κυριότερη διαφορά είναι ότι ο πράκτορας λογισμικού εξόρυξης γνώσης αποφεύγει την συγκεκριμένη υλοποίηση που βασίζεται στη γλώσσα και βασίζει τις τεχνικές ανάλυσης στη χρήση μεγάλης συλλογής από σχετικά κείμενα τα οποία ανακτώνται από το Διαδίκτυο, ενώ αυτός της ανάλυσης γνώσης

χρησιμοποιεί τεχνικές από την IR και την NLP για τη βελτίωση της κατανόησης του περιεχομένου.

Η ολοκλήρωση των δεδομένων είναι ένα σημαντικό πεδίο στη διαχείριση των συστημάτων πληροφορικής σήμερα. Η χρήση αυτόνομων ή ημι-αυτόνομων πρακτόρων λογισμικού μπορεί αποδοτικά να προωθήσει τη διαδικασία της ολοκλήρωσης των δεδομένων. Και σε αυτό το παράδειγμα παρουσιάσαμε ένα πλαίσιο εργασίας για ολοκλήρωση δεδομένων που βασίζεται σε ευφυείς διαμεσολαβητές και πως αυτοί οι ευφυείς διαμεσολαβητές υποστηρίζουν αυτή τη διαδικασία.

1.5.2. ΚΑΝΟΝΕΣ ΣΥΣΧΕΤΙΣΗΣ - ASSOCIATION RULES

Οι κανόνες συσχέτισης (association rules) είναι μία από τις σημαντικότερες τεχνικές εξόρυξης από δεδομένα στον τομέα της εξόρυξης δεδομένων (data mining). Ειδικότερα, η μάθηση μέσω των κανόνων συσχέτισης (association rule learning) με σκοπό την ανακάλυψη ενδιαφέρων συσχετίσεων μεταξύ μεταβλητών σε πάρα πολύ μεγάλο όγκο δεδομένων. Τέτοια δεδομένα μπορεί να προκύπτουν είτε από εξειδικευμένες εμπορικές εφαρμογές είτε από τη χρήση των δεδομένων που βρίσκονται ή προκύπτουν από τα δεδομένα του διαδικτύου. Για παράδειγμα, μια διαφημιστική εταιρεία θα μπορούσε να χρησιμοποιήσει αυτή τη μεθοδολογία από την ανάλυση αγορών σε μία δεδομένη επιχείρηση από τους πελάτες της, με σκοπό τη στοχευμένη προώθηση, δηλαδή την όσο το περισσότερο cost-effective διαφήμισή τους.

Όμως το πιο ενδιαφέρον θα ήταν η ανακάλυψη κανόνων οι οποίοι να δημιουργούν και να περιγράφουν το προφίλ του χρήστη μέσα από τη χρήση του Διαδικτύου, όπου αυτό θα μπορούσε να μοντελοποιηθεί σαν μία δενδρική εννοιολογική δομή εννοιών.

Η παρούσα Δ.Δ. προτείνει σε αυτό το σημείο μια καινοτόμο μεθοδολογία για την ανίχνευση πλησιέστερων εννοιών μεταξύ δύο Οντολογιών, με την χρήση κανόνων συσχέτισης και ειδικότερα του Αλγόριθμου APRIORI.

1.5.2.1. ΠΑΡΑΔΕΙΓΜΑ ΧΡΗΣΗΣ ΑΛΓΟΡΙΘΜΟΥ APRIORI

Η κυριότερη εφαρμογή τους είναι η ανάλυση του καλαθιού της νοικοκυράς (market-basket analysis), όπου η αγορά ενός συνόλου προϊόντων (για παράδειγμα το περιεχόμενο ενός καλαθιού σούπερ - μάρκετ) αντιμετωπίζεται ως μία μεμονωμένη συναλλαγή. Σκοπός είναι η εύρεση τάσεων (αλληλεξαρτήσεων/ συσχετίσεων) μεταξύ ενός αριθμού συναλλαγών, οι οποίες στη συνέχεια μπορούν να αναλυθούν και μέσω αυτών να αναγνωρισθούν αγοραστικά πρότυπα, τα οποία στη συνέχεια θα χρησιμοποιηθούν ως πληροφορία (Chara Kostakis, 2007). Η έννοια των κανόνων συσχέτισης εισήχθη το 1993 και αναφερόταν στην εξαγωγή συσχετίσεων στα πεδία βάσεων δεδομένων. Έκτοτε, έχει τύχει ευρείας έρευνας και εφαρμογής.

Ένας κανόνας συσχέτισης είναι μία έκφραση της μορφής X=>Y (X implies Y), όπου X και Y είναι σύνολα τιμών των πεδίων. Η σημασία ενός τέτοιου κανόνα είναι ότι οι περισσότερες εγγραφές της βάσης δεδομένων που περιέχουν το X, περιέχουν και το Y. Σκοπός είναι η εύρεση συσχετίσεων μεταξύ των στηλών ενός συνόλου (πίνακα) δεδομένων. Η σημαντικότητα ενός κανόνα συσχέτισης καθορίζεται από το ποσοστό εφαρμογής του κανόνα στο σύνολο της βάσης δεδομένων. Ένας κλασικός κανόνας συσχέτισης είναι για παράδειγμα της μορφής: *«Το 70% των καταναλωτών που αγοράζουν ψωμί, αγοράζουν και βούτυρο, όπως επίσης και στο 60% των αγορών έχουν αγορασθεί ψωμί και βούτυρο».* Έτσι, κάθε κανόνας συσχέτισης έχει δύο μεταβλητές μέτρησης, την επιβεβαίωση (support) και την αξιοπιστία (confidence). Η αξιοπιστία μετράει ουσιαστικά την ισχύ του κανόνα, δηλαδή είναι το ποσοστό των συναλλαγών που περιέχουν το Y επί του αριθμού των συναλλαγών που περιέχουν το X. Αντιπροσωπεύει, δηλαδή, τη δεσμευμένη πιθανότητα του Y δεδομένου του X. Η επιβεβαίωση αναφέρεται ουσιαστικά σε στατιστική σημαντικότητα και είναι το ποσοστό που εμφανίζονται το X και το Y μαζί επί του συνόλου της βάσης δεδομένων (πόσο συχνά δηλαδή συμβαίνει το πρότυπο αυτό στη βάση δεδομένων). Το πρόβλημα εύρεσης κανόνων συσχέτισης αναφέρεται στην εύρεση εκείνων των κανόνων που έχουν μία καθορισμένη από το χρήστη ελάχιστη επιβεβαίωση και αξιοπιστία.

Βασικά προβλήματα που αφορούν στην εύρεση κανόνων συσχέτισης αναφέρονται στην πολυπλοκότητα των αλγορίθμων και στη δυσκολία επιλογής των χρήσιμων κανόνων από το σύνολο των κανόνων που προκύπτουν. Το πρώτο πρόβλημα αναφέρεται στο γεγονός ότι ο αριθμός των κανόνων αυξάνεται εκθετικά με τον αριθμό των πεδίων. Οι σημερινοί

αλγόριθμοι που εξάγουν κανόνες συσχέτισης μπορούν να μειώσουν αποτελεσματικά τον αριθμό αυτόν, βασισμένοι σε κατώφλια (thresholds), δηλαδή την επιβεβαίωση και την αξιοπιστία, για μέτρηση της ποιότητας των κανόνων. Το δεύτερο πρόβλημα αναφέρεται στο γεγονός ότι οι χρήσιμοι κανόνες που συνήθως προκύπτουν είναι μόνο ένα μικρό ποσοστό του συνόλου των κανόνων. Το πρόβλημα αυτό ερευνάται σε σχέση με την υποστήριξη προς το χρήστη (user support), όταν αναζητά κανόνες μέσα στους κανόνες που έχουν εξαχθεί, καθώς και με την ανάπτυξη επιπλέον μέτρων ποιότητας στους κανόνες.

1.5.2.2. APRIORI ΑΛΓΟΡΙΘΜΟΣ

Ο αλγόριθμος APRIORI παίρνει το όνομά του από το ότι από ένα πολύ μεγάλο πλήθος δεδομένων μπορεί να ανακαλύψει και να συσχετίσει μεταξύ τους μεταβλητές οι οποίες δεν είναι εκ των προτέρων γνωστές (apriori), εν αντιθέσει με τις μεθοδολογίες που είναι βασισμένες στην στατιστική ανάλυση και πρέπει εκ των προτέρων να δηλωθεί η συσχέτιση, για την οποία το σύστημα πρέπει να ανακαλύψει την τιμή αληθείας της (ναι – όχι – σε ποιο ποσοστό υπάρχει εξάρτηση).

Στη συνέχεια περιγράφεται αναλυτικά ο βασικότερος αλγόριθμος συσχέτισης, ο Apriori. Ο αλγόριθμος Apriori παρουσιάστηκε αρχικά το 1994 από τον Agrawal. Σχεδόν όλοι οι αλγόριθμοι συσχέτισης βασίζονται στην αρχική ιδέα του, και τον χρησιμοποιούν τα περισσότερα εμπορικά συστήματα εξόρυξης από δεδομένα που υπάρχουν αυτή τη στιγμή στην αγορά.

Υποθέτουμε ότι $X = \{x_1, \ldots\ldots, x_n\}$ είναι ένα σύνολο από προϊόντα. Έστω *D* ένα σύνολο από αγορές, όπου κάθε αγορά *T* είναι μία λίστα από προϊόντα, όπου $T \subseteq X$. Κάθε υποσύνολο από προϊόντα $X \cup X$ ονομάζεται λίστα προϊόντων (itemset). Ο κανόνας συσχέτισης είναι μία έκφραση της μορφής X=>Y, όπου $X \subset X$, $Y \subset X$ και $X \cap Y = 0$. Ο κανόνας X=>Y, σε ένα σύνολο από αγορές *D*, ισχύει με αξιοπιστία c, αν το c% των αγορών που ανήκουν στο *D* και περιέχουν το X, περιέχουν επίσης και το Y. Ο κανόνας X=>Y έχει επιβεβαίωση s, αν το s% των αγορών που ανήκουν στο *D* περιέχουν το $X \cup Y$.

Όπως γίνεται φανερό από όσα προαναφέρθηκαν, η αξιοπιστία προκύπτει από την επιβεβαίωση ως εξής: $c(X => Y) = s(X \cup Y) / s(X)$.

Όπως ήδη αναφέρθηκε, ένα βασικό πρόβλημα στην εξαγωγή κανόνων συσχέτισης είναι ο τεράστιος αριθμός κανόνων που εξάγονται. Για την ακρίβεια, ο αριθμός των κανόνων αυξάνεται εκθετικά, καθώς το $|X|$ αυξάνεται. Προκειμένου να μειωθεί ο αριθμός των κανόνων που εξάγονται, εισάγεται ο περιορισμός της ελάχιστης επιβεβαίωσης και της ελάχιστης αξιοπιστίας (misupp και minconf).

Έτσι, το πρόβλημα εύρεσης κανόνων συσχέτισης μπορεί να διαιρεθεί στα δύο υπό-προβλήματα (Agrawal R., Srikant R., 1994):

(1) Εύρεση όλων των συνδυασμών των προϊόντων που έχουν επιβεβαίωση πάνω από την ελάχιστη επιβεβαίωση (η οποία καθορίζεται από το χρήστη). Όλοι αυτοί οι συνδυασμοί ονομάζονται μεγάλες λίστες από προϊόντα (large itemsets) και όλοι οι υπόλοιποι συνδυασμοί μικρές λίστες από προϊόντα (small itemsets).

(2) Εξόρυξη κανόνων συσχέτισης, βασισμένη στις μεγάλες λίστες από προϊόντα, οι οποίες έχουν μεγαλύτερη αξιοπιστία από αυτήν που καθορίζει ο χρήστης.

Η γνώση για τις τιμές της επιβεβαίωσης όλων των υπολογισμών του X διασφαλίζεται μέσω της ιδιότητας της επιβεβαίωσης των προϊόντων: όλα τα υποσύνολα μίας μεγάλης λίστας από προϊόντα πρέπει να είναι και αυτά μεγάλες λίστες από προϊόντα. Έτσι, το πρόβλημα της εξόρυξης κανόνων συσχέτισης εντοπίζεται στην εύρεση των μεγάλων λιστών από προϊόντα, δεδομένης μίας ελάχιστης επιβεβαίωσης.

Η εύρεση των *μεγάλων λιστών* από προϊόντα, για να αποφύγει ένα εξαντλητικό ψάξιμο όλων των δυνατών συνδυασμών, βασίζεται στο ότι: *μία λίστα από προϊόντα είναι μεγάλη λίστα από προϊόντα αν κάθε υποσύνολο της είναι μεγάλη λίστα από προϊόντα.* Ο αλγόριθμος Apriori βρίσκει *μεγάλες λίστες προϊόντων* κάνοντας προσπέλαση πολλές φορές στο σύνολο εκπαίδευσης Τ. Στην πρώτη προσπέλαση υπολογίζεται η *επιβεβαίωση* κάθε διαφορετικού προϊόντος ξεχωριστά και στη συνέχεια καθορίζεται ποία από αυτά είναι *μεγάλες λίστες* από προϊόντα. Σε κάθε επόμενη προσπέλαση, αρχίζουμε έχοντας τις *μεγάλες λίστες* από προϊόντα που βρέθηκαν στην προηγούμενη προσπέλαση. Από αυτές δημιουργούμε νέες πιθανές *μεγάλες λίστες* από προϊόντα, που καλούνται *υποψήφιες (candidate) μεγάλες λίστες* από προϊόντα. Στη συνέχεια μετράμε την ακριβή *επιβεβαίωση* αυτών και καθορίζουμε ποίες είναι οι πραγματικά *μεγάλες λίστες* από προϊόντα. Οι τελευταίες αποτελούν τη βάση για το επόμενο βήμα. Η αποτελεσματικότητα στην εύρεση των *μεγάλων λιστών* από προϊόντα, λόγω μεγάλης πολυπλοκότητας, αποτελεί κριτήριο για την αποτελεσματικότητα συνολικά ενός αλγορίθμου εύρεσης κανόνων συσχέτισης.

Τα βήματα του αλγόριθμου Apriori έχουν ως εξής:

1 Βρες τα προϊόντα που εμφανίζονται περισσότερο από την ελάχιστη επιβεβαίωση (minimum support). δηλαδή το σύνολο Lι = μεγάλες λίστες από 1 προϊόν (large 1-itemsets)

2. Από κ= 2 και όσο το L_{k-1} δεν είναι κενό:

ί) βρες το σύνολο Ck των υποψηφίων μεγάλων λιστών από κ προϊόντα (candidate large k-itemsets) με βάση το L_{k-1}

ii) βρες ποία από αυτά εμφανίζονται περισσότερο από την ελάχιστη επιβεβαίωση και φτιάξε το σύνολο L_k= μεγάλες λίστες από κ προϊόντα

3.Για κάθε στοιχείο των L_1, L_2, ….L_n βρες ποία ικανοποιούν την ελάχιστη αξιοπιστία (minimum confidence).

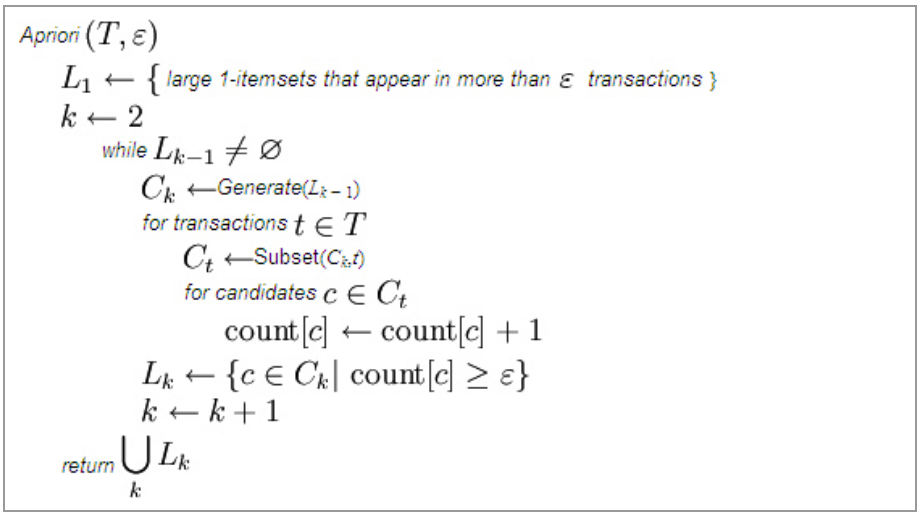
Πιο αναλυτικά:

Στο πρώτο βήμα ο αλγόριθμος μετρά τις εμφανίσεις του κάθε προϊόντος ξεχωριστά για να καθορισθούν οι *μεγάλες λίστες* μεγέθους ενός προϊόντος.

Το δεύτερο βήμα, είναι μία επαναλαμβανόμενη σειρά από υποβήματα. Κάθε επανάληψη, έστω η κ επανάληψη, περιλαμβάνει δύο υποβήματα. Αρχικά, μεγάλες *λίστες* από $_{K-1}$ προϊόντα Lk-1, που βρέθηκαν στην προηγούμενη (κ-1) επανάληψη χρησιμοποιούνται για να δημιουργηθούν οι *υποψήφιες μεγάλες λίστες* από κ προϊόντα (Ck), με βάση την αρχή ότι μία λίστα προϊόντων είναι *μεγάλη λίστα* προϊόντων μόνον εάν κάθε υποσύνολο της είναι *μεγάλη λίστα* προϊόντων.

Στη συνέχεια, σαρώνουμε το σύνολο εκπαίδευσης για να βρούμε την *επιβεβαίωση* των *υποψήφιων μεγάλων λιστών* από κ προϊόντα. Οι επαναλήψεις σταματούν όταν δεν ευρεθούν *υποψήφιες μεγάλες λίστες* προϊόντων. Τότε, στο τρίτο βήμα. από κάθε *μεγάλη λίστα* προϊόντων προκύπτουν κανόνες από τους οποίους γίνονται τελικά αποδεκτοί όσοι έχουν μεγαλύτερη από την *ελάχιστη αξιοπιστία.*

Σε μορφή ψευτοκώδικα (pseudo code) ο αλγόριθμος Apriori μπορεί να απεικονιστεί ως εξής:



1.5.2.3. Παραδειγμα Εφαρμογης Αλγοριθμου APRIORI

Έστω, για παράδειγμα, το Σύνολο Εκπαίδευσης που εμφανίζεται στον παρακάτω. Έστω επίσης, ότι ο χρήστης θέτει ελάχιστη *επιβεβαίωση* ίση με 50% και ελάχιστη αξιοπιστία ίση με 70%.

Σύνολο Εκπαίδευσης

κόκα-κόλα	γάλα	τυρί	φρυγανιές	κρέας
τυρί	βούτυρο	γάλα	νερό	
κρέας;	γάλα	ψωμί		
γάλα	τυρί	σοκολάτα		

Εάν ακολουθήσουμε τον αλγόριθμο apriori βήμα – βήμα έχουμε:

1	Από το πρώτο βήμα προκύπτει (σε παρένθεση η επιβεβαίωση αυτών):	**L1 = {γάλα (100%), τυρί (75%), κρέας (50%)}**
2	Στην πρώτη επανάληψη του δεύτερου βήματος, οι υποψήφιες μεγάλες λίστες από 2 προϊόντα είναι: Μετρώντας την επιβεβαίωση αυτών προκύπτει: Από την επιβεβαίωση αυτής της λίστας προϊόντων προκύπτει ότι L3 = { } και οι επαναλήψεις του δεύτερου βήματος σταματούν.	**L2 = {γάλα-τυρί (75%), γάλα-κρέας (50%), τυρί-κρέας (25%)}** **C2 = {γάλα-τυρί. γάλα-κρέας}** **L3 = { }**
3	Στο τρίτο βήμα, από όλους τους δυνατούς κανόνες που προκύπτουν από το L2:	**γάλα -> τυρί και γάλα -> κρέας**
Σ	Ο πρώτος έχει αξιοπιστία 75% ενώ ο δεύτερος 50%. Συνεπώς, εξάγεται μόνον ένας κανόνας συσχέτισης	**γάλα -> τυρί**

1.5.3. ΚΑΝΟΝΕΣ ΤΑΞΙΝΟΜΗΣΗΣ

Οι κανόνες ταξινόμησης, στη περιοχή της τεχνητής νοημοσύνης και ιδιαίτερα στον τομέα της εξόρυξης δεδομένων (data mining) και στη μηχανική μάθηση (machine learning), είναι η τεχνική ή το σύνολο των τεχνικών που αποσκοπούν να προβλέψουν ιδιότητες δεδομένων, σε επίπεδο ομάδων (groups) για τα δεδομένα (data instances). Παραδείγματος χάριν, με βάση ήδη υπάρχοντα ή νέα δεδομένα για την ιστορικότητα του καιρού σε μία περιοχή, η πρόγνωση του καιρού της επόμενης ημέρας θα μπορούσε να χαρακτηριστεί με αυτή την τεχνική σαν “ηλιοφανής”, ”βροχερή” ή ”συννεφιασμένη” και ούτω καθ’ εξής.

Μεταξύ των ποιο διαδεδομένων τεχνικών που υπάγονται στους κανόνες ταξινόμησης είναι τα νευρωνικά δίκτυα και τα δένδρα αποφάσεων. Άλλες λιγότερο διαδεδομένες τεχνικές είναι οι λίστες αποφάσεων, σε λογιστική παλινδρόμηση, οι ταξινομητές Bayes και οι ταξινομητές SVM (support vector machines) (Clark, P, and Niblett, T., 1989)

Στα πλαίσια της παρούσας Δ.Δ. και για λόγους οι οποίοι σχετίζονται με την συγκρισιμότητα της δενδρικής μορφής απεικόνισης των οντολογιών γίνεται παρακάτω μία σύντομη αναφορά στα δένδρα αποφάσεων.

Δένδρα Αποφάσεων (decisions trees)

Ένα δένδρο αποφάσεων ορίζεται μία δενδρική μορφή η οποία προκύπτει και κατασκευάζεται από ένα σύνολο δεδομένων (training set), οι οποίες με τη σειρά τους χαρακτηρίζονται από χαρακτηριστικά (attributes) και μία κλάση (label).

Η κατασκευή του δένδρου γίνεται με τη συσχέτιση των χαρακτηριστικών και των κλάσεων. Οι κόμβοι χαρακτηρίζονται είτε σαν ενδιάμεσοι (internal) είτε σαν τερματικοί (leaf) κόμβοι (nodes). Οι διακλαδώσεις των συνδέσεων των εσωτερικών κόμβων γίνονται με τον έλεγχο τιμών χαρακτηριστικών (attributes) έτσι ώστε κάθε τερματικός κόμβος να αποτελεί μία κλάση

Το δένδρο αποφάσεων χρησιμοποιούνται σύμφωνα με Safavian S.R. και Landgreba P. (Safavian, S.R. and Landgrebe, D., 1991) για να:

1. Ταξινομούν μεγάλο ποσοστό νέων δεδομένων
2. Να γενικεύουν από ένα υπαρκτό δείγμα εκπαίδευσης
3. Να ενημερώνουν το δείγμα εκπαίδευσης σε περίπτωση νέων δεδομένων
4. Να γίνεται ποιο εύκολα η δομή τους αντιληπτή

Τα κυριότερα στατιστικά πακέτα σήμερα υλοποιούν τα δένδρα αποφάσεων.

Μεταξύ άλλων χρησιμοποιήθηκαν στη μάθηση των μηχανών (machine learning), όπου έπρεπε να μειώνεται αποτελεσματικά το πλήθος της προς απόκτησης γνώση για αποφυγή υπερφόρτωσης (bottle neck) των έμπειρων συστημάτων (Feigenbaum, E.A., 1981).

1.5.4. ΚΑΝΟΝΕΣ ΟΜΑΔΟΠΟΙΗΣΗΣ – CLUSTERING RULES

Μία άλλη ευρέως διαδεδομένη μέθοδος για την ανεύρεση γνώσης είναι οι κανόνες ομαδοποίησης. Οι κανόνες ομαδοποίησης ανήκουν στην κατηγορία των μη εποπτευομένων (unsupervised) μεθόδων μάθησης από τα δεδομένα τους. Πιο συγκεκριμένα ενώ οι κανόνες συσχέτισης προαπαιτούν την αρχική διευθέτηση των δεδομένων από τα οποία θα ανακτηθεί η γνώση, οι κανόνες ομαδοποίησης δεν απαιτούν κάτι τέτοιο. Η μόνη συνδρομή του χρήστη πριν την έναρξη του χρήστη η οποία επηρεάζει και την ακρίβεια των ομάδων οι οποίες θα εξαχθούν, είναι ο αριθμός οποίος δείχνει τα

επιθυμητά clusters (ομάδες – κλάσεις) οι οποίες θα προκύψουν σαν αποτέλεσμα από τη διαδικασία. Οι ομάδες – κλάσεις, στο τέλος της διαδικασίας περιγράφονται από αντίστοιχο αριθμό κανόνων.

Η βασική λογική των κανόνων ομαδοποίησης είναι να δημιουργεί συνεχώς ομάδες δεδομένων των οποίων τα μέλη να έχουν περισσότερες ομοιότητες παρά διαφορές. Κατ' αυτόν τον τρόπο δημιουργούνται νοητές ομάδες δεδομένων οι οποίες έχουν δεδομένες ή περίπου δεδομένες αποστάσεις από κάποιο κεντρικό σημείο της ομάδος το οποίο ονομάζεται κέντρο ή centroid του cluster.

Ο κυριότερος αλγόριθμος στα πλαίσια των κανόνων ομαδοποίησης, είναι ο αλγόριθμος k-means (MacQueen, J.B., 1967).

Σύμφωνα με αυτόν τον αλγόριθμο εκτελούνται τα ακόλουθα βήματα:

1. Διαχωρισμός σε k ομάδες (k εισάγεται από το χρήστη)
2. Ορισμός προτύπου (mean) ανά ομάδα
3. Συνεχής μετακίνηση των εγγραφών στις k ομάδες, μέχρις ότου για κάθε εγγραφή να βρεθεί μία τέτοια ομάδα, από της οποίας το κέντρο (mean) να απέχει το λιγότερο η δεδομένη εγγραφή, σε σχέση με α κέντρα των υπολοίπων k-1 means ή ικανοποίηση συνθήκης ελαχίστου σφάλματος. Νέος υπολογισμός προτύπων ομάδων.

Γίνεται άμεσα κατανοητό ότι η χρήση της μεθόδου των κανόνων ομαδοποίησης είναι πολύ μεγάλης αξίας, διότι μπορεί να ομαδοποιεί σχεδόν αυτόματα πελάτες, προτιμήσεις ενδιαφέροντα, αποτελέσματα ποιοτικών χαρακτηριστικών όπως δημοσκοπήσεων, κατηγοριοποίηση προσωπικοτήτων κλπ.

Μεταξύ των μειονεκτημάτων των κανόνων ομαδοποίησης, είναι και το γεγονός ότι το αποτέλεσμα της διαδικασίας μπορεί να ερμηνευτεί με διαφορετικούς τρόπους. Δηλαδή, είναι συχνό το φαινόμενο να πρέπει να αποφασιστεί εάν γειτνιάζοντα clusters αποτελούν ουσιαστικά ένα ενιαίο ή τελείως διαφορετικά μεταξύ τους.

Το παραπάνω μειονέκτημα, μαζί με το γεγονός ότι βασικό χαρακτηριστικό της μεθόδου, που είναι η προϋπόθεση της μη οργάνωσης των δεδομένων, δεν επιτρέπει την χρήση της στην προτεινόμενη μεθοδολογία ONARM, δεδομένου ότι η ONARM εφαρμόζεται σε στοιχεία εκπαίδευσης που αναπαρίσταται με τη χρήση οντολογιών – γράφων.

1.5.5. Έρευνες Σε Γραφους Δενδρικης Μορφης

1.5.5.1. Εισαγωγη στις Δενδρικες Δομες

Ένα από τα πιο σημαντικά προβλήματα στον τομέα της έρευνας για την επίλυση προβλημάτων με αυτοματοποιημένο τρόπο είναι και αυτός ο οποίος αφορά προβλήματα τα οποία δύναται να απεικονισθούν σαν διαγράμματα γράφων με την μορφή δένδρων (tree graphs ή απλούστερα trees). Το χαρακτηριστικό σε αυτά τα προβλήματα είναι ότι η έρευνα η οποία καλούμαστε να υλοποιήσουμε ουσιαστικά υλοποιείται σε ένα σύνολο πιθανών λύσεων . Επίσης χαρακτηριστικό είναι ότι το εκάστοτε σύστημα, προκριμένου να βρει μία ή περισσότερες λύσεις, προσπαθεί συνεχώς να μετασχηματίζει μία τρέχουσα κατάσταση (state) σε μία πιθανή επόμενη, προκειμένου κατ' αυτόν τον τρόπο η πιθανή επόμενη να συμπέσει με την τελική, η οποία και θα αναπαριστά την λύση του προβλήματος. Είναι φανερό, ότι κατά την επίλυση ενός τέτοιου προβλήματος έχει πολύ μεγάλη σημασία τόσο η «ακρίβεια» της τελικής λύσεως όσο και το «κόστος» εξεύρεσης αυτής. Πολλές φορές βρισκόμαστε ενώπιον του διλήμματος εάν μία «τοπική τελική λύση» είναι αρκετή για την επίλυση του προβλήματος και άρα μπορεί να θεωρηθεί σαν «τελική» ή πρέπει το σύστημα να την αγνοήσει σαν μη optimal και να συνεχίσει την έρευνά του για την εξεύρεση της απολύτου optimal λύσεως και οπότε να θεωρείται αυτή ως τελική.

Παραδείγματος χάριν, εάν έχει ευρεθεί από το σύστημα αναζήτησης ένα σύνολο λύσεων το οποίο να είναι optimal για το 95% των δεδομένων – περιπτώσεων «αξίζει» να συνεχίσει το σύστημα την έρευνα και στο υπόλοιπο 5% των περιπτώσεων, με την ανάλογη χρήση των διατεθειμένων πόρων (σύστημα – χρόνος – προσπάθεια);

Για το σκοπό αυτό έχει εισαχθεί στον παραπάνω τομέα έρευνας η έννοια των heuristics (F. Chabris, 1987) . Ένα heuristic είναι μία μέθοδος που ακολουθείται προς την επίλυση ενός προβλήματος, η οποία τις περισσότερες φορές είναι μη επίσημη (formal), αλλά όμως πολύ γρήγορα οδηγεί σε μία λύση ενός προβλήματος η οποία είναι αποδεκτά κοντά στην καλύτερη πιθανή λύση του συγκεκριμένου προβλήματος. Παραδείγματα heuristics είναι οι αυθαίρετες παραδοχές (rules of thumb), μία προσεκτική υπόθεση (educated guess),

ενστικτώδεις κρίσεις (intuitive judgments) ή παραδοχές βασισμένες στην κοινή λογική (common sense)

Από τα ανωτέρω συνάγεται ότι προκειμένου να φτάσουμε στην τελική λύση (Goal) του προβλήματος περνάμε από ενδιάμεσα στάδια (states), ξεκινώντας από την αρχική κατάσταση (Start). Κατ' αυτόν τον τρόπο έχουμε:

f(N): S -> G,
όπου N ανήκει στο σύνολο $\{n_1, n_2, \ldots, n_k\}$ *που περιγράφουν το πρόβλημα (state space).*

Δενδρικές Δομές - Ορισμοί

Στη διαχείριση της Πληροφορίας η έννοια "tree" εννοεί την δομή εκείνη των πληροφοριών – δεδομένων τα οποία προσπαθούν να εξομοιώσουν την δενδρική δομή στη φύση, μέσω κόμβων (nodes) και κλάδων (edges). Είναι ευρέως αποδεκτό ότι για να αποτελέσει μία δομή «δενδρική δομή», πρέπει να ικανοποιεί ορισμένους περιορισμούς:

1. Να είναι μη κυκλική (acyclic)
2. Να συνδέονται κατά κάποιο τρόπο όλοι οι κόμβοι.
3. Όλοι οι κλάδοι (edges) να είναι ίσης αξίας (ισοβαρείς)

Υπάρχουν τέλος τα directed καθώς και τα undirected trees / graphs. Στα directed trees υπάρχει η έννοια της κατεύθυνσης στο κλάδο που συνδέει δύο γειτνιάζοντες κόμβους A, B, έτσι ώστε **A->B**, ενώ στο undirected το αντίθετο, δηλαδή ισχύει **A-B** , με οποιαδήποτε φορά επισκεψιμότητος.

Στις δενδρικές δομές, όπως προαναφέρθηκε, συναντάμε τις έννοιες των κόμβων. Πιο συγκεκριμένα, ένα κόμβος σε μία δενδρική δομή, δύναται να είναι είτε ένας κοινός κόμβος είτε ένας κόμβος – φύλλο (τερματικός) (leaf ή terminal). Σαν leaf ορίζεται αυτός ο κόμβος, ο οποίος δεν έχει περαιτέρω children (διαδόχους κόμβους). Όλοι οι υπόλοιποι κόμβοι ονομάζονται εσωτερικοί κόμβοι (interior nodes)

Δενδρικές Δομές – Τρόποι Αναπαράστασης

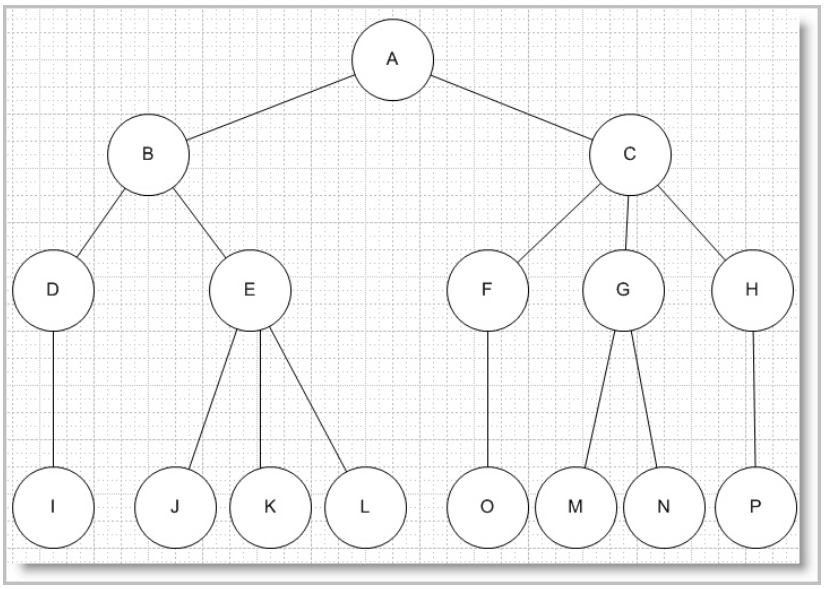
Ένας γράφος με δενδρική δομή μπορεί να αναπαρασταθεί με τέσσερις τρόπους (F. Chabris, 1987), οι οποίοι είναι ισοδύναμοι μεταξύ τους και ελεύθερα ανταλλάξιμοι χωρίς

περιορισμούς, δεν χάνουν δηλαδή πληροφορία, αλλά η κάθε μία γραφική απεικόνιση, δίνει βαρύτητα σε διαφορετικά χαρακτηριστικά του γράφου, τα οποία κατά περίπτωση θα μας απασχολήσουν. Τέτοια χαρακτηριστικά είναι:

- Μονοπάτια (σύνολο κλάδων στη σειρά {ordered})
- Κόμβοι
- Επίπεδα δενδρικής δομής (Υπολογισμός μέγιστου βάθους κλπ)
- Αποστάσεις μεταξύ κόμβων, που δεικνύουν ομοιότητες μεταξύ εννοιών
- Έννοιες, περιεχόμενες έννοιες κλπ
- Ορισμός τάξεων εννοιών (classes, subclasses)
- Ιδιότητες τάξεων (class properties)

Κατ΄ αυτόν τον τρόπο, μπορούμε να για ένα υποθετικό γράφο – δένδρο να ορίσουμε τις τέσσερις μορφές του, όπως εμφανίζονται παρακάτω διαγράμματα:

(α) Σαν κλασσική δενδρική μορφή



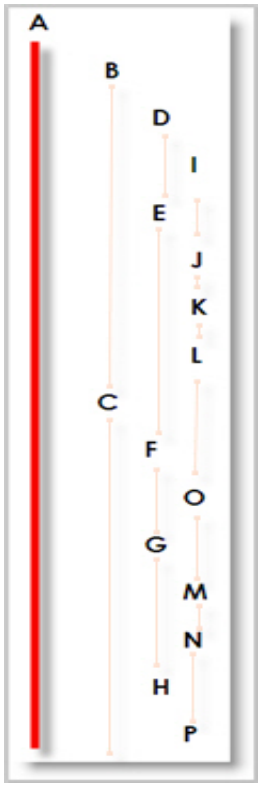
(β) Σαν Λίστα Κόμβων

Η ανωτέρω δενδρική δομή δύναται επίσης και να αναπαρασταθεί ως λίστα όπως φαίνεται πιο κάτω:

(A (B (D (I), E (J, K, L)), C (F (O), G (M, N), H (P))))

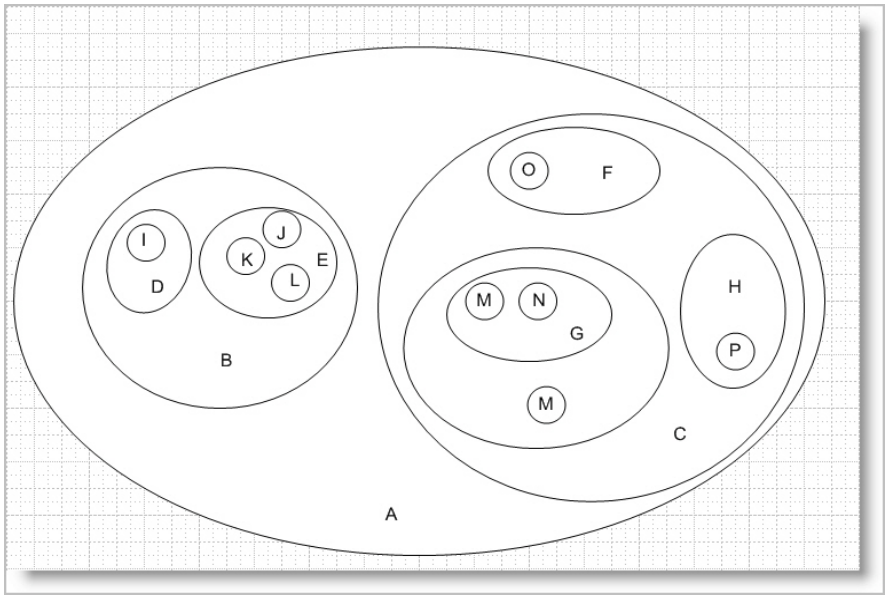
Η παρουσίαση της δενδρικής δομής είναι πολύ εύχρηστη στην περίπτωση υλοποιήσεων δενδρικών δομών προγραμματιστικά, ορισμένες μάλιστα γλώσσες σαν την LISP (LISt Processing), εγγενώς υποστηρίζουν τις ανωτέρω δομές, τύπου λίστας και όλες τις διαδικασίες (push, pop, κλπ) επί αυτών. Σε αυτή την απεικόνιση δίνεται μεγαλύτερη βαρύτητα στους κόμβους παρά στους κλάδους, όπως γίνεται εύκολα αντιληπτό.

(γ) Σαν ανεστραμμένη ιεραρχική δομή



Στο διπλανό σχήμα βλέπουμε το προαναφερθέν γράφημα, αλλά σε αυτή τη περίπτωση έχει αναπαρασταθεί σαν μία ανεστραμμένη δενδρική δομή, η οποία παραμένει ισοδύναμη με την αρχική. Στο διπλανό σχεδιάγραμμα, οι γραμμές δεν ενώνουν κόμβους του γράφου, δηλαδή δεν αποτελούν κλάδους του δένδρου, εν αντιθέσει χρησιμοποιούνται για την οπτικοποίηση του βάθους (level) του κάθε κόμβου της δενδρικής δομής. Έτσι βλέπουμε ότι ο κόμβος A βρίσκεται στο ανώτατο επίπεδο 0, δηλαδή αποτελεί τη ρίζα του δένδρου, οι κόμβοι B και C βρίσκονται στο $2^{ο}$ επίπεδο, οι D, E, F, G, H στο $3^{ο}$, οι I,J,K,L,O,M,N,P στο $4^{ο}$ οι οποίοι και αποτελούν leaf nodes και δίνουν κατ' αυτόν τον τρόπο το max level (μέγιστο βάθος δένδρου) ίσου με 4. Αυτού του είδους η αναπαράσταση χρησιμοποιείται στα decision trees (J. R. Quinlan, 1987).

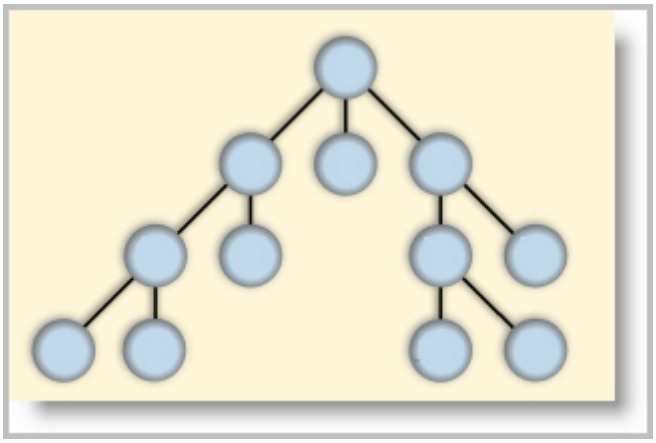
(δ) Υπό την μορφή συνόλων, που περικλείουν το ένα το άλλο:



Οι κυριότεροι αλγόριθμοι για την εξεύρεση λύσεων σε έναν γράφο – δένδρο είναι οι παρακάτω:

- A*
- B*
- Bellman-Ford Algorithm
- Best-first Search
- Bidirectional Search
- Breadth-First Search
- Depth-First Search
- Depth-Limited Search
- Dijkstra's Algorithm
- Floyd-Warshall Algorithm
- Hill Climbing Algorithm
- Iterative deepening depth-first search
- Johnson's Algorithm
- Uniform-Cost Search

Από τους ανωτέρω αλγορίθμους, λόγω της μη ύπαρξης βαρών στα κλαδιά τους (ισότιμα), θα εξετάσουμε τους δύο αλγορίθμους Breadth First Search (BFS) καθώς και Depth First Search (DFS). Ας υποθέσουμε ότι το κατωτέρω δένδρο, αναπαριστά όλο το state space, μέσα στο οποίο είναι δυνατό να εξευρεθεί η λύση του προβλήματος. Σκοπός είναι να εξερευνηθεί προς τούτο η κατωτέρω δενδρική δομή.



Αυτό θα γίνει μέσω των αλγορίθμων BFS και DFS, οι οποίοι και παρουσιάζονται αναλυτικότερα κατωτέρω.

1.5.5.2. BREADTH FIRST SEARCH (BFS)

Ή έρευνα γράφου κατά BFS (Knuth, Donald E. , 1997), υλοποιείται χρησιμοποιώντας μία δομή τύπου QUEUE, στην οποία και αποθηκεύονται οι κορυφές του γράφου που έχουν επισκεφθεί.

Η κύρια ιδέα είναι ότι ξεκινώντας σε ένα γράφο – δένδρο από την ρίζα, τότε, προχωρούμε και εξετάζουμε όλους τους παραπλήσιους κόμβους γειτνίασης. Κατόπιν για καθένα από αυτούς τους κόμβους, επισκεπτόμεθα καθέναν από τους γειτνιάζοντες αυτούς οι οποίοι δεν έχουν επισκεφθεί και συνεχίζουμε κατ' αυτόν τον τρόπο αναζητώντας την συνθήκη επίλυσης του προβλήματος (στόχος – goal).

Ο Αλγόριθμος BFS παρατίθεται κατωτέρω κωδικοποιημένα:

*1. Put the root node on the **queue**.*

*2. Pull a node from the beginning of the **queue** and examine it.*

2.1 If the searched element is found in this node, quit the search and return a result.

*2.2 Otherwise, push all the (so-far-unexamined) successors (the direct child nodes) of this node into the end of the **queue**, if there are any.*

*3. If the **queue** is empty, every node on the graph has been examined -- quit the search and return "not found".*

4. Repeat from Step 2.

1.5.5.3. DEPTH FIRST SEARCH (DFS)

Σύμφωνα με τον Knuth (Knuth, Donald E. , 1997), η γενική ιδέα είναι η ίδια μόνον που αντί για την δομή queue χρησιμοποιείται η stack με αναδρομή.

*1. Put the root node on the **stack**.*

*2. Pull a node from the beginning of the **stack** and examine it.*

2.1 If the searched element is found in this node, quit the search and return a result.

*2.2 Otherwise, push all the (so-far-unexamined) successors (the direct child nodes) of this node into the end of the **stack**, if there are any.*

*3. If the **stack** is empty, every node on the graph has been examined -- quit the search and return "not found".*

4. Repeat from Step 2.

1.5.5.4. ΑΠΟΤΕΛΕΣΜΑΤΑ BFS - DFS

Εάν στο προαναφερθέν δένδρο αριθμήσουμε τους κόμβους από το 1 έως το 12, τότε προκύπτουν οι δύο αριθμήσεις όπως φαίνονται κατωτέρω στα δύο διαγράμματα:

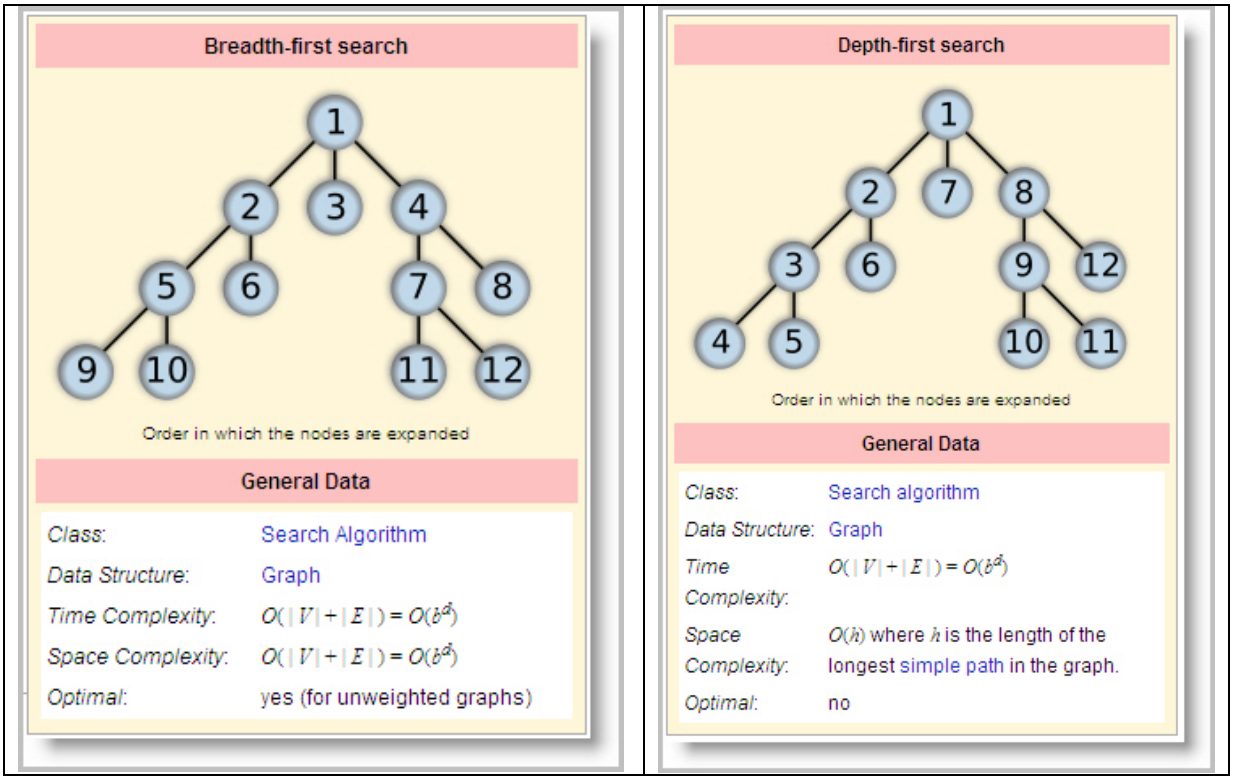
Παρατηρούμε ότι στην περίπτωση της BFS οι κόμβοι αριθμούνται κατά «πλάτος» ή «οριζόντια», ενώ στην περίπτωση αρίθμησης με την DFS, οι κόμβοι αριθμούνται κατά «βάθος» ή «κάθετα».

Στα πλαίσια της παρούσας Δ.Δ. και σε συσχέτιση με τον αλγόριθμο APRIORI, θεωρείται υποχρεωτικό, όπως η αρίθμηση ενός κόμβου K_i, συμβολίζεται ως $A(K_i)$ και ισχύει πάντα: $A(K_i) > A(K_j)$, όπου $i < j$, δηλαδή ο κόμβος K_i είναι parent node του K_j

Κατ΄ αυτόν τον τρόπο, ο αριθμός που προκύπτει σαν αρίθμηση για ένα child node, είναι πάντα μεγαλύτερος από τον αριθμό που αναπαριστά την αρίθμηση του προγόνου του κόμβου.

Είναι φανερό, ότι μία τέτοια διάταξη μπορεί να προκύψει μόνο από την BFS αρίθμηση, όπως φαίνεται στο παρακάτω σχήμα.

Ο ακριβής τρόπος με τον οποίο προκύπτει η επιλογή του εκάστοτε αριθμού για την αρίθμηση ενός κόμβου, δεν έχει σημασία στα πλαίσια της παρούσης. Για λόγους απλούστευσης υπολογισμών, χωρίς να χάνεται η γενικότητα, η αρίθμηση γίνεται με την απλή αύξηση κατά 1 από κόμβο σε κόμβο που επισκεπτόμαστε.



1.5.5.5. Σύγκριση BFS & DFS Αλγορίθμων

Στο ανωτέρω σχήμα φαίνεται η γραφική σύγκριση των δύο αυτών αλγορίθμων για την έρευνα δενδρικών δομών, στις οποίες τα paths είναι no-weighted, δηλαδή, ισοβαρή - ισότιμα.

Σύμφωνα με τον Knuth (Knuth, Donald E., 1997), η DFS έρευνα φαίνεται πιο «εύκολα» (cleaner) από την έρευνα κατά BFS. Αυτό κατά τον Knuth γίνεται διότι υλοποιείται με πολύ φανερό τρόπο, χωρίς να κάνει χρήση εξωτερικών δεδομένων, παρά μόνο κάνοντας αναδρομική κλήση recursion στον εαυτό της.

Το στοιχείο όμως αυτό πού κάνει πιο εύκολα κατανοητή την DFS είναι ταυτόχρονα και το μεγάλο σημείο αδυναμίας της, δηλαδή η αναδρομική κλήση της. Η υλοποίηση της DFS κατ' αυτόν τον τρόπο, όταν εφαρμόζεται σε πολύ μεγάλο όγκο δεδομένων, δηλαδή σε δένδρα

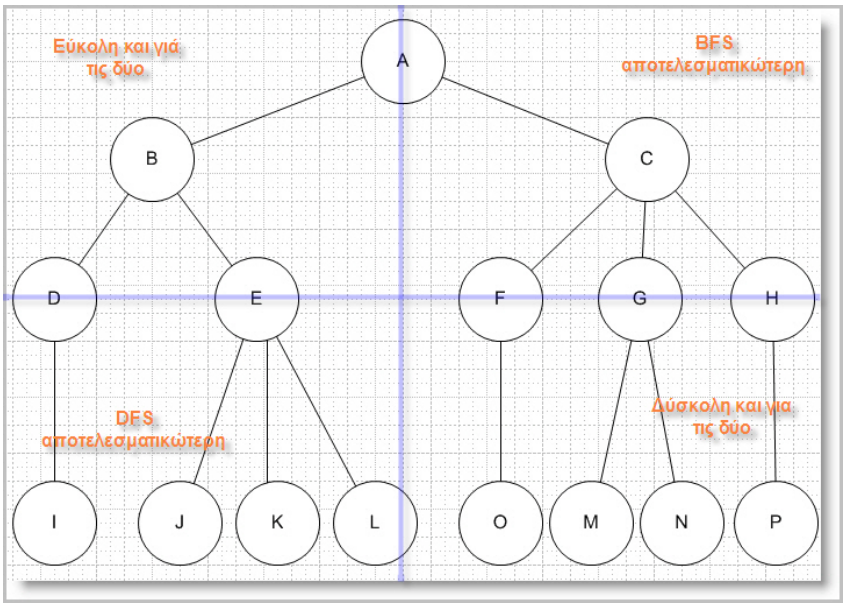
όπου υπάρχουν πολύ μεγάλα απλά (χωρίς πολλαπλές διακλαδώσεις) μονοπάτια, δεν εγγυάται την εξεύρεση ούτε καν μίας optimal λύσεως, πολλώ δε μάλλον της ευκταίας τελικής λύσεως. Αυτό συμβαίνει διότι κατά την διαδικασία backtracking, αναπτύσσει και αφήνει ημιτελώς ανοικτά πολύ μεγάλα τμήματα του προς εξέταση γράφου, με αποτέλεσμα την εξάντληση των πόρων του συστήματος, ειδικότερα της μνήμης. Έτσι, σε περίπτωση αναζήτησης αλγορίθμου έρευνας, ο οποίος πρόκειται να υλοποιηθεί σε γράφους – οντολογίες, οι οποίες πρόκειται να αναπαρασταθούν σε εν γένει φορητές συσκευές, τότε προτείνεται ο αλγόριθμος BFS, ο οποίος στα ίδια επίπεδα πολυπλοκότητας χρόνου, όπως φαίνεται στο κατωτέρω σχήμα, επιστρέφει optimal λύση.

Ένα παράδειγμα της καταλληλότητος του BFS είναι η υλοποίηση ενός WEB robot για την αυτοματοποιημένη αναζήτηση πληροφοριών από το WEB. Εν αντιθέσει με τον BFS, ο DFS, εάν εχρησιμοποιείτο, πολύ γρήγορα θα «εγκατέλειπε» την προσπάθεια, διότι σε κάθε site που θα επεσκέπτετο θα έπρεπε να το εξετάσει εξονυχιστικά, μέχρι της τελευταίας του σελίδας (leaf node), χωρίς τοπικό, αρκετά ικανοποιητικό αποτέλεσμα, προκειμένου να επισκεφθεί ένα άλλο WEB site. Επίσης, το σύστημα – WEB server που θα εδέχετο τόσες κλήσεις – hits ταυτόχρονα, από ένα τέτοιο WEB robot, πολύ γρήγορα θα κατέρρεε. Η BFS, σε αυτή την περίπτωση θα επιμέριζε τον αριθμό κλήσεων – hits σε ποιο πολλά sites κάθε φορά, με αποτέλεσμα την επιτυχή έκβασή της.

Στο παρακάτω σχεδιάγραμμα, φαίνεται η σύγκριση των δύο, σε σχέση με την τοποθέτηση χωροθετικά της λύσης (goal) στό σύνολο των πιθανών λύσεων (state space) που διερευνώνται από τις BFS και DFS.

Παρατηρούμε, ότι όταν οστόχος της έρευνας βρίσκεται στο πάνω αριστερά τεταρτημόριο, τότε η διαδικασία αναζήτησης είναι όμοια αποδοτική και για την BDF και για την DFS έρευνα. Στο τεταρτημόριο που βρίσκεται επάνω δεξιά η εξεύρευση του στόχου είναι αποτελεσματικώτερη στην BFS έρευνα. Στο κάτω αριστερό τεταρτημόριο αναδεικνύεται η DFS αποτελεσματικώτερη, ενώ ο χώρος του κάτω δεξιά τετρτημορίου παραμένει όμοια δύσκολος για την εξεύρευση του στόχου.

Τα ανωτέρω ισχύουν για ισορροπημένες (balanced) δομές γράφων - οντολογιών. Στα πλαίσια της παρούσης Δ.Δ., προτάθηκαν νέα μέτρα ενδιαφέροντος, εξερεύνησης ενδιαφερόντων κόμβων και μελετήθηκε η συμπεριφορά τους σε μή ισορροπημένες (unbalanced) δενδρικές δομές - οντολογίες.



Εικόνα – Χωροταξική απεικόνιση αποτελεσματικότητας BFS & DFS ερευνών

1.5.5.6. Γραφηματα Δενδρικων Δομων και Κανονες Συσχετισης

Πολλοί αλγόριθμοι εξόρυξης γνώσης με συσχέτιση (induction), δημιουργούν δενδρικές δομές όπως αναφέρθηκε παραπάνω. Ο Quinlan (J. R. Quinlan, 1987), έχει προτείνει ένα τρόπο με τον οποίο ένα δένδρο λήψης αποφάσεων, μπορεί να μετασχηματιστεί σε ένα αντιπροσωπευτικό σύνολο κανόνων παραγωγής (production rules).

Η μεθοδολογία υλοποιείται σε δύο διακριτά βήματα. Πρώτον, μέσω του συνόλου (set) εγγραφών των συναλλαγών (transactions), που είναι αυτές, βάσει των οποίων δημιουργείται η δενδρική δομή, επιχειρώντας να γενικεύσει και να αξιολογήσει την αξιοπιστία επί του συνόλου τους. Δεύτερον, επαναπροσδιορίζοντας τους ήδη δημιουργηθέντες κανόνες από το πρώτο βήμα, με το να τους ελαττώσει σε πλήθος, χωρίς όμως σημαντική μείωση της αξιοπιστίας. Κατ' αυτόν τον τρόπο, μέσω της ανωτέρω διαδικασίας (refinement), το τελικό σύνολο κανόνων που προκύπτει είναι σημαντικά μικρότερο από το δένδρο το οποίο περιγράφει και προήλθε, καθώς επίσης έχει μεγαλύτερη ακρίβεια στην περιγραφή της γενικής λύσης, αναδεικνύοντας περιοχές συγκέντρωσης γνώσης οι οποίες προηγουμένως είχαν χαρακτηριστεί ως μη κατηγοριοποιήσιμες (unclassified), λόγω της επίδρασης των παραγόντων με πολύ χαμηλή πιθανότητα εμφάνισης και συσχέτισης. Ουσιαστικά ο Quinlan προτείνει heuristics με σκοπό την αποτελεσματικότερη έκβαση της έρευνας στο πεδίο τιμών του προβλήματος, με στόχο την

εξεύρεση εκείνης της τιμής που ικανοποιεί το δυνατόν εγγύτερα, αν όχι επακριβώς, την εύρεση του τελικού σκοπού - στόχου (goal).

Τέλος, μέσω της μεθοδολογίας εξαγωγής κανόνων συσχέτισης, παρέχεται η δυνατότητα συνδυασμού διαφορετικών δενδρικών δομών για το ίδιο προς κατηγοριοποίηση γνωστικό πεδίο (classification domain).

Σε γενικές γραμμές, κατά Quinlan, ένα δένδρο αποφάσεων μπορεί να εκφραστεί σαν ένα σύνολο κανόνων της μορφής:

If LEFT- HAND-SIDE then CLASS (certainty factor)

Μία τέτοια έκφραση του δένδρου, επιτρέπει την πιο ολοκληρωμένη αναπαράσταση μίας πολύπλοκης δενδρικής δομής από τον knowledge expert καθώς επίσης και την πιο εύκολη επεξεργασία από τα υπολογιστικά συστήματα, δεδομένου ότι η απεικόνιση αυτού του τύπου έχει υιοθετηθεί σαν η βάση για τα έμπειρα συστήματα (expert systems) και όχι μόνον.

Κατά τη διαδικασία τής πρώτης φάσης, δηλαδή εκείνης της δημιουργίας κανόνων από το δένδρο, αυτοί δημιουργούνται ακολουθώντας τα paths (κλάδους) μέχρι τα leaves (τελικούς κόμβους). Για κάθε τέτοιο κανόνα, για τον οποίο ξεκινάμε από τη ρίζα του δένδρου έως κάποιο τελικό φύλλο (leaf) ορίζονται συνθήκες τις οποίες πρέπει να ικανοποιούν όλοι οι ενδιάμεσοι κόμβοι (interior) έως αυτό το τελικό φύλλο.

Έτσι έχουμε:

If $X_1 \bigwedge X_2 \ldots\ldots \bigwedge X_n$ then class C , όπου class C είναι η κλάση του φύλλου.

Στην περίπτωση πού έχουμε σύνδεση εννοιών οι οποίες είναι ξένες μεταξύ τους (disjunctive concepts) (Bundy A., Silver B. and Plummer D., 1985), τότε, επειδή η ικανοποίηση μόνων ξεχωριστά των εννοιών που όμως συσχετίζονται μεταξύ τους, χωρίς να είναι απαραίτητη η ταυτόχρονη αλήθειά τους, δημιουργεί πρόβλημα στην συσχέτιση αυτή, διότι η πληροφορία αυτή δεν είναι δυνατό να απεικονισθεί στο σύστημα. Έτσι, η απάλειψη τέτοιων περιπτώσεων ουσιαστικά προωθεί το μοντέλο αναπαράστασης της γνώσης σε ένα πιο γενικευμένα αληθές μοντέλο, αποφεύγοντας να εστιάζει σε επί μέρους, αληθείς μεν κατά μόνας, αλλά μη σχετιζόμενες πληροφορίες εν τω συνόλω τους.

Το επόμενο βήμα, εφ' όσον ολοκληρώθηκε το βήμα παραγωγής του συνόλου των κανόνων, είναι να εξευρεθούν εκείνοι οι κανόνες οι οποίοι είναι οι πιο αντιπροσωπευτικοί του συνόλου των κανόνων, ή των υποσυνόλων του.

Αυτό επιτυγχάνεται ως εξής:

Για να κατηγοριοποιηθεί μία περίπτωση, βρίσκεται ένας κανόνας που να εφαρμόζεται σε αυτή. Εάν υπάρχουν παραπάνω του ενός, επιλέγεται αυτός με τον υψηλότερο παράγοντα ασφάλειας (certainty factor). Ο παράγοντας ασφάλειας εισάγεται και θεωρείται δεδομένος από την αρχή του προβλήματος (in a heuristic mode). Εάν δεν εφαρμόζεται κανένας κανόνας, τότε η κλάση αυτή είναι η πιο συχνά εμφανιζόμενη στο σύνολο εκπαίδευσης (training set).

Στα πλαίσια της παρούσας Δ.Δ., μία οντολογία θεωρείται και είναι ουσιαστικά ένα δένδρο αποφάσεων. Όταν πλοηγούμεθα προς κάποιο τελικό κόμβο (leaf node) αυτής, ουσιαστικά επισκεπτόμεθα ένα σύνολο ενδιάμεσων κόμβων οι οποίοι ικανοποιούν την συνθήκη της εξειδίκευσης και καταλήγουμε σε αυτόν τον τελικό κόμβο. Στην περίπτωση που είμαστε σε θέση να επισκεφθούμε τον ίδιο τελικό κόμβο (leaf node), μέσω διαφορετικών μονοπατιών (paths), τότε πραγματοποιούμε εξεύρευση συσχετίσεων μεταξύ αυτών των μονοπατιών και του τελικού κόμβου (goal), με σκοπό την ανακάλυψη των δημοφιλέστερων μονοπατιών και υπομονοπατιών (paths – subpaths) ως πρός αυτόν τον τελικό κόμβο.

Μία άλλη μεθοδολογία τύπου heuristic, η οποία επίσης εξάγει κανόνες συσχέτισης από δενδρικές δομές είναι αυτή που προτείνεται από τον Guangfei Y. et. al. (Guangfei Yang, Kaoru Shimada, Shingo Mabu, Kotaro Hirasawa and Jinglu Hu, 2007), με χρήση Sequential Connected Chains (SCC), και την χρήση μεθοδολογιών Γενετικού Προγραμματισμού σε γενικότερες δομές γράφων – δικτύων. Σύμφωνα με αυτή την μεθοδολογία, σε κάθε γράφο, εκτός από την ρίζα – αρχικό κόμβο (SP {starting point}) υπάρχει ο τελικός κόμβος (EP {end point}) και οι ενδιάμεσοι, οι οποίοι ονομάζονται είτε σαν judgment κόμβοι είτε σαν processing nodes. Σε αυτή την μοντελοποίηση οι judgment nodes λαμβάνουν το ερέθισμα (input) από το περιβάλλον και αποφασίζεται ακολούθως ποιος processing κόμβος θα εκτελεσθεί κατόπιν σε σειριακή μορφή.

Ειδικότερα, όσον αφορά την χρήση του SCC σε γράφους οντολογιών, αυτός ορίζεται σαν το σύνολο των κλάσεων σε μία οντολογία που αποτελούνται από μία αλυσίδα (chain) από

judgment nodes, έτσι ώστε κάθε ένας από αυτούς να έχει ακριβώς ένα προηγούμενο judgment node και ένα επόμενο. Επίσης, κάθε judgment node έχει ακριβώς ένα αντίστοιχο processing node.

Κατ' αυτόν τον τρόπο δημιουργούνται κανόνες οι οποίες συσχετίζουν τις κλάσεις ανάμεσά τους. Στην διαδρομή αυτή των κανόνων που δημιουργούνται εξετάζεται η εφαρμοσιμότητα (applicability) των ήδη παραχθέντων κανόνων στον επόμενο κόμβο κρίσης (judgment) και ο αλγόριθμος συνεχίζει κατ' αυτόν τον τρόπο.

Τέλος, το SCC heuristic, μετά την εξεύρεση των πιθανών κανόνων συσχετίσεις εισάγει το δεύτερο heuristic το οποίο είναι το Maximum Rule Length (MRL). Αυτό είναι ένα μέτρο το οποίο εισάγεται έτσι ώστε να περιορισθεί το μήκος των κανόνων συσχέτισης βασιζόμενο στην παραδοχή ότι ακόμα και μικρότερου μήκος κανόνες παραγωγής εμπεριέχουν την γενικευμένη γνώση και πληροφορία και ταυτόχρονα οι πολύ μεγάλοι σε μέγεθος κανόνες εισάγουν πληροφορία η οποία ήδη υπάρχει σε μικρότερους κανόνες.

Είναι ενδιαφέρον - και αυτό θα μελετηθεί – η εξεύρεση και άλλων τέτοιων heuristics για την μείωση του κόστους (σε resources) αυτής της έρευνας της συσχέτισης, όπως αυτών που περιγράφηκαν ανωτέρω, για την αναζήτηση συσχετιζόμενων paths σε οντολογίες.

Κεφάλαιο 2. Θεωρητικό Υπόβαθρο Προτεινόμενης Μεθοδολογίας

Σύγκριση και Αντιστοίχιση Οντολογιών

Αυτό πού δίνει πρόσθετη αξία στην γνώση η οποία είναι καταγεγραμμένη και αποθηκευμένη στον παγκόσμιο ιστό είναι η χρήση σημασιολογικών τεχνικών και μεταπληροφορίας έτσι ώστε να προκύπει και να εξελίσσεται συνεχώς σε αυτό που ονομάζεται semantic WEB. Επειδή η κατασκευή ιστοτόπων έχει γίνει μία σχετικά εύκολη διαδικασία και έχει τυποποιηθεί σε μεγάλο βαθμό, υπάρχει πληθώρα περιεχομένου, για κάθε είδους εφαρμογή καταγεγραμμένο στο διαδίκτυο. Έτσι, προκύπτει το θέμα της συσχέτισης των πληροφοριών αυτών, που σαν έννοιες αναπαριστούν το ίδιο περιεχόμενο αλλά διαφέρουν ως προς την περιγραφή τους. Αυτή η διαφοροποίηση προκύπτει από την ανάγκη για διαφορετική χρήση της ίδιας γνώσης καθώς επίσης και λόγω της διαφοροποιημένης αντίληψης για την ίδια γνώση. Αυτό είναι απόρροια που παρουσιάζεται στα πλαίσια διαφορετικών απαιτήσεων χρηστών, που είναι οργανωμένοι σε πολλαπλές γνωσιακές κοινότητες χρηστών (user communities), κάθε μία από τις οποίες αντιλαμβάνεται την πραγματικότητα και την περιγράφει υπό διαφορετικό πρίσμα.

Σε αυτά τα πλαίσια θεωρείται δεδομένη η ύπαρξη πολλαπλών οντολογιών οι οποίες περιγράφουν το ίδιο ή παραπλήσιο γνωσιακό πεδίο (knowledge domain) και υπάρχει ανάγκη συσχέτισής τους. Για το σκοπό έχει αρχίσει πρόσφατα έρευνα στο πεδίο ενοποιήσεως οντολογιών, όπου βασικό ρόλο διαδραματίζει αντιστοίχηση «όμοιων» εννοιών ή ολόκληρων κλάδων και υπό-οντολογιών, υπό την μορφή υπογράφων. Ακόμα δε περισσότερο, υπάρχει έρευνα στην ενοποίηση επί μέρους οντολογιών, σε πιο πολύπλοκες meta-οντολογίες, ανωτέρου επιπέδου που θα είναι σε θέση να καλύπτουν όλο και μεγαλύτερο εύρος μεταπληροφορίας για διαφορετικά γνωσιακά πεδία.

Ακόμα περισσότερο, η διαδικασία της σύγκρισης γίνεται πιο δύσκολη καθ' όσον δεν περιγράφονται όλες οι οντολογίες με τον ίδιο τρόπο απεικόνισης. Έτσι υπάρχουν οντολογίες οι οποίες περιγράφονται μέσω σχεσιακών βάσεων δεδομένων, άλλες μέσω καθαρά γλωσσολογικών δομών, ή μαθηματικοποιημένων γράφων και ούτω το καθεξής.

Έτσι, υπάρχει ένα πολύ μεγάλο εύρος στη συγκρισιμότητα τέτοιων δομών, που ποικίλουν από θεωρητικούς τελείως τρόπους, έως πιο πρακτικούς. Είναι σημαντικό να σημειωθεί σε αυτό το σημείο, ότι η πολυπλοκότητα και η διαφοροποίηση των αναγκών στον τρόπο απεικόνισης και σύγκρισης (mapping) δεν έχουν οδηγήσει ακόμα την έρευνα σε αυτό το πεδίο σε κάποια προτυποποίηση η οποία να έχει γίνει αποδεκτή. Η προσπάθεια με το πλησιέστερο αποτέλεσμα ήταν αυτή που εκφράστηκε σαν τελικό παραδοτέο από το ONTO WEB EU initiative το 2002, ενώ η τεχνολογία είναι διαθέσιμη από το 1994 περίπου στο τομέα του knowledge engineering.

Στα πλαίσια αυτά, σημαντικό ρόλο έχει η επίλυση επί μέρους προβλημάτων στη διαδικασία της αντιστοίχισης οντολογιών, τα οποία περιγράφονται αναλυτικά παρακάτω ως μέρος αυτής.

2.1. ΑΝΤΙΣΤΟΙΧΗΣΗ ΟΝΤΟΛΟΓΙΩΝ

Ορισμός

¨Όπως αναφέρθηκε σε προηγούμενη ενότητα, μία οντολογία είναι μία ιεραρχία εννοιών ή γενικότερα ένα σύνολο εννοιών που έχουν διαταχθεί (ordered) μερικά ή ολικά. Στην γενική περίπτωση μπορούμε να θεωρήσουμε ότι είναι partially – ordered (porder). Επίσης, μία οντολογία έχει ένα σύνολο από ιδιότητες και χαρακτηριστικά, οι οποίες παίρνουν τη μορφή μαθηματικών συναρτήσεων, με συγκεκριμένα πεδία ορισμού και τιμών, από το σύνολο των εννοιών και πάνω σε αυτές.

Γενικεύοντας τα ανωτέρω, ένας μετασχηματισμός (mapping) μιας οντολογίας μπορεί να αναπαρασταθεί σαν ένα ζεύγος O = (S, A) όπου το S αναπαριστά το λεξικό της οντολογίας και το A το λεξικό υπό την έννοια που αυτό θα εκλάβει μέσω αυτού του μετασχηματισμού. Τα ανωτέρω ονομάζονται οντολογικά αξιώματα (ontology axioms) (Kalfoglou, Y., Schorlemmer, M., 2002a).

Κατά τη διαδικασία mapping μεταξύ δύο οντολογιών, ουσιαστικά γίνεται προσπάθεια συσχέτισης του λεξικού της μίας οντολογίας με το λεξικό της άλλης, υπό την έννοια ότι οι δύο οντολογίες αναφέρονται στο ίδιο πεδίο εφαρμογής, έτσι ώστε αυτά τα οποία

αναφέρθηκαν σαν οντολογικά αξιώματα να παραμείνουν ισχυρά και μετά την συσχέτιση και τις διεργασίες λόγω αυτής.

2.1.1. ΜΟΡΦΙΣΜΟΙ

Μορφισμοί (Morphisms) είναι μετασχηματισμοί οι οποίοι συντηρούν τις μαθηματικές δομές που περιγράφουν τις οντολογίες και συσχετίζονται με την χαρτογράφηση και απεικόνισή τους τους σημαντικά.

Ένας μετασχηματισμός f μίας οντολογίας $O_1 = (S_1, A_1)$ σε μία οντολογία $O_2 = (S_2, A_2)$ ονομάζεται morphism, εάν στο μερικώς διατεταγμένα α, β για τα οποία ισχύει (α <= β => f(a) <= f(b))

Κατά τα ανωτέρω ισχύει ότι με βάση τους μορφισμούς σε οντολογίες ότι:

Υπάρχει μία επιμέρους απεικόνιση – χαρτογράφηση μίας οντολογίας O_1 σε μία οντολογία O_2, εάν υπάρχει υποοντολογία O_1' έτσι ώστε να υπάρχει πλήρης απεικόνιση – χαρτογράφηση από την O_1' στην O_2.

2.1.2. ΕΜΠΛΟΥΤΙΣΜΕΝΕΣ ΟΝΤΟΛΟΓΙΕΣ

Ένα από τα πιο σημαντικά ζητήματα που πρέπει να αντιμετωπίζονται και επηρεάζει και την μεθοδολογία που πρέπει να υιοθετηθεί στη διαδικασία mapping είναι αυτό των οντολογιών που έχουν εμπλουτιστεί και εμπλουτίζονται συνεχώς με πραγματώσεις (instances) νέων εννοιών και ονομάζονται εμπλουτισμένες οντολογίες (Populated Ontologies). Αυτό είναι ένα πολύ κεντρικό θέμα, δεδομένου ότι ο πληθυσμός αυτών (population of instances) αλλά και το είδος, χαρακτηρίζει και την ίδια την οντολογία. Αυτό ανάγεται σε μείζον θέμα όταν οι οντολογίες αυτές έχουν αναπτυχθεί από διαφορετικές ομάδες χρηστών (user communities), και έτσι πρέπει κατά την διαδικασία mapping να λαμβάνεται υπ' όψιν και η σημασιολογική παράμετρος της έννοιας, μέσα στα πλαίσια του γνωστικού πεδίου αλλά και η δομική – μορφολογική του ύπαρξη και συσχέτιση, μέσα σε αυτό, με άλλες έννοιες της οντολογίας.

2.1.3. Αντιστοιχηση, Ευθυγραμμιση, Επικοινωνια και Ενοποιηση Οντολογιων

Δεδομένων των ανωτέρω, έχουμε τα ακόλουθα διακριτά βήματα στις λειτουργίες συσχετίσεων, μεταξύ των οντολογιών:

- *Διαδικασία αντιστοίχησης (mapping):*
  Αποτελείται από ένα σύνολο συναρτήσεων τα οποία συσχετίζουν έννοιες του λεξιλογίου της μίας οντολογίας, με έννοιες του κοινού λεξιλογίου της άλλης.
- *Διαδικασία ευθυγράμμισης (alignment):*
  Αποτελείται από ένα σύνολο δυαδικών σχέσεων μεταξύ αμφότερων των λεξιλογίων των δύο οντολογιών.
- *Διαδικασία επικοινωνίας (articulation):*
  Η διαδικασία χρήσης ενδιάμεσης οντολογίας, προκειμένου να επιτευχθεί ο ενδιάμεσος τρόπος αντιστοίχησης των εννοιών των δύο οντολογιών.
- *Διαδικασία ενοποίησης (merging):*
  Η διαδικασία ενοποίησης περιλαμβάνει την σύνθεση των δύο οντολογιών. Ο γενικός τρόπος με τον οποίο αυτή επιτυγχάνεται στις διάφορες μεθοδολογίες, είναι μέσω της υλοποίησης της ελάχιστης ένωσης (minimal union), μεταξύ των δύο λεξιλογίων και των αξιωμάτων που διέπουν τις δύο οντολογίες και συνεχίζουν να ισχύουν στο προηγούμενο βήμα (articulation).

2.2. Προηγουμενες Προσεγγισεις Αντιστοιχησης Οντολογιων

Πρόσφατα, ο αριθμός των τεχνικών αντιστοίχησης οντολογιών έχει αυξηθεί σημαντικά (http://www.ontologymatching.org)

Οι τεχνικές αντιστοίχησης οντολογιών ποικίλουν στα πρότυπα εισόδου και εξόδου καθώς και στους τρόπους της παρέμβασης του χρήστη. Έχει υπάρξει μη επαρκής μελέτη στην υπολογιστική αξιολόγηση των τεχνικών αντιστοίχησης οντολογιών στη βιβλιογραφία (π.χ. (F. Giunchiglia, M. Yatskevich, P. Avesani, P. Shvaiko, 2009)ʹ (Y. Kalfoglou and M. Schorlemmer, 2003), (S.Kaza, and H. Chen, 2008).

Παρακάτω, παρουσιάζονται σχετιζόμενες τεχνικές αντιστοίχησης οντολογιών που εστιάζουν στον τρόπο της παρέμβασης του χρήστη.

- Η τεχνική FCA-Merge (G. Stumme, and A. Maedche, 2001) εφαρμόζει τυπική ανάλυση εννοιών (Formal Concept Analysis) για να δημιουργήσει ημι-αυτόματες αντιστοιχήσεις μεταξύ των εξεταζόμενων οντολογιών που έχουν δεδομένα. Οι συγγραφείς υποστηρίζουν ότι συνήθως οι οντολογίες δεν έχουν δεδομένα, δηλαδή δεν έχουν instances, επομένως η τεχνική FCA-Merge θα μπορούσε να γεμίσει τις οντολογίες χρησιμοποιώντας ένα σύνολο κειμένων φυσικής γλώσσας. Όμως, πρέπει να γίνουν πολλές παραδοχές. Τα κείμενα πρέπει να είναι σχετικά με κάθε μια από τις εξεταζόμενες οντολογίες, πρέπει να καλύπτουν όλες τις έννοιες από αυτές και πρέπει να διαχωρίζουν τις έννοιες αρκετά καλά. Οι έννοιες που δεν μπορούν να αντιστοιχιστούν πρέπει να συμπληρωθούν από τον χρήστη (ή το σύνολο των κειμένων πρέπει να επεκταθεί, για να βελτιωθεί η αντιστοίχιση). Η FCA-Merge, μετά την εφαρμογή των συνηθισμένων preprocessing βημάτων (π.χ, stemming, tokenizer, κ.λ.π.), συσχετίζει απλές λέξεις ή σύνθετες εκφράσεις στα κείμενα με μια έννοια από τις οντολογίες. Έπειτα, αναπαριστά το σύνολο των κειμένων που καθορίστηκαν σε κάθε μια οντολογία με το μοντέλο «Vector Space» και παράγει δύο τυπικές έννοιες με σκοπό τελικά να παράγει περιορισμένο δίκτυο εννοιών (η στρατηγική περιορισμού είναι εμπνευσμένη από τον κανόνα συσχέτισης εξόρυξης). Κατά τη διάρκεια της τελευταίας φάσης ο χρήστης με το χέρι, με καθοδήγηση από την FCA-Merge, κατασκευάζει την συγχωνευμένη οντολογία. Αυτή η κατασκευή είναι ημι-αυτόματη καθώς απαιτεί γνώση για το συγκεκριμένο πεδίο της οντολογίας. Ο μηχανικός γνώσης (knowledge engineer), έχει να επιλύσει πιθανές αντιθέσεις και διπλές υπάρξεις εννοιών, αλλά υπάρχει αυτόματη υποστήριξη από την FCA-Merge με τη χρήση ενός μηχανισμού ερωτήσεων / απαντήσεων που βασίζεται στις ευριστικές μεθόδους, που έχει σκοπό να οδηγήσει και να εστιάσει την προσοχή του μηχανικού σε ειδικά μέρη της διαδικασίας κατασκευής, ως προς την τελική αντιστοίχιση.
- Η τεχνική CAIMAN (M. S. Lacher, and G. Groh, 2001) εφαρμόζει μεθόδους ανάκτησης της πληροφορίας, που εφαρμόζεται σε συλλογή κειμένων που ορίστηκαν σε κάθε μια οντολογία, με σκοπό να δημιουργήσει αυτόματα τις αντιστοιχήσεις. Για κάθε μια

έννοια στην πρώτη οντολογία, η CAIMAN υπολογίζει το μέτρο πιθανότητας για κάθε έννοια της δεύτερη οντολογίας να είναι ο αντίστοιχος κόμβος. Ο υπολογισμός της πιθανότητας βασίζεται σε ένα απλό κανόνα συνημίτονου των χαρακτηριστικών διανυσμάτων των δύο εννοιών. Το χαρακτηριστικό διάνυσμα κάθε έννοιας δημιουργείται χρησιμοποιώντας ένα τροποποιημένο διάνυσμα Rocchio.

- Η τεχνική GLUE (A. Doan, J. Madhavan, P Domingos, and A. Halevy, 2002) εφαρμόζει μεθόδους μηχανιστικής μάθησης για να δημιουργήσει ημιαυτόματες σημασιολογικές αντιστοιχήσεις μεταξύ συμπληρωμένων με δεδομένα οντολογιών. Πρώτα, για κάθε έννοια σε μια οντολογία η GLUE βρίσκει την πιο παρεμφερή έννοια στην άλλη οντολογία. Η ομοιότητα προσδιορίζεται σαν την συνδυασμένη κατανομή πιθανότητας (π.χ. συντελεστής Jaccard) των εννοιών σε αντιστοιχία με τα instances των οντολογιών. Με σκοπό να υπολογίσουμε την ομοιότητα, η GLUE χρησιμοποιεί σύνολο ταξινομητών. Τελικά, η GLUE προσπαθεί να αξιοποιήσει τους διαθέσιμους περιορισμούς του πεδίου και των γενικών ευριστικών με σκοπό να βελτιώσει την ακρίβεια της αντιστοίχησης. Πιο ειδικά, η GLUE χρησιμοποιεί «relaxation labeling» για να διαχειρίζεται μια ευρεία ποικιλία περιορισμών ευριστικών και πεδίου. Η παρέμβαση του χρήστη απαιτείται όταν οι έννοιες δεν μπορούν να αντιστοιχιστούν. (3-34% των εννοιών σε αρκετά πραγματικά πεδία).

- Η τεχνική IF-Map (Y. Kalfoglou, and M. Schorlemmer, 2002) εφαρμόζει θεωρία ροής πληροφορίας για να δημιουργήσει αυτόματες αντιστοιχήσεις ανάμεσα σε οντολογίες που έχουν δεδομένα.. Η IF-Map βρίσκει λογικούς infomorphisms, αν υπάρχουν κάποιοι, ανάμεσα στις εξεταζόμενες οντολογίες και τους αναπαριστά σε RDF. Οι λογικοί infomorphisms αναπαριστούν την ροή πληροφορίας ανάμεσα σε ξεχωριστά συστατικά ενός κατανεμημένου συστήματος. Τα συστατικά αναπαρίστανται από τοπικές λογικές που περιγράφουν τα διαφορετικά λεξιλόγια, δηλαδή τις διαφορετικές instances (μαζί με τους διαφορετικούς τους περιορισμούς) και τις ταξινομήσεις σε διαφορετικούς τύπους, που χρησιμοποιούνται σε κάθε συστατικό.

- Η τεχνική ONION (P. Mitra, and G. Wiederhold, 2002) χρησιμοποιεί γλωσσολογικές και δομημένες ευριστικές μεθόδους για να προσδιορίσει τις αντιστοιχήσεις, και

τελικά ημι-αυτόματα να δημιουργήσει τις αντιστοιχήσεις. Οι αντιστοιχήσεις στην συνάρθρωση γίνονται έμμεσα μέσω αντιστοιχήσεων εννοιών των εξεταζόμενων οντολογιών σε μια μοιρασμένη προβολή, που δημιουργείται σκόπιμα. Οι γλωσσολογικές αντιστοιχήσεις βασίζονται είτε σε λεξικά όπως το WordNet ή σε μια συλλογή κειμένων σχετιζόμενα με τις εξεταζόμενες οντολογίες. Οι δομημένες αντιστοιχήσεις ερευνώνται μετά τις γλωσσολογικές αντιστοιχήσεις. Τελικά, οι προτεινόμενες αντιστοιχήσεις πιστοποιούνται από τον χρήστη. Επίσης, η ONION περιλαμβάνει ένα συστατικό μάθησης στο σύστημα που εκμεταλλεύεται τα σχόλια των χρηστών για να δημιουργεί καλύτερες αντιστοιχήσεις στο μέλλον για παρόμοιες οντολογίες.

- Επίσης υπάρχουν κάποιες άλλες τεχνικές που έχουν στόχο να καθοδηγήσουν τον χρήστη να κάνει τις αντιστοιχήσεις. Η τεχνική PROMPT (N. F. Noy, and M. Musen, 2000), (καθώς και οι τεχνικές SMART (N. F. Noy, and M. Musen, 1999) και PROMPTDIFF (N. F. Noy, and M. Musen, 2002)) εφαρμόζει την γλωσσολογική ομοιότητα και ευριστικές μεθόδους για να καθοδηγήσει τον χρήστη στην δημιουργία της αντιστοίχησης. Πρώτα η PROMPT δημιουργεί μια αρχική λίστα από αντιστοιχήσεις με γλωσσολογική ομοιότητα που βασίζεται στα ονόματα των κλάσεων. Έπειτα, ο χρήστης ξεκινά μια διαδικασία που είτε έχει προταθεί από την PROMPT ή χρησιμοποιώντας ένα περιβάλλον editing για οντολογίες. Ας σημειωθεί ότι η PROMPT είναι διαθέσιμη σαν plug-in για τον editor Protégé. Τότε η PROMPT εκτελεί την διαδικασία και αυτόματα εκτελεί επιπρόσθετες αλλαγές, παράγει μια λίστα από προτάσεις, και προσδιορίζει αντιθέσεις. Η PROMPT σχεδιάζεται για να διατηρεί την εστίαση του χρήστη και να παρέχει πληροφορίες στο χρήστη.

- Η CHIMAERA (D.L. McGuinness, R. Fikes, J. Rice, S. Wilder, 2000) είναι ένα παρόμοιο περιβάλλον browser-based editing, και διαγνωστικό εργαλείο για τον Ontolingua editor. Όπως στο PROMPT ο χρήστης είναι υπεύθυνος για να παίρνει αποφάσεις που θα επηρεάσουν τη διαδικασία αντιστοίχησης. Η CHIMAERA αναλύει τις οντολογίες που θα συγχωνευθούν, και αν βρεθούν γλωσσολογικές μέθοδοι, η συγχώνευση

γίνεται αυτόματα, αλλιώς σε διαφορετική περίπτωση δίνεται σήμα στον χρήστη να προχωρήσει για περαιτέρω ενέργειες.

- Η τεχνική του ITTalks (S. Prasad, Y. Peng, and T. Finin, 2004) εφαρμόζει είτε μια στατιστική ευριστική μέθοδο ή ένα αλγόριθμο ταξινόμησης κειμένου με σκοπό να ελεγχθούν όλα οι πιθανές αντιστοιχήσεις ή να ελεγχθούν οι αντιστοιχήσεις που προσδιορίστηκαν από τον χρήστη. Σύμφωνα με την ευριστική μέθοδο, αν μια έννοια στην πρώτη οντολογία ταιριάζει με την πλειοψηφία των παιδιών μιας έννοιας στη δεύτερη οντολογία, τότε η δεύτερη έννοια είναι καλύτερη αντιστοίχηση από τα παιδιά της. Ο αλγόριθμος ταξινόμησης κειμένου χρησιμοποιείται για να δημιουργήσει ένα συντελεστή για κάθε μια έννοια στις οντολογίες. Τότε, όταν ελέγχουμε δυο έννοιες $A \in E\ G1$, $B \in G2$, υπολογίζεται η δεσμευμένη πιθανότητα P(A/B), βασιζόμενη στους συντελεστές των A και B, που προσδιορίζουν την αντιστοίχηση.
- Η τεχνική EER-CONCEPTOOL (E. Compatangelo, and H. Meisel, 2004) εφαρμόζει μια λογική περιγραφής με σκοπό να δημιουργήσει μια προβολή συνάρθρωσης για την αντιστοίχηση των εξεταζόμενων οντολογιών. Εκτελεί ένα αριθμό βημάτων της τυπικής έννοιας και της γλωσσολογικής ανάλυσης με σκοπό να προτείνει στο χρήστη πιθανές αντιστοιχίες. Όμως, σε κάθε βήμα ο χρήστης είναι αποκλειστικά υπεύθυνος για την αποδοχή ή απόρριψη κάθε μίας από τις προτάσεις για τις αντιστοιχίσεις της έννοιας, των χαρακτηριστικών ή των προσδιοριστικών (identifiers). Επιπλέον, ο χρήστης μπορεί να εισαγάγει πιο πολύπλοκες αντιστοιχίσεις ανάμεσα σε ενώσεις ή τομές εννοιών σε κάθε οντολογία βασιζόμενος σε αυτές που προτάθηκαν από το σύστημα.
- Η τεχνική MAFRA (A. Maedche, B. Motik, N. Silva, and R. Volz, 2002) εφαρμόζει σημασιολογική ομοιότητα και ευριστικές μεθόδους βασιζόμενη στην ομοιότητα δυο εννοιών για να πραγματοποιήσει την αντιστοίχηση. Χρησιμοποιεί ένα μέτρο σημασιολογικής ομοιότητας που βασίζεται στην υπόθεση ότι όσο περισσότερη πληροφορία μοιράζονται από κοινού δυο έννοιες, τόσο πιο όμοιες είναι. Η πληροφορία που μοιράζεται από τις δυο έννοιες προσδιορίζεται από το πληροφοριακό περιεχόμενο των πιο ειδικών subsumer. Αφού έχουν προσδιοριστεί

οι ομοιότητες η ευριστική φάση της «σημασιολογικής γεφυροποίησης – semantic bridging» είναι υπεύθυνη για α) προσδιορισμό των αντιστοιχήσεων μεταξύ των εννοιών, των χαρακτηριστικών και των σχέσεων, β) υποστήριξη της αντιστοίχησης με «γέφυρες» για έννοιες που δεν έχουν μια συγκεκριμένη αντίστοιχη έννοια-στόχο.

- Η τεχνική S-MATCH (F. Giunchiglia, P.Shvaiko, M. Yatskevich, 2004) επικεντρώνεται στην σημασιολογική αντιστοίχηση και εφαρμόζει γλωσσολογικές και βασισμένες στη λογική ευριστικές μεθόδους για να δημιουργήσει αυτόματα την ισχυρότερη σημασιολογική σχέση ανάμεσα σε κάθε ζευγάρι εννοιών στις εξεταζόμενες οντολογίες. Διαχωρίζει ανάμεσα σε μια έννοια που σχετίζεται με μια ετικέτα (αυτό που η ετικέτα σημαίνει στον πραγματικό κόσμο) και μια έννοια που σχετίζεται με έναν κόμβο (αυτό που σημαίνουν οι έννοιες των ετικετών που προσδιορίζουν τους κόμβους). Έτσι, σε μια προ-διαδικαστική φάση όλες οι ετικέτες μεταφράζονται σε μια εσωτερική γλώσσα, που βασίζεται στην αναλογική λογική, μετά τη λεκτική επεξεργασία. Έπειτα, υπολογίζει τις σχέσεις ανάμεσα σε κάθε ζευγάρι εννοιών ετικετών, βασιζόμενη σε δύο πηγές a priori γνώσεις: η μία βγαίνει από λεκτικές ομοιότητες χρησιμοποιώντας χειρισμό strings, και μια που προκύπτει από το Wordnet χρησιμοποιώντας τις «αισθήσεις».
- Η τεχνική OLA (J. Euzenat, P.Valtchev, 2004) εφαρμόζει σημασιολογική ομοιότητα ανάμεσα σε έννοιες, που βασίζεται στις ομοιότητες των εννοιών (contributor pairs) σχετιζόμενες με εκείνες που πρόκειται να ελεγχθούν (anchor pair). Χρησιμοποιεί διαφορετικά μέτρα σημασιολογικής ομοιότητας για να υπολογίσει την ομοιότητα ενός ζευγαριού. Σε ένα από τα 8 υποστηριζόμενα στοιχεία μιας οντολογίας, δηλαδή κατηγορίες (π.χ. κλάση, αντικείμενο, σχέση κ.λ.π.), ορίζεται ένα διαφορετικό μέτρο ομοιότητας. Όλα αυτά σχηματίζουν την τελική ομοιότητα χρησιμοποιώντας βάρη.

Επιπλέον, υπάρχουν τεχνικές που βασίζονται στο συνδυασμό των διαφορετικών διαδικασιών αντιστοίχησης (D. Aumueller, H.H. Do, S. Massmann, and E. Rahm, 2005), (W. Hu., and Y. Qu, Falcon-AO, 2008), που παρουσιάζουν αξιοσημείωτα αποτελέσματα με όρους ακρίβειας (http://oaei.ontologymatching.org).

Επίσης, υπάρχουν τεχνικές που δυνητικά θα μπορούσαν να χρησιμοποιηθούν στην αντιστοίχηση οντολογιών όπως μεταφραστές (π.χ. OntoMorph (H. Chalupksy, 2000)) ή

ολοκληρωτές (π.χ. (E. Hovy, 1998)). Τελικά, ένα παρόμοιο πρόβλημα είναι αυτό της αντιστοίχησης του σχήματος σε βάσεις δεδομένων. Όμως, οι περισσότερες τεχνικές αντιστοίχησης βάσεων δεδομένων δεν είναι επαρκείς για την αντιστοίχηση οντολογιών για το λόγο ότι δεν χειρίζονται διαφορές στην ορολογία, ότι παρουσιάζουν φτωχά αποτελέσματα στην περίπτωση της μικρής δομημένης ομοιότητας, στην απουσία instances κ.λ.π.

2.3. Αξιολογηση - Αποδοτικοτητα Στην Αντιστοιχηση Οντολογιων

Παρακάτω παρουσιάζονται η πολυπλοκότητα της κάθε μεθοδολογίας ως προς τον χρόνο, όπου θεωρήθηκαν για τα χειρότερα σενάρια.

- Στην FCA - Merge τεχνική, το πρόβλημα του υπολογισμού των πλεγμάτων εννοιών (concept lattices) έχει εκθετική πολυπλοκότητα στη χειρότερη περίπτωση (worst case). Πρακτικά, ο αλγόριθμος Next Closure για τον υπολογισμό των πλεγμάτων εννοιών είναι O(B(K) * |D| * (|G1| + | G2|) (Chinchor N., Robinson P., 1997)), ενώ χρησιμοποιείται και ο αλγόριθμος Nourine/Raynaud με πολυπλοκότητα O(B(K) * (|D| + (|G1| + | G2|)) * (|D|)., Επίσης ας σημειωθεί ότι η χρονική πολυπλοκότητα του αλγόριθμου TITANIC, ενός τρίτου αλγορίθμου που χρησιμοποεί σε ειδικού τύπου lattices (iceberg lattices) η FCA – Merge, είναι ανάμεσά τους. Στά ανωτέρω, όπου K είναι ένα περιεχόμενο (context), B(K) το πλέγμα εννοιων του περιεχομένου (concept lattice) και D ο αριθμός των λαμβανομένων υπο εξέταση εγγράφων (documents)

- Η τεχνική CAIMAN απαιτεί μια χρονοβόρα φάση με σκοπό να δημιουργηθεί το χαρακτηριστικό διάνυσμα κάθε έννοιας στις εξεταζόμενες οντολογίες. Ο υπολογισμός ενός πίνακα ομοιότητας για οποιεσδήποτε έννοιες, βασιζόμενος στα χαρακτηριστικά διανύσματα, είναι O(|G1| * | G2|).

- Η GLUE είναι μια χρονοβόρα τεχνική. Για κάθε ζευγάρι εννοιών (A, B), A∈E G1, B∈G2, υπολογίζει την ομοιότητά τους εφαρμόζοντας τον Naive Bayesian

συντελεστή. (θεωρώντας ότι όλοι οι βασικοί learners εκπαιδεύονται ταυτόχρονα, που είναι O(|G1| * | G2| * (|n1| + |n2|)), όπου n1, n2, είναι instances των G1,G2 αντίστοιχα. Αυτό συμβαίνει γιατί ο Naive Bayesian συντελεστής είναι γραμμικός με το εκπαιδευόμενο σύνολο. Επίσης, η πολυπλοκότητα του "relaxation labeling", που εξαρτάται από τους συντελεστές ομοιότητας, είναι γραμμική με των αριθμό ετικετών, επομένως είναι O(|G1| + | G2|). Ας σημειωθεί επίσης ότι οι ιδιότητες σύγκλισης δεν είναι ακόμα καλά κατανοητές και είναι υποκείμενες για τη σύγκλιση σε ένα τοπικό μέγιστο.

- Η IF-MAP έχει μια εκθετική χρονική πολυπλοκότητα και έτσι δεν μπορεί να εφαρμοστεί σε μεγάλου μεγέθους οντολογίες. Αυτό συμβαίνει γιατί πρέπει αυτόματα να παράγει όλες τις πιθανές αντιστοιχίσεις ανάμεσα στις οντολογίες. Επομένως, ο χώρος αναζήτησης είναι εκθετικός και αν προσπαθήσει η IF-MAP να το μειώσει χρησιμοποιώντας τις instances και χρησιμοποιώντας τον περιορισμό ότι η αντιστοίχηση πρέπει να σέβεται την ιεραρχία εννοιών.

- Στην τεχνική ONION, οι γλωσσικοί συνδυασμοί που βασίζονται σε λεξικά απαιτούν μια εξαντλητική αναζήτηση σε εξαιρετικά μεγάλου μεγέθους λεξικά με πολλές χιλιάδες έννοιες, όπως το WordNet με 106 έννοιες. Επομένως, μια κατασκευή ομοιότητας λέξης στο WordNet είναι O(|G1| * | G2| * |GWordNet|).

- Οι γλωσσικοί συνδυασμοί που βασίζονται σε ένα σύνολο κειμένων απαιτούν μια χρονοβόρα προ-επεξεργασία με σκοπό να κατασκευάσουν το διάνυσμα του πλαισίου κάθε έννοιας στις εξεταζόμενες οντολογίες. Ο υπολογισμός του πίνακα ομοιότητας για οποιεσδήποτε έννοιες, που βασίζεται στα διανύσματα πλαισίου, είναι O(|G1| * | G2|). Επίσης, ας σημειωθεί ότι ο ισομορφισμός του γράφου είναι ένα NP πρόβλημα.

- Η τεχνική αντιστοίχησης του συστήματος ITTalks απαιτεί για κάθε ζευγάρι εννοιών είτε να ελέγξει τα παιδιά τους ή να ταξινομήσει τα καθορισμένα υποδειγματικά κείμενα σε κάθε έννοια. Επομένως, η πολυπλοκότητά της είναι O(|G1| * | G2| * D * O(classify)), όπου D είναι ο μέσος αριθμός υποδειγματικών κειμένων που ορίζονται σε μια έννοια και O(classify) η πολυπλοκότητα του της ταξινόμησης ενός κειμένου, μια μάλλον πολύπλοκη διαδικασία.

- Στην τεχνική EER-CONCEPTOOL χρησιμοποιούνται διαφορετικοί μηχανισμοί αναφοράς στην περιγραφική λογική με σκοπό να εκτελεστούν οι τυπικές αναλύσεις εννοιών και πλαισίου αναφοράς. Παρόλο που οι οντολογίες αναπαρίστανται χρησιμοποιώντας την περιγραφική λογική, χωρίς απώλεια της γενικότητας, μια τυπική ανάλυση εννοιών μειώνεται σε traversals των δομών των οντολογιών, για κάθε έννοια της οντολογίας Ga που είναι O((|G1| + | G2|) * Ga). Επιπλέον, η γλωσσολογική ανάλυση απαιτεί, για κάθε έννοια των εξεταζόμενων οντολογιών, μια έρευνα στο WordNet, ένα εξαιρετικά μεγάλο λεξικό με 106 έννοιες, που είναι O((|G1| + | G2|) * |GWordNet|).
- Στην τεχνική MAFRA η χρονική πολυπλοκότητα της υιοθετούμενου σημασιολογικού μέτρου ομοιότητας για δύο έννοιες είναι O(d), όπου d είναι το βάθος της οντολογίας. Επομένως η χρονική πολυπλοκότητα υπολογίζοντας ένα πίνακα ομοιότητας είναι O(|G1| * | G2| * d). Η φάση σημασιολογικής γεφύρωσης μειώνεται για να έχει πρόσβαση στον πίνακα ομοιότητας και να εκτελέσει οντολογικά traversals , που δεν είναι σημαντικά πιο χρονοβόρα από τον υπολογισμό του πίνακα ομοιότητας.
- Στην τεχνική S-MATCH η πιο χρονοβόρα φάση είναι ο υπολογισμός του πίνακα CN. Για κάθε ζευγάρι εννοιών κόμβων πρέπει να εφαρμοστεί ένας ικανοποιητικός λύτης του προβλήματος. Το πρόβλημα της «ικανοποιητικότητας» (satisfiability problem) είναι ένα γνωστό πρόβλημα NP πληρότητας που απαιτεί εκθετικό χρόνο. Η συνολική πολυπλοκότητα αυτής της φάσης είναι O(|G1| * | G2| * 2V), όπου Ω ο αριθμός των λογικών μεταβλητών στις προτάσεις που αναπαρίστανται στην background γνώση.
- Στην τεχνική OLA, για κάθε ζευγάρι εννοιών εκτελείται μια επαναληπτική διαδικασία με σκοπό να υπολογίσει την ομοιότητα. Η πολυπλοκότητα αυξάνεται αν υπάρχουν επαναλαμβανόμενες εξαρτήσεις (2 ζευγάρια κόμβων συμβάλλουν το ένα στο άλλο). Γενικά, ένα μη γραμμικό πρόβλημα πρέπει να λυθεί με σκοπό να υπολογίσει την ομοιότητα.

Τελικά, ας σημειωθεί ότι πρόσφατα υπάρχει ένα ενδιαφέρον για την αποδοτικότητα των προτεινόμενων τεχνικών αντιστοίχησης οντολογιών στη βιβλιογραφία. Για παράδειγμα, η εργασία που παρουσιάζεται στο (J. R. Quinlan, 1987) προσπαθεί να μειώσει το χώρο

αναζήτησης εισάγοντας συγκεκριμένες στρατηγικές για να επιλέξει τα ζευγάρια εννοιών ελέγχοντας για την αντιστοίχηση. Έτσι, η χρονική πολυπλοκότητα μειώνεται σε $O((|G_1| + |G_2|) * \log(|G_1| + |G_2|))$.

Κεφάλαιο 3. Προτεινόμενη Μεθοδολογία

3.1. Ερευνητικα Θεωρητικα Θεματα

Στο δεύτερο μέρος παρουσιάζεται αναλυτικά η μεθοδολογία ONARM σχετικά με την αντιστοίχηση εννοιών μίας οντολογίας στις έννοιες μίας άλλης. Η μεθοδολογία αυτή είναι βασισμένη σε κανόνες συσχέτισης, βάσει των οποίων γίνεται η εξόρυξη δεδομένων.

3.1.1. Γενικη Παρουσιαση

Η μηχανική των οντολογιών (Ontology Engineering), δηλαδή ο σχεδιασμός, η ανάπτυξη, η συντήρηση και η διαμοίραση των οντολογιών, είναι μια αναπτυσσόμενη διαδικασία της Μηχανικής της Γνώσης. Επιτρέπει την οργάνωση της πληροφορίας σε ταξονομίες των εννοιών, που αναπαρίστανται με χαρακτηριστικά (attributes), και σχέσεις μεταξύ εννοιών, οι οποίες αναπαρίστανται με σχέσεις τύπου IS-A, συναρτήσεις, περιορισμούς κλπ. Οι οντολογίες βρίσκουν αποδοχή σε πολλές εφαρμογές, π.χ. στην ανάκτηση πληροφοριών (A. Pretschner, and S. Gauch, 1999), στη διαχείριση κειμένων (M. S. Lacher, and G. Groh, 2001), στην επικοινωνία ευφυών διαμεσολαβητών (M. N. Huhns, and M.P. Singh, 1997), στην οικονομία (A. Firat and S. Madnick, 2001), και στο ηλεκτρονικό εμπόριο(B. Omelayenko, 2001), (Virtual Vineyards: http://www.inc.com/magazine/19960615/1966.html).

Όμως οι οντολογίες επιβάλλονται από την εκρηκτική ανάπτυξη του Σημασιολογικού Ιστού, όπου χρησιμοποιούνται για να περιγράψουν τη σημασιολογία (semantics) των δεδομένων. Χρησιμοποιούνται για την δόμηση των δεδομένων εννοιολογικά και για τη διαμοίραση της γνώσης. Οι οντολογίες μπορούν να σχεδιαστούν και να υλοποιηθούν από διαφορετικές ομάδες ανθρώπων με παρόμοια ενδιαφέροντα, δηλαδή κοινότητες μέσα στην επονομαζόμενη κοινωνία της πληροφορίας, είτε μέσω των διαδικασιών της μηχανικής της γνώσης ή δια μέσου των αυτόματων μεθόδων εξόρυξης της γνώσης.

Μια από τις πιο σημαντικές ιδιότητες του Διαδικτύου και του Σημασιολογικού Ιστού είναι η αποκέντρωση της πληροφορίας (T. Berners-Lee, 1999), (http://www.w3.org/DesignIssues/Principles.html).

. Οι οντολογίες μπορούν να σχεδιαστούν και να υλοποιηθούν από διαφορετικές κοινότητες χωρίς να υιοθετούν κοινά πρότυπα για την ανταλλαγή πληροφορίας. Επίσης, η χρήση συνεργιών ανταλλαγής πληροφοριών (interoperability processes), έχει αυξηθεί με την ανάπτυξη συστημάτων για την υποστήριξη της αλληλεπίδρασης διαφορετικών τύπων κοινοτήτων χρηστών. Όμως, πολλοί ερευνητές (π.χ. (M. S. Lacher, and G. Groh, 2001), (A. Maedche, and S. Staab, 2000), (P. Mitra, and G. Wiederhold, 2002), (G. Stumme, and A. Maedche, 2001)), υποστηρίζουν ότι, η ευρύτερη αποδοχή από τους χρήστες κοινών έννοιων, σε πολλαπλά συστήματα οντολογιών, για πολλούς γωσιακούς χώρους, δεν δύναται να θεωρείται πάντα δεδομένη, από όλους τους χρήστες (H. Wache, T. V¨ogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hubner, 2001). Κρίνεται δε, πιο αποδοτικό αν μια μικρότερη κοινότητα χρηστών εμπλέκεται στη διαδικασία δημιουργίας οντολογιών για συσγκεκριμένο γνωσιακό χώρο. Γενικά, οι κοινότητες χρηστών δεν μπορούν να εξαναγκαστούν να υλοποιήσουν κοινά πρότυπα κατά την υλοποίση οντολογιών που να καλύπτουν ταυτόχρονα τα πολλαπλά γνωσικά τους πεδία. Έπειτα η πραγμτοποίση μιας αποδοτική ανταλλαγή πληροφοριών, που βασίζεται σε δύο οντολογίες οντολογία, ανάγεται στο πρόβλημα του προσδιορισμού των αναλογιών μεταξύ τους, δηλαδή τον προσδιορισμό του συνόλου των παρόμοιων, επικαλυπτόμενων ή μοναδικών εννοιών τους και άλλων ενδιαφερόντων χαρακτηριστικών τους. Αυτό το πρόβλημα είναι μια περίπτωση του γενικώτερου προβλήματος της διαλειτουργικότητος (π.χ. (J.Park, and S. Ram,2004)).

Η μηχανική των δεδομένων και της γνώσης έχει εστιάσει στον προσδιορισμό των αναλογιών τόσο ανάμεσα στις οντολογίες όσο και ανάμεσα στα σχήματα των βάσεων δεδομένων αντίστοιχα (π.χ. (N. Choi, I.Y. Song, and H. Han, 2006), (Y. Kalfoglou and M. Schorlemmer, 2003), (E. Rahm, and P.A. Bernstein, 2001), (P. Shvaiko, and J. Euzenat, 2005)).

Οι οντολογίες χαρακτηρίζονται από ετερογένεια και ασυμβατότητα ως προς τις έννοιες τους. Η ετερογένεια και η ασυμβατότητα οφείλεται στην ύπαρξη της γνώσης σε διαφορετικές δομές ή σε διαφορετικά περιβάλλοντα με διαφορετικές σημασιολογίες, παρ' όλο που μπορεί να περιγράφονται με τις ίδιες λέξεις. Η αντιστοίχηση της οντολογίας έχει σκοπό να ξεπεράσει την δομική και σημασιολογική ετερογένεια και ασυμβατότητα, ορίζοντας αναλογίες μεταξύ στοιχείων ανόμοιων οντολογιών. Η δομική ετερογένεια έχει αναλυθεί και μελετηθεί σε μεγάλη έκταση στη βιβλιογραφία ταιριάσματος σχήματος (schema alignment) (E. Rahm, and P.A. Bernstein, 2001). Μία αντιστοίχηση μπορεί να

δημιουργηθεί είτε άμεσα μεταξύ δυο οντολογιών (ευθυγράμμιση - alignment) ή έμμεσα, δια μέσου μιας αντιστοίχησης, σε μια τρίτη οντολογία, που και οι δυο από αυτές μοιράζουν ένα κοινό ανώτερο μοντέλο (συνάρθρωση – articulation).

Η εργασία της αντιστοίχησης οντολογιών γίνεται περισσότερο με το χέρι, ίσως υποστηριζόμενη με μια γραφική διεπαφή. Φυσικά, κάνοντας την αντιστοίχηση των οντολογιών με το χέρι είναι μία εξαιρετικά χρονοβόρα και επιρρεπής σε λάθη διαδικασία. Οι τεχνικές αντιστοίχησης οντολογιών που παρουσιάζονται στην βιβλιογραφία συνήθως βασίζονται σε συντακτικές και/ ή σημασιολογικές ευριστικές μεθόδους. Οι τελευταίες έχουν μελετηθεί σε ευρεία επιστημονικά πεδία συμπεριλαμβάνοντας την μηχανιστική μάθηση, τα πλέγματα εννοιών, τις σχηματικές θεωρίες, τις βάσεις δεδομένων και την γλωσσολογία. Σχεδόν σε όλες από αυτές η παρέμβαση του χρήστη είναι απαραίτητη, επομένως είναι ημι-αυτόματες. Συνήθως, όταν δεν είναι δυνατή μια αυτόματη απόφαση, αυτές οι τεχνικές προτείνουν πιθανές αναλογίες, προσδιορίζουν αντιφάσεις και προτείνουν λύσεις και ενέργειες. Τότε ο χρήστης κάνει την τελική επιλογή.

Στην τρέχουσα εργασία, παρουσιάζουμε μια νέα τεχνική αντιστοίχησης οντολογιών, την τεχνική ONARM, η οποία, αφού δοθούν δύο οντολογίες, είναι ικανή να ταιριάξει τις έννοιες στην μια οντολογία σε εκείνες της άλλης, χωρίς καμία εμπλοκή του χρήστη, αυτόματα. Η προτεινόμενη τεχνική εξετάζει την δομή των προς διερεύνηση οντολογιών, δηλαδή τις ιεραρχίες των εννοιών για να προσδιορίσει την αντιστοίχηση. Ειδικότερα, βασίζεται στον κανόνα συσχέτισης εξόρυξης για να εξάγει τους κανόνες συσχέτισης από αυτές τις ιεραρχίες εννοιών. Η συσχέτιση είναι μια από τις πιο δημοφιλείς εργασίες εξόρυξης δεδομένων. Οι κανόνες συσχέτισης μπορούν να χρησιμοποιηθούν για να αναπαραστήσουν συχνά σχήματα στα δεδομένα, στη μορφή των εξαρτήσεων ανάμεσα στις έννοιες-χαρακτηριστικά. Οι εξαγόμενοι κανόνες συσχέτισης θεωρούνται ως έμμεσοι στην περιγραφή των σχέσεων των εννοιών. Ας σημειωθεί ότι, παρόλη την βάση ή την αντιπαράθεση της δήλωσης ότι η αντιστοίχηση των οντολογιών είναι παρόμοια με την αντιστοίχηση των σχημάτων των βάσεων δεδομένων (Y. Kalfoglou and M. Schorlemmer, 2003), (N. F. Noy, and M. Klein, 2002) η προτεινόμενη μεθοδολογία μπορεί να εφαρμοστεί και στις δύο περιπτώσεις.

Επίσης παρουσιάζονται προηγούμενες προσπάθειες και παρόμοια συστήματα, η καθεαυτού τεχνική και πειραματικά αποτελέσματα του ελέγχου της ακρίβειάς της και της

αποδοτικότητας της. Επίσης αναλύεται η χρονική της πολυπλοκότητα σε σχέση με άλλες μεθοδολογίες.

3.1.2. Η ΠΡΟΤΕΙΝΟΜΕΝΗ ΤΕΧΝΙΚΗ ONARM

3.1.2.1. ΓΕΝΙΚΑ

Η βασική ιδέα της προτεινόμενης τεχνικής είναι να δημιουργήσει μια ομοιότητα ανάμεσα σε 2 έννοιες των εξεταζόμενων οντολογιών O_1, O_2, που βασίζεται στις τοποθετήσεις τους στις δομές της οντολογίας και να γενικεύσει αυτό για το σύνολο των εννοιών τους (Tatsiopoulos C., Boutsinas B., submitted).

Η τοποθεσία ενός κόμβου, που αναπαριστά μια έννοια μέσα σε μια δομή μιας οντολογίας, προσδιορίζει τις γειτονικές της έννοιες. Θεωρούμε ότι η σημασία μιας έννοιας επίσης χαρακτηρίζεται από την σημασία των γειτονικών της εννοιών, όπως ο δημιουργός της οντολογίας έμμεσα προσδιόρισε ορίζοντας τη δομή της οντολογίας.

Ας σημειωθεί ότι από μόνη της η δομική αντιστοίχηση δεν είναι ικανοποιητική για την αντιστοίχηση των οντολογιών. Φυσικά, η σημασία της έννοιας επίσης χαρακτηρίζεται από τη γλωσσολογική ανάλυση της έννοιας με τη βοήθεια ενός λεξιλογίου μεγάλης κλίμακας όπως το WordNet, με τη χρήση μιας συλλογής κειμένων, λεξικογραφικές αποστάσεις κ.λ.π. Η προτεινόμενη τεχνική αποδέχεται όλες αυτές τις πηγές γνώσης με σκοπό να δημιουργηθεί ένα μέτρο ομοιότητας. Όμως, η τελευταία προσδιορίζεται από τη θέση μιας έννοιας μέσα στην οντολογία.

Οι τεχνικές αντιστοίχησης γράφου θα μπορούσαν να χρησιμοποιηθούν με σκοπό να εξετάσουν την ομοιότητα της θέσης 2 εξεταζόμενων εννοιών. Αφού επικεντρωνόμαστε στην αποδοτικότητα, απορρίψαμε τέτοιες τεχνικές εξαιτίας τις χρονικής πολυπλοκότητάς τους (για παράδειγμα η χρονική πολυπλοκότητα του ισομορφισμού γράφων είναι εκθετική). Η προτεινόμενη τεχνική θεωρεί κάθε μονοπάτι της δομής της οντολογίας ως μια πηγή γνώσης. Εφαρμόζει τον κανόνα συσχέτισης εξόρυξης γνώσης με σκοπό να προσδιορίσει τις κυρίαρχες γειτονικές έννοιες μιας εξεταζόμενης οντολογίας.

Στην παρούσα Δ.Δ., θεωρούμε τον κανόνα συσχέτισης εξόρυξης γνώσης που είναι γνωστός ως «the market basket problem», όπου οι έννοιες-χαρακτηριστικά αναπαριστούν προϊόντα και η αρχική βάση δεδομένων είναι ένα σύνολο αγορών καταναλωτών (συναλλαγές). Αυτό το ιδιαίτερο πρόβλημα έχει μελετηθεί εκτενώς στην εξόρυξη δεδομένων. Θεωρούμε κανόνες σχέσεων ανάλογους με τη μορφή «90% των καταναλωτών που αγοράζουν το προϊόν x επίσης αγοράζει και το προϊόν y» (π.χ (Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A.I., 1996), (S. Brin, R. Motwani, J.D. Ullman, S. Tsur, 1997), (J. S. Park, M. S. Chen, P. S. Yu, 1995)). Τυπικά, ο κανόνας συσχέτισης είναι ένας κανόνας της μορφής $X \Rightarrow Y$, όπου X,Y ονομάζονται αντίστοιχα προηγούμενος και επόμενος του κανόνα και $X, Y \subset I = \{i_1, i_2, \ldots i_j\}$, έτσι ώστε $X \cap Y = \emptyset$ και i_k, $1 \leq k \leq j$ είναι ένα στοιχείο στη βάση δεδομένων των συναλλαγών D. Η πληροφοριακή δύναμη (που ονομάζεται interestingness) καθενός κανόνα συσχέτισης μετριέται με δυο δείκτες: τον δείκτη Υποστήριξης (Support) που μετρά την αναλογία των συναλλαγών στην D περιέχοντας και το X και το Y και τον δείκτη Εμπιστοσύνης (Confidence) που μετρά την εξαρτημένη πιθανότητα του επόμενου δεδομένου του προηγούμενου.

Ειδικότερα, η προτεινόμενη τεχνική θεωρεί κάθε μονοπάτι της δομής της οντολογίας ως μια συναλλαγή. Έπειτα, εφαρμόζει τον κανόνα συσχέτισης εξόρυξης στο σύνολο συναλλαγών για κάθε οντολογία. Θεωρούμε ότι οι εξαγόμενοι κανόνες συσχέτισης προσδιορίζουν τις προηγούμενες γειτονικές έννοιες για κάθε εξεταζόμενη έννοια. Επομένως, η ομοιότητα αυτών των κανόνων συσχέτισης ορίζει την ομοιότητα των εννοιών που βασίζεται στην τοποθεσία.

Η γλωσσολογική ανάλυση λαμβάνεται επίσης υπόψη. Όμως, χρησιμοποιείται για να αυξήσει ή να μειώσει αποκτούμενη ομοιότητα που βασίζεται στην τοποθεσία. Με αυτή την έννοια, μπορεί να χρησιμοποιηθεί κάθε ευριστική ή γλωσσολογική ανάλυση που προτείνεται στην βιβλιογραφία. Επίσης, μπορεί να χρησιμοποιηθεί το σύνολο των αποτελεσμάτων τέτοιων ευριστικών μεθόδων, όπως για το παράδειγμα που παρουσιάστηκε στα (M. Ehrig, and S. Staab, 2003), (M. Ehrig, and Y. Sure, 2005). Σε αυτή τη δημοσίευση, υιοθετούμε μια τέτοια ευριστική μέθοδο: εξετάζουμε την ταυτότητα των ετικετών των εννοιών, ενώ χρησιμοποιούμε ένα κοινό λεξιλόγιο και για τις δυο οντολογίες.

Η προτεινόμενη τεχνική μπορεί να εφαρμοστεί σε δομές οντολογιών που σχηματίζουν ένα άμεσο ακυκλικό γράφο. Έτσι, υποστηρίζει πολλαπλή κληρονομικότητα. Ο απαιτούμενος

τυπικός ορισμός των εξεταζόμενων οντολογιών περιέχει δύο κύρια κομμάτια που διαμοιράζονται από τους πιο τυπικούς ορισμούς μιας οντολογίας στη βιβλιογραφία: έννοιες και μια ιεραρχική σχέση IS-A. Έτσι ορίζουμε την οντολογία ως ακολούθως:

Μία οντολογία είναι ένα ζευγάρι G=(C,r), όπου C είναι ένα σύνολο από έννοες και r είναι μια μερική σειρά του C, δηλαδή μια δυαδική σχέση r $\in$ C x C, η οποία είναι αυτοπαθής, μεταβατική και αντισυμμετρική.

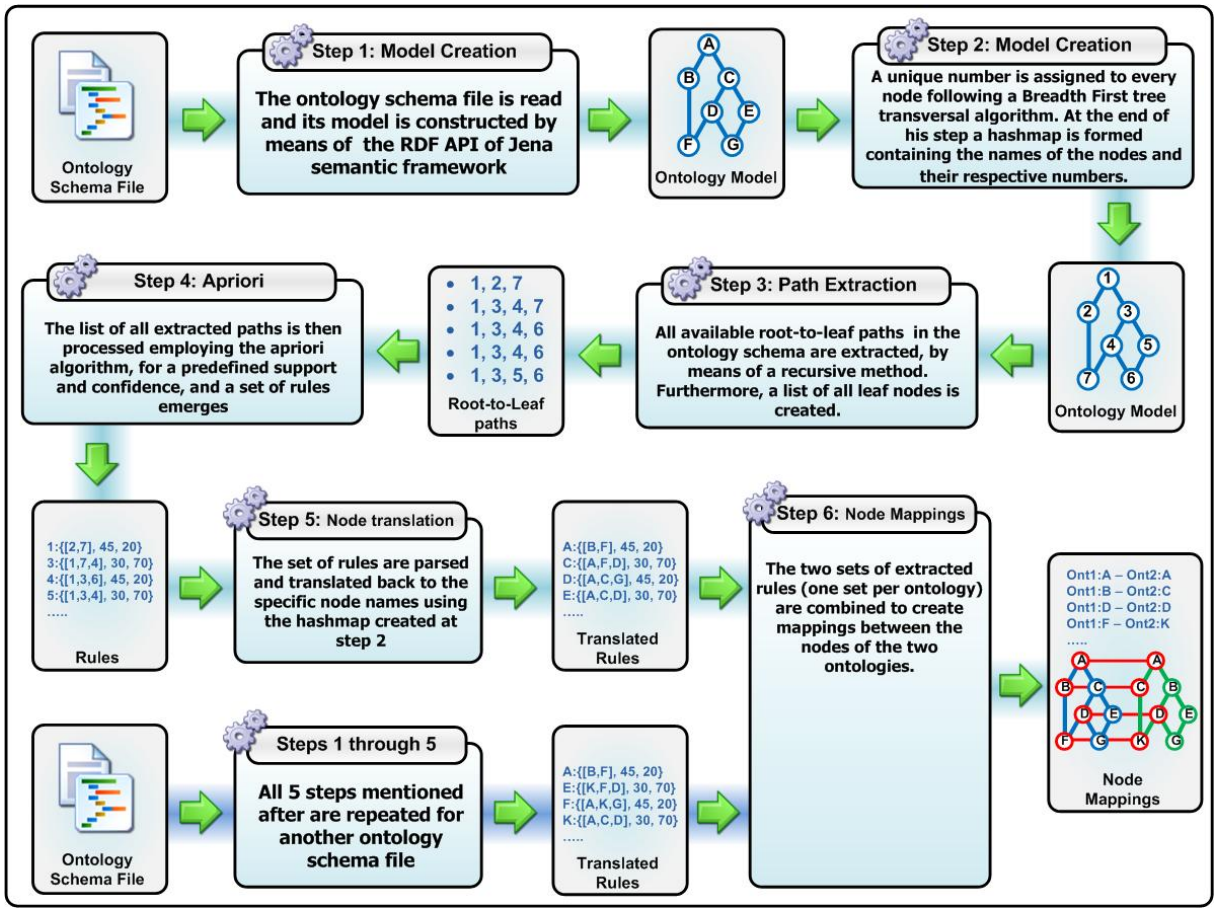
3.1.2.2. ΑΝΑΛΥΤΙΚΗ ΠΑΡΟΥΣΙΑΣΗ ΤΗΣ ΠΡΟΤΕΙΝΟΜΕΝΗΣ ΜΕΘΟΔΟΛΟΓΙΑΣ

Η προτεινόμενη μεθοδολογία ONARM, αποτελείται από τα ακόλουθα διακριτά βήματα, προκειμένου να δημιουργηθούν συσχετίσεις – αντιστοιχίσεις μεταξύ των κόμβων δύο οντολογιών:

- *Βήμα 1ο – Εισαγωγή & Ανάγνωση Οντολογίας*
- *Βήμα 2ο – Δημιουργία Μοντέλου Μνήμης Οντολογίας*
- *Βήμα 3ο – Εξαγωγή Συνδέσεων Κόμβων*
- *Βήμα 4ο – Επεξεργασία με τον Αλγόριθμο Apriori – Δημιουργία Κανόνων Συσχέτισης*
- *Βήμα 5ο – Αντιστοίχηση Ονοματολογίας Κόμβων (backnaming)*
- *Βήμα 6ο – Αντιστοίχηση Κόμβων Οντολογιών*

Διάγραμμα Ροής Προτεινόμενης Μεθοδολογίας

Παρακάτω παρουσιάζεται η βασική ροή και όλα τα ενδιάμεσα βήματα της προτεινόμενης μεθοδολογίας



Η προτεινόμενη μεθοδολογία παίρνει σαν εισαγωγή δύο οντολογίες, οι οποίες όπως φαίνεται στο ανωτέρω σχήμα εισάγονται στο σύστημα σαν δύο Ontology Schema Files. Κατόπιν το σύστημα τις επεξεργάζεται ανεξάρτητα. Στα ανωτέρω πλαίσια, θεωρείται μια εκ των δύο οντολογιών Ox, για περαιτέρω περιγραφή της μεθοδολογίας.

Βήμα 1ο – Εισαγωγή & Ανάγνωση Οντολογίας

Όπως παρουσιάζεται στο παραπάνω διάγραμμα, το πρώτο βήμα είναι η οντολογία να απεικονιστεί στο περιβάλλον Protégé, και να σωθεί προς περαιτέρω επεξεργασία στην μορφή RDF/RDFS, σαν ένα Ontology Schema File. Είναι σημαντικό να αναφερθεί στο σημείο αυτό ότι κάθε υπάρχουσα οντολογία η οποία είναι ήδη περιγεγραμμένη σε Protégé, μπορεί να χρησιμοποιηθεί από την προτεινόμενη μεθοδολογία. Ακολούθως χρησιμοποιείται το Application Programming Interface (API), της Java, όπου χρησιμοποιείται το πακέτο Jena, το οποίο επεξηγήθηκε σε προηγούμενο κεφάλαιο λεπτομερώς. Κατ' αυτόν τον τρόπο όπως φαίνεται και στο διάγραμμα, διαβάζεται η περιγεγραμμένη σε textual μορφή (RDF / RDFS) οντολογία και μετατρέπεται σε μορφή επεξεργάσιμη και αποθηκευμένη στην κύρια μνήμη

του συστήματος, περιέχοντας και όλα τα δομικά στοιχεία αποθηκευμένα σε ξεχωριστές μεταξύ τους προγραμματιστικές δομές. Αυτά τα δομικά στοιχεία της οντολογίας είναι οι συνδέσεις μεταξύ των κόμβων – εννοιών (paths), οι κόμβοι – έννοιες (nodes – concepts), η έννοια του «τρέχοντος κόμβου» (current node), οι κόμβοι – γονείς (parent nodes), οι κόμβοι – αδέρφια (sibling nodes), οι κόμβοι – παιδιά (children nodes) καθώς και σε ξεχωριστή δομή οι κόμβοι – φύλλα (leaf nodes) της υπό εξέταση δενδρικά αναπαριστάμενης οντολογίας – γράφου.

Βήμα 2ο – Δημιουργία Μοντέλου Μνήμης Οντολογίας

Κατά το δεύτερο βήμα της προτεινόμενης μεθοδολογίας, δημιουργείται μία αριθμημένη κατά αύξουσα αρίθμηση λίστα των κόμβων της οντολογίας, έτσι ώστε να διατηρείται η ιεραρχική δομή της οντολογίας, υπό την μορφή μίας σειράς ακεραίων αριθμών. Η αρίθμηση επιτυγχάνεται χρησιμοποιώντας τον αλγόριθμο Breadth First Search (BFS), ο οποίος έχει περιγραφεί αναλυτικά σε προηγούμενο τμήμα της παρούσης. Κατ' αυτόν τον τρόπο, οι ακέραιοι αυξάνονται οριζόντια, αριθμώντας την δενδρική δομή και κάθε τυχαίος κόμβος Ni έχει αριθμηθεί με ένα ακέραιο k έτσι ώστε k > m, όπου m είναι ένας άλλος ακέραιος, ο οποίος έχει αριθμήσει τον parent node M, του τρέχοντος κόμβου N. Κατ' αυτόν τον τρόπο η μεθοδολογία εγγυάται ότι για κάθε ζεύγος κόμβου τύπου child – parent, το παραπάνω ισχύει πάντα, σε όλη την οντολογία. Επίσης, προκύπτει, ότι οι κόμβοι σε βαθύτερα επίπεδα (deeper levels) στη δενδρική δομή του γράφου, έχουν πάντα και τις μεγαλύτερες αριθμητικές τιμές στους κόμβους τους. Το αποτέλεσμα στο βήμα 2, είναι μία προγραμματιστική δομή τύπου hashtable, η οποία περιλαμβάνει όλους τους κόμβους με τους αντιστοίχους ακεραίους της αρίθμησής τους.

Βήμα 3ο – Εξαγωγή Συνδέσεων Κόμβων

Κατόπιν, στο 3ο βήμα της μεθοδολογίας, εκτελείται μία Depth First Search (DFS) η οποία έχει παρουσιαστεί αναλυτικά σε προηγούμενο κεφάλαιο της παρούσης, με σκοπό να εξαχθούν όλα τα πιθανά μονοπάτια της οντολογίας, από τον αρχικό κόμβο της (root) έως το σύνολο των τελικών κόμβων της (leaf nodes), το αποτέλεσμα της οποίας είναι μία λίστα με όλα τα πιθανά μονοπάτια της οντολογίας με αριθμημένους τους συμμετέχοντες κόμβους της. Η προτεινόμενη μεθοδολογία έχει σχεδιαστεί και υλοποιηθεί κατά τέτοιο τρόπο ώστε

να επιτρέπει πολλαπλή κληρονομικότητα επί των κόμβων (multiple inheritance), δηλαδή κάθε κόμβος να δύναται να έχει πολλαπλό αριθμό από κόμβους parent, υπό τον εξής ορισμό:

L(i) είναι το επίπεδο του κόμβου i και ισχύει

Li = max(l_1, l_2,…, l_n), όπου,

l_1, l_2,…, l_n είναι η αρίθμηση των επιπέδων στα οποία ο κόμβος i ανήκει ταυτόχρονα.

Για να επιλυθεί το ανωτέρω, κατά την διαδικασία αρίθμησης των κόμβων, οι ακέραιοι οι οποίοι επιλέγονται για να αριθμήσουν τους κόμβους δεν είναι συνεχείς, αλλά προκύπτουν σε συνδυασμό με την ανωτέρω διαδικασία, διατηρώντας έτσι την προϋπόθεση την οποία απαιτεί ο αλγόριθμος Apriori που ακολουθεί αμέσως μετά και η οποία (προϋπόθεση) συνοψίζεται ως εξής:

Για κάθε ζεύγος κόμβων I, J, ισχύει $O_{(J)} > O_{(I)}$, *όπου ο κόμβος J είναι parent node του κόμβου I και* $O_{(N)}$, *συμβολίζει την τάξη (order) του τυχαίου κόμβου N*

Το τρίτο βήμα της μεθοδολογίας, ολοκληρώνεται με τη κλήση αναδρομικών συναρτήσεων (recursive), οι οποίες δημιουργούν την τελική λίστα των paths της οντολογίας.

Βήμα 4ο – Επεξεργασία με τον Αλγόριθμο Apriori – Δημιουργία Κανόνων Συσχέτισης

Κατά το τέταρτο βήμα της μεθοδολογίας ONARM, εκτελείται ο αλγόριθμος Apriori, με σκοπό την εξαγωγή των πιο σημαντικών κανόνων συσχέτισης κόμβων, που χαρακτηρίζουν την υπό εξέταση οντολογία. Για να εξαχθούν ασφαλή συμπεράσματα, ο αλγόριθμος Apriori, ο οποίος παρουσιάστηκε εκτενώς σε προηγούμενο κεφάλαιο, εκτελείται με όλο το φάσμα των δυνατών συνδυασμών των support και confidence παραμέτρων (s,c). Πιο συγκεκριμένα, έχει υλοποιηθεί επί πλέον διαδικασία η οποία εκτελεί αυτόν τον αλγόριθμο για (s,c) = (5%, 5% - 100%, 100%), με βήμα αύξησης 5% και για τα δύο αντίστοιχα. Αυτό κρίθηκε απαραίτητο, έτσι ώστε να ανιχνευθούν με περισσότερη ακρίβεια τα διαστήματα στα οποία η μεθοδολογία δίνει καλύτερα αποτελέσματα κατά τη διαδικασία αξιολόγησής της, σε συνάρτηση με τις παραμέτρους s,c

Το αποτέλεσμα του βήματος αυτού είναι η δημιουργία ενός συνόλου (set), κανόνων, όπως παραδείγματος χάριν οι παρουσιαζόμενοι παρακάτω:

1: { [2, 7], 45, 20}

3: { [1, 7], 30, 70}

7: { [1, 3, 6], 45, 20}

2: { [1, 6, 7], 30, 70}

……

Το αποτέλεσμα είναι ένα set κανόνων συσχέτισης για κάθε συγκεκριμένο ζεύγος (c,s), με την ονομασία R(Ox) οι οποίοι αναφέρονται στην οντολογία Ox

Βήμα 5ο – Αντιστοίχηση Ονοματολογίας Κόμβων (backnaming)

Στα πλαίσια του πέμπτου βήματος της μεθοδολογίας, χρησιμοποιείται το σύνολο των κανόνων R(Ox) από το προηγούμενο βήμα και επανονομάζονται οι κόμβοι με τα αρχικά ονόματά τους αντί της ονοματολογίας που τους δόθηκε προσωρινά, με βάση την αρίθμησή τους. Έτσι ο ανωτέρω πίνακας από το προηγούμενο βήμα μετατρέπεται σε:

A: { [B, G], 45, 20}

C: { [A, G], 30, 70}

G: { [A, C, F], 45, 20}

B: { [A, F, G], 30, 70}

……

Το καινούργιο set κανόνων που προκύπτει για τους κανόνες συσχέτισης της οντολογίας Ox, ονομάζεται R'(Ox).

Τα ανωτέρω βήματα (1-5) εκτελούνται επίσης και για την δεύτερη οντολογία Oy, δίνοντας έτσι ένα αντίστοιχο αποτέλεσμα κανόνων συσχέτισης, R'(Oy).

Βήμα 6ο – Αντιστοίχηση Κόμβων Οντολογιών

Στο έκτο και τελικό βήμα της μεθοδολογίας ONARM, χρησιμοποιούνται τα δύο set κανόνων που προέκυψαν από τα προηγούμενα βήματα, R'(Ox) και R'(Oy). Η μεθοδολογία ONARM έχει σχεδιαστεί έτσι ώστε παραμετρικά, κατά την επιθυμία του χρήστη του συστήματος, να ξεκινά τη σύγκριση από το ίδιο η από διαφορετικό root node κάθε φορά, έτσι ώστε να είναι δυνατή η επικέντρωση σε διαφορετικούς υπογράφους της κάθε οντολογίας κάθε φορά, αυξάνοντας έτσι τη χρηστικότητα και μειώνοντας τον περιττό χρόνο ανίχνευσης των πιθανά σχετιζομένων κόμβων – εννοιών. Για την διαδικασία αντιστοίχησης κόμβων στα πλαίσια του βήματος αυτού, έχουν χρησιμοποιηθεί οι παρακάτω παράμετροι:

1. **S** (support) και **C** (confidence) όπως έχουν οριστεί στο βήμα του αλγόριθμου APRIORI .
2. **α**, ένα όριο (Threshold), τέτοιο ώστε, εάν |Sx –Sy| > THR(α) , τότε K=0
3. **β**, μία παράμετρος, τέτοια ώστε, K ← Average(Sx+Sy) * β

   όπου,

   β > 1 εάν Sx και Sy είναι leaf nodes,

   (ένας ή και οι δύο, στις Ox and Oy αντίστοιχα)

   β = 1 σε κάθε άλλη περίπτωση
4. **γ**, μία παράμετρος, τέτοια ώστε, Kt = K * γ

   όπου,

   γ > 1 εάν N ≡ N', οι δύο κόμβοι έχουν ακριβώς το ίδιο λεκτικό όνομα.

   γ = 1 εάν N, N' αλλιώς, δηλαδή οι κόμβοι N, N' βρέθηκαν να έχουν παραπλήσια αλλά όχι ακριβώς ίδια ονόματα.

Από τα ανωτέρω η ONARM μεθοδολογία δημιουργεί ένα πίνακα συσχέτισης, διαστάσεων [n*m] , που περιλαμβάνει τους κόμβους των δύο οντολογιών Ox και Oy, λαμβάνοντας υπ' όψιν ποια εκ των δύο αναγνωρίζει σαν αναφορά προς σύγκριση (Reference Ontology). Για το σκοπό αυτό από όλα τα πιθανά set που εμπεριέχονται στα R'(Ox) και R'(Oy) εξάγονται συσχετίσεις του τύπου :

R'(Ox) : A → (B, s1, c1), A → (BC, s2, c2), A → (CDF, s3, c3), ….., A → (N1N2..Nn, sj, cj)

Το οποίο μετασχηματίζεται σε:

R'(Ox) : **A**: {(B, s1, c1), (BC, s2, c2), ….., (N1N2..Nn, sj, cj)} και καλείται {**1**}*

Όμοια παίρνουμε και από την δεύτερη οντολογία

R'(Oy) : **A'**: { (B', s1, c1), (B'C', s2, c2), ….., (N1'N2'..N'n, sj, cj) } και καλείται {**2**}*,

όπου A, B, C, D, F και A', B', C', D', F' είναι κόμβοι και (s, c) support και confidence για τον αλγόριθμο Apriori.

Ακολούθως ο ανωτέρω πίνακας συσχέτισης, αποκτά τιμές, σύμφωνα με τον παρακάτω ψευδοκώδικα:

While (i, j) are taking values from {1} and {2}* respectively:*
For every (i ,j) that are columns and rows in the [n x m] matrix
a. Calculate K
b. Calculate Kt
c. Fill cell (i, j) with Kt

d. Reduce considered cases by using constraint of maximal selection of Kt, for subsequence analyses.

Continue until all (i,j) are filled and reduced.

Τα αποτελέσματα που παράγονται από αυτό το βήμα, παρουσιάζονται σε δύο μορφές, (α) με τη μορφή [Ox]: Κόμβος -> [Oy]: Κόμβος, confidence. Επίσης (β) με τη μορφή του προηγούμενου δυσδιάστατου πίνακα και των τιμών του

3.1.3. ΑΞΙΟΛΟΓΗΣΗ - ΑΠΟΔΟΤΙΚΟΤΗΤΑ ΜΕΘΟΔΟΛΟΓΙΑΣ ONARM

Οι πολυπλοκότητες του χρόνου που παρουσιάζονται παρακάτω θεωρήθηκαν για τα χειρότερα σενάρια και ως προς την μεθοδολογία ONARM.

Στην προτεινόμενη τεχνική ONARM, η εξαγωγή των κανόνων συσχέτισης εκτελείται ξεχωριστά για κάθε οντολογία, που είναι O(|G_i| * |C_k|), όπου |C_k| είναι ο αριθμός όλων των υποψήφιων συνόλων στοιχείων που ελέγχονται. Σύμφωνα με το [1], η πολυπλοκότητα της τοποθέτησης συνόλων στοιχείων μεγέθους k είναι O(klog(|G_i|/k)) (ωστόσο σε τυχαίες βάσεις δεδομένων υπάρχουν πολύ λίγα σύνολα στοιχείων). Στην ONARM το μέγιστο μέγεθος είναι d, το βάθος της οντολογίας. Επομένως, στην τελευταία περίπτωση η πολυπλοκότητα είναι $O(\sum_{j=1}^{d} j \log(|Gi|/j))$, που είναι O(d2log(|Gi|). Έτσι η συνολική πολυπλοκότητα της εξαγωγής είναι O(d2log(|Gi|). Ο χρησιμοποιημένος πίνακας ομοιότητας λαμβάνεται στο O(|G1| * | G2|).

3.1.4. ΕΥΘΥΓΡΑΜΜΙΣΗ ΕΝΔΙΑΦΕΡΟΝΤΩΝ ΜΕΡΩΝ ΟΝΤΟΛΟΓΙΩΝ

Η μηχανική των οντολογιών (Ontology Engineering), δηλαδή ο σχεδιασμός, η ανάπτυξη, η συντήρηση και η διαμοίραση των οντολογιών, είναι μια αναπτυσσόμενη διαδικασία της μηχανικής της γνώσης. Επιτρέπει δηλαδή, την οργάνωση της πληροφορίας σε ταξονομίες των εννοιών, που αναπαριστούνται με χαρακτηριστικά (attributes), και σχέσεις μεταξύ εννοιών, οι οποίες αναπαριστούνται με σχέσεις όπως τύπου IS-A, συναρτήσεις, περιορισμούς κλπ. Οι οντολογίες βρίσκουν αποδοχή σε πολλές εφαρμογές, π.χ. στην

ανάκτηση πληροφοριών, στη διαχείριση κειμένων, στην επικοινωνία πρακτόρων λογισμικού, στην οικονομία, και στο ηλεκτρονικό εμπόριο, και μπορούν να σχεδιαστούν και να υλοποιηθούν από διαφορετικές κοινότητες χωρίς την υιοθέτηση κοινών προτύπων για την ανταλλαγή πληροφορίας .

Οι οντολογίες επιβάλλονται από την εκρηκτική ανάπτυξη του Σημασιολογικού Ιστού, όπου χρησιμοποιούνται για να περιγράψουν τη σημασιολογία (semantics) των δεδομένων. Λόγω της αποκεντρωμένης φύσης του Διαδικτύου και του Σημασιολογικού Ιστού, είναι αναπόφευκτο ότι διαφορετικές κοινότητες μέσα στην επονομαζόμενη κοινωνία της πληροφορίας αναπαριστούν και μεταχειρίζονται τις ίδιες βασικές έννοιες με διαφορετικούς τρόπους. Για παράδειγμα, η βασική έννοια «Πρόσωπο» αναλύεται τελείως διαφορετικά σε μια ιατρική οντολογία από μία επιχειρησιακή οντολογία.

3.1.4.1. Το προβλημα της Ευθυγραμμισης Οντολογιων

Η ανάγκη για συγχώνευση / ευθυγράμμιση (merging / aligning), προκύπτει όταν τέτοιες οντολογίες χρειάζεται να ολοκληρωθούν για να διαλειτουργήσουν ως προς την χρηστικότητα των δεδομένων που περιγράφονται σε αυτές. Τα πλεονεκτήματα μιας απλής οντολογίας είναι προφανή (Hovy, E., 1998), δηλαδή βοηθά στην προτυποποίηση της ορολογίας, στην μεταφορά γνώσης, στη διευκόλυνση της διαλειτουργικότητος κ.λ.π. Ας σημειωθεί, ότι η συγχώνευση σκοπεύει στη δημιουργία μιας απλής συνεκτικής που συμπεριλαμβάνει την πληροφορία από όλες τις πηγές ενώ η ευθυγράμμιση (alignment) στο να κάνει τις πηγές συνεπείς και συνεκτικές τη μία με την άλλη αλλά να τις διατηρεί ξεχωριστές.

Σε αυτή την εργασία, θεωρούμε τη διαδικασία της ευθυγράμμισης (alignment) οντολογιών, που μπορεί να χρησιμοποιούν διαφορετικά ή ίδια λεξιλόγια και μπορούν να έχουν επικαλυπτόμενο περιεχόμενο. Πιο ειδικά, ορίζουμε μια νέα παραλλαγή του προβλήματος ευθυγράμμισης (alignment) των οντολογιών: ανανεώνοντας κάθε μία από τις εξεταζόμενες οντολογίες χρησιμοποιώντας τη γνώση που συμπεριλαμβάνεται στην άλλη υπό την προϋπόθεση ότι *η* ανανέωση εκτελείται μόνο για τα μέρη της κάθε οντολογίας που θεωρούνται ενδιαφέροντα για τους χρήστες της.

Αυτό το πρόβλημα είναι σημαντικό για εφαρμογές έξυπνου λογισμικού, που χρησιμοποιούν οντολογίες, όπως για παράδειγμα οι ευφυείς διαμεσολαβητές, στις περιπτώσεις που οποτεδήποτε ένας διαμεσολαβητής χρειάζεται να ανανεώσει τη γνώση του από άλλους και να ανταλλάξει πληροφορία αυτόνομα με αυτούς μέσω μηνυμάτων.

Σε αυτή τη δημοσίευση, ορίζουμε διαφορετικά μέτρα ενδιαφέροντος κομματιών των οντολογιών. Αυτά τα μέτρα αξιοποιούν τη δομή των οντολογιών. Ας σημειωθεί, ότι παρά την υποστήριξη ή την αντιπαράθεση της δήλωσης ότι η αντιστοίχηση / συγχώνευση των οντολογιών είναι παρόμοια με την αντιστοίχηση / ολοκλήρωση του σχήματος μιας βάσης δεδομένων ((Kalfoglou, Y., Schorlemmer, M., 2003), (Noy, N.F., Klein, M., 2004)), η προτεινόμενη οντολογία μπορεί να εφαρμοστεί και στις δύο περιπτώσεις. Στα πλαίσια της παρούσης διατριβής παρουσιάζεται αυτή η μεθοδολογία εξεύρεσης δεικτών ενδιαφέροντος (interestingness), μεταξύ εννοιών των οντολογιών και κατόπιν παρουσιάζεται η πειραματική αξιολόγησή τους

3.1.4.2. Σχετιζομενες Τεχνικες Αντιστοιχησης - Ευθυγραμμισης

Υπάρχουν μερικές τεχνικές που παρουσιάζονται στη βιβλιογραφία για την συγχώνευση και ευθυγράμμιση (alignment) των οντολογιών. Ας σημειωθεί, όμως, ότι όλες μελετούν την συγχώνευση ή την ευθυγράμμιση (alignment) ολόκληρων των εξεταζόμενων οντολογιών. Οι βασικές ιδέες κάποιων από αυτών μπορούσαν να εφαρμοστούν στην ευθυγράμμιση (alignment) ενδιαφερόντων μερών αφού αυτά είχαν εντοπιστεί.

Η μεθοδολογία που παρουσιάζεται στο (Hovy, E., 1998) συγχωνεύει τις αντίστοιχες έννοιες και έπειτα εφαρμόζει ευριστικές μεθόδους επιβεβαίωσης με σκοπό να εμποδίσει και από δομικές και από βασισμένες στο περιεχόμενο ασυνέπειες. Για παράδειγμα, ελέγχει: 1) για νέες υπερ-έννοιες μιας συγχωνευμένης έννοιας στην εξεταζόμενη οντολογία, 2) αν αμοιβαίες απαιτήσεις «εξάρθρωσης» (disjointness) ικανοποιούνται από τις συγχωνευμένες έννοιες, όπως στις αρχικές οντολογίες, 3) αν δημιουργούνται οποιοιδήποτε κύκλοι μετά τη συγχώνευση, 4) αν υπάρχουν οποιοδήποτε παραβιάσεις τυπικών περιορισμών μετά τη συγχώνευση.

Η Chimaera (McGuinness, D.L., Fikes, R., Rice, J., Wilder, S., 2000) είναι ένα εργαλείο που εστιάζει την προσοχή του συντάκτη της οντολογίας σε συγκεκριμένα τμήματα της

οντολογίας που διασυνδέονται σημασιολογικά και πρέπει να διορθωθούν ή να συγχωνευτούν περαιτέρω. Πιο ειδικά, η Chimaera ενώνει σημασιολογικά ίδιους όρους από διαφορετικές οντολογίες, έτσι ώστε αναφέρονται με το ίδιο όνομα στην προκύπτουσα οντολογία, και προσδιορίζει όρους που θα έπρεπε να συσχετιστούν με ένταξη, disjointness, ή σχέσεις instance και να παρέχει υποστήριξη για την εισαγωγή αυτών των σχέσεων. Μετά τη συγχώνευση, η Chimaera υποστηρίζει την αναγνώριση λογικών ασυνεπειών και την πιστοποίηση της προκύπτουσας οντολογίας.

Η τεχνική FCA-Merge (Stumme, G., Maedche, A, 2007) εφαρμόζει τυπική ανάλυση εννοιών (Formal Concept Analysis) για να δημιουργήσει ημι-αυτόματες αντιστοιχήσεις μεταξύ των εξεταζόμενων οντολογιών που έχουν δεδομένα. Βασίζεται σε ένα σύνολο κειμένων φυσικής γλώσσας. Αναπαριστά το σύνολο των κειμένων που καθορίστηκαν σε κάθε μια οντολογία με το μοντέλο «Vector Space», παράγει δύο τυπικές έννοιες από αυτά τα σύνολα και να παράγει ένα περιορισμένο δίκτυο εννοιών. Κάθε τέτοια έννοια είναι υποψήφια ως μια έννοια, μια σχέση, ή μια νέα ένταξη στην συγχωνευμένη οντολογία. Ο χρήστης πρέπει να αλληλεπιδράσει με το σύστημα με σκοπό να επιλύσει πιθανές αντιθέσεις, διπλοεγγραφές, αλλά υπάρχει αυτόματη υποστήριξη από την FCA-Merge με τη χρήση ενός μηχανισμού ερωτήσεων / απαντήσεων που βασίζεται στις ευριστικές μεθόδους. Για παράδειγμα, δύο ή περισσότερες έννοιες των αρχικών οντολογιών παράγουν την ίδια τυπική έννοια. Αυτό δείχνει ότι οι έννοιες θα μπορούσαν να συγχωνευτούν σε μια έννοια στην νέα οντολογία. Ο χρήστης ερωτάται ποια από τα ονόματα να διατηρήσει.

Το εργαλείο PROMPT Noy, N.F., Musen, M., 2000), (καθώς και το PROMPTDIFF Noy, N.F., Musen, M., 2002)) είναι σχεδιασμένο για διατηρεί την εστίαση του χρήστη και να παρέχει ανάδραση στο χρήστη κατά τη διάρκεια της αντιστοίχησης και έπειτα κατά τη διάρκεια της συγχώνευσης και της ευθυγράμμισης (alignment). Πρώτα η PROMPT δημιουργεί μια αρχική λίστα από αντιστοιχήσεις με γλωσσολογική ομοιότητα που βασίζεται στα ονόματα των κλάσεων. Έπειτα, ο χρήστης ξεκινά μια διαδικασία που είτε έχει προταθεί από την PROMPT ή χρησιμοποιώντας ένα περιβάλλον editing για οντολογίες. Τότε η PROMPT εκτελεί την διαδικασία και αυτόματα εκτελεί επιπρόσθετες αλλαγές, παράγει μια λίστα από προτάσεις, και προσδιορίζει αντιθέσεις. Υπάρχουν συγκεκριμένες τέτοιες λειτουργίες για την συγχώνευση / ευθυγράμμιση (alignment). Για παράδειγμα, η λειτουργία «perform a deep copy» περιλαμβάνει την αντιγραφή όλων των parents μιας κλάσης μέχρι την αρχή της

ιεραρχίας και όλες τις κλάσεις και slots που αναφέρεται. Μετά από τέτοιες λειτουργίες, αντιθέσεις που θα μπορούσαν να εμφανιστούν είναι: 1) αντιθέσεις ονόματος (περισσότερα από ένα πλαίσια με το ίδιο όνομα), 2) εξαρτημένες αναφορές (ένα πλαίσιο αναφέρεται σε άλλο πλαίσιο που δεν υπάρχει), 3) πλεονασμό στην ιεραρχία της κλάσης (περισσότερα από ένα μονοπάτια από μια κλάση σε ένα κόμβο-γονέα που δεν είναι η αρχή), κ.λ.π.

Το Ontomorph (Chalupksy, H., 2000) είναι ένα εργαλείο για την υποστήριξη της μετάφρασης συμβολικά αναπαριστώμενης γνώσης σε κάποια διαφορετική μορφή. Βασίζεται σε ένα μείγμα συντακτικών και σημασιολογικών κριτηρίων. Μπορεί, επίσης, να χρησιμοποιηθεί για να υποστηρίξει ενέργειες συγχωνεύσεις που βασίζονται στη γνώση. Αυτό επιτυγχάνεται βρίσκοντας, καταρχήν, σημασιολογικές επικαλύψεις και έπειτα σχεδιάζοντας μετατροπές για να φέρουν τις αρχικές οντολογίες σε αμοιβαία συμφωνία. Τελικά, ελέγχει την προκύπτουσα γνώση για συνέπεια, ομοιομορφία και μη-πλεονασμό. Μετά τον έλεγχο, αν είναι απαραίτητο επαναλαμβάνει τα προηγούμενα βήματα.

Το OntoDNA (Kiu, C.-C., Lee, C.S., 2006) είναι ένα αυτόματο εργαλείο αντιστοίχησης και συγχώνευσης οντολογιών. Χρησιμοποιεί τυπικά ανάλυση εννοιών για να πάρει τις ιδιότητες και τις ενυπάρχουσες σχέσεις ανάμεσα σε οντολογικές έννοιες ετερογενών οντολογιών. Πιο συγκεκριμένα, παράγει δυο τυπικές έννοιες από μια αρχική και μια προκύπτουσα οντολογία. Μετά από μια προ-γλωσσολογική επεξεργασία αυτών των τυπικών εννοιών, εφαρμόζονται συγκεκριμένοι κανόνες αντιστοίχησης για να προσαρμόσουν τους στόχους τους. Έτσι, οι ανακτηθείσες δομές των οντολογικών εννοιών ενεργούν ως background γνώση για να ξεκαθαρίσουν σημασιολογικές ερμηνείες στην επόμενη φάση, όπου χρησιμοποιούνται ομάδες τεχνικών (Self-Organising Map και K-means). Τελικά οι οντολογικές έννοιες της προκύπτουσας οντολογίας ανανεώνονται στην αρχική οντολογία βασιζόμενες σε συγκεκριμένους κανόνες συγχώνευσης.

Στο (Spiliopoulos, V., Valarakos, A.G., Vouros, G.A., Karkaletsis, V., 2007) το πρόβλημα της ευθυγράμμισης (alignment) μειώνεται με την ανακάλυψη των υπαγόμενων σχέσεων ανάμεσα σε στοιχεία της οντολογίας. Η προτεινόμενη τεχνική υπολογίζει τις υπαγόμενες σχέσεις μεταξύ ζευγαριών εννοιών που εξάγονται από ένα εργαλείο αντιστοίχησης. Έτσι, τα ζευγάρια εννοιών αναπαριστούνται ως χαρακτηριστικά διανύσματα μήκους ίσο με τον αριθμό των διακριτών ιδιοτήτων των αρχικών και της τελικής οντολογίας.

Στο (Bock, J., Topor, R., Volz, R, 2007) παρουσιάζεται ένας ημιαυτόματος αλγόριθμος συγχώνευσης, όπου οι χρήστες μπορούν να διαλέξουν κατάλληλα αποτελέσματα από ένα σύνολο προτάσεων. Βασίζεται στο συνδυασμό Answer Set Programming με γλωσσολογική background γνώση. Η τελευταία χρησιμοποιείται για να ανακαλύπτει αντιστοιχίες μεταξύ εννοιών που βασίζονται στην συνωνυμία ενώ η Answer Set Programming υπολογίζει πολλά σύνολα απαντήσεων που παρέχουν επιλογές συγχώνευσης ανάμεσα στις οποίες μπορεί να διαλέξει ο χρήστης. Για να περιοριστεί ο εκθετικός αριθμός των πιθανών λύσεων συγχώνευσης σε ένα λογικό αριθμό, υπάρχουν περιορισμοί, στη μορφή διαισθητικών κανόνων συγχώνευσης, που μπορεί να εφαρμόσει ο χρήστης.

Τελικά, ένα παρόμοιο πρόβλημα είναι εκείνο της ολοκλήρωσης μιας βάσης δεδομένων (π.χ. (Schmitt, I., Saake, G., 1998)). Όμως, οι περισσότερες τεχνικές συγχώνευσης και ολοκλήρωσης σχημάτων βάσεων δεδομένων δεν είναι κατάλληλες για την αντιστοίχηση και συγχώνευση / ευθυγράμμιση (alignment) των οντολογιών για το λόγο ότι δεν χειρίζονται διαφορές στην ονοματολογία, παρουσιάζουν φτωχά αποτελέσματα στην περίπτωση μικρής δομικής ομοιότητας, στην απουσία instances κ.λ.π.

3.1.4.3. Προτεινομενα Μετρα Ενδιαφεροντος στη Ευθυγραμμιση Οντολογιων

Θεωρούμε ότι η αντιστοίχιση των ενδιαφερόντων μερών δυο εξεταζόμενων οντολογιών είναι μια διαδικασία, όπου δημιουργούνται αντιστοιχήσεις μεταξύ στοιχείων των εξεταζόμενων οντολογιών, αντιστοιχίζοντας έννοιες από μια οντολογία αναφοράς σε έννοιες της άλλης οντολογίας (π.χ. (Doan, A., Madhavan, J., Domingos, P., Halevy, A, 2002), (Giunchiglia, F., Shvaiko, P., Yatskevich, M., 2004), (Hovy, E., 1998), (Kalfoglou, Y., Schorlemmer, M., 2002b), (McGuinness, D.L., Fikes, R., Rice, J., Wilder, S., 2000), Noy, N.F., Musen, M., 2000), (Prasad, S., Peng, Y., Finin, T., 2002), (Stumme, G., Maedche, A, 2007), (Tang, J., Li, J., Liang, B., Huang, X., Li, Y., Wang, K., 2006), (Tatsiopoulos C., Boutsinas B., 2009), (Christos Tatsiopoulos, Basilis Boutsinas, Konstantinos Sidiropoulos, 2009)).

Δεδομένης μιας τέτοιας αντιστοίχησης και ενός συνόλου ενδιαφερόντων μερών για κάθε εξεταζόμενη οντολογία, τότε θα μπορούσε να χρησιμοποιηθεί ένας αλγόριθμος ευθυγράμμισης, με σκοπό να κάνει τα ενδιαφέροντα μέρη της οντολογίας O_1 συνεπή και συνεκτικά με την οντολογία αναφοράς O_2. Φυσικά, τέτοια ευθυγράμμιση εφαρμόζεται αν η

O_2 αναπαριστά τουλάχιστον περισσότερη γνώση από την O_1, εφόσον αφορά τα ενδιαφέροντα μέρη της δεύτερης.

Παρακάτω ορίζουμε μερικά μέτρα ενδιαφέροντος (interestingness measures). Όλα βασίζονται στις δομές της οντολογίας. Ας σημειωθεί ότι έχουμε ερευνήσει πολλά άλλα μέτρα, όμως τα ακολούθως προτεινόμενα παρουσιάζουν σημαντικά αποτελέσματα κατά τη διάρκεια εμπειρικών τεστ (Christos Tatsiopoulos, Basilis Boutsinas, Konstantinos Sidiropoulos, 2009):

1. **rel_cD% - Percentage Direct Child Nodes:** ο αριθμός των κόμβων που άμεσα συνδέονται με ένα συγκεκριμένο κόμβο, ως ποσοστό των αριθμών των κόμβων στην οντολογία.
2. **rel_cI% - Percentage Indirect Child Nodes:** αριθμός των κόμβων στον υπογράφο που ξεκινάνε από ένα συγκεκριμένο κόμβο, ως ποσοστό των αριθμών των κόμβων στην οντολογία.
3. **rel_b% - Percentage Brother Nodes:** ο αριθμός των Άμεσων Παιδιών Κόμβων του πατέρα κόμβου (άμεσος πρόγονος) ενός συγκεκριμένου κόμβου, ως ποσοστό των αριθμών των κόμβων στην οντολογία.
4. **mdisbr – Mean Distance of Brother Nodes:** η μέση απόσταση ενός συγκεκριμένου κόμβου από τους αδελφούς του κόμβους. Η απόσταση δύο κόμβων d(x,y) υπολογίζεται χρησιμοποιώντας το μέτρο ανομοιότητας που παρουσιάζεται στο *Error! Reference source not found.*. Η ανομοιότητα ανάμεσα σε οποιεσδήποτε τιμές χαρακτηριστικών αναπαρίσταται από την απόσταση μεταξύ των αντίστοιχων κόμβων της δενδρικής δομής όπως παρουσιάζεται στον παρακάτω τύπο:

$$d(X,Y) = \frac{1}{f_l(X,Y)} xAverage\left(\frac{l(X) - f_l(X,Y)}{\max(p(X))}, \frac{l(Y) - f_l(X,Y)}{\max(p(Y))}\right) x \frac{p(X,Y)}{\max(p(X)) + \max(p(Y))},$$

όπου X και Y αναπαριστούν τους δύο κόμβους, $f_l(X,Y)$ είναι το επίπεδο του πλησιέστερου κοινού κόμβου πατέρα, δηλαδή το επίπεδο του κοντινότερου κοινού προηγούμενου κόμβου, l(X) είναι το επίπεδο του κόμβου X, δηλαδή το βάθος του κόμβου, max(p(X)) είναι το μήκος του μέγιστου μονοπατιού που αρχίζει από την

κορυφή και πάει σε ένα φύλλο και περιέχει τον κόμβο X, p(X,Y) είναι το μήκος του κατευθυνόμενου μονοπατιού (αριθμός ακρών) που περιέχουν το X και το Y. Αν δεν υπάρχει μονοπάτι που ενώνει τα X και Y τότε p(X,Y) = p(X, f_l(X,Y)) + p(Y, f_l(X,Y))). Επίσης p(X,X) = 0. Ας σημειωθεί ότι μικρές τιμές του d(X,Y) αναπαριστούν περισσότερη ομοιότητα. Η μέση απόσταση και οι αδελφοί κόμβοι υπολογίζονται από τον παρακάτω αλγόριθμο:

```
for each Brother Node i of node j
        calculate d(i,j)
        set count+=1, dsum+= d(i,j)
return dsum/count
```

5. **nden(k) – Network Density of range k:** Η πυκνότητα δικτύου εύρους k ενός συγκεκριμένου κόμβου i είναι ο αριθμός των κόμβων που συνδέονται ή μπορεί να προσεγγιστούν από τον i, μέσω ενός μονοπατιού μήκους το μέγιστο k. Ας σημειωθεί ότι έχουμε υλοποιήσει τον υπολογισμό του nden(k) δυναμικά.
6. **in% - Percentage Incoming Paths:** indegree (d_{in}(i)) ενός κόμβου i, δηλαδή ο αριθμός των ακρών που έχουν τον i ως τελικό τους κόμβου, ως ποσοστό των συνολικών εισερχόμενων και εξερχόμενων μονοπατιών.
7. **out% - Percentage Outcoming Paths:** outdegree (d_{out}(i)) ενός κόμβου i, δηλαδή ο αριθμός των ακρών που έχουν τον i ως αρχικό τους κόμβου, ως ποσοστό των συνολικών εισερχόμενων και εξερχόμενων μονοπατιών.
8. **n_l(i)% - Percentage Level Distribution:** Η κατανομή επιπέδου ενός συγκεκριμένου κόμβου i είναι ο αριθμός των κόμβων που ανήκουν στο επίπεδο του κόμβου i ((l(i)), δηλαδή το μήκος του μονοπατιού (αριθμός ακρών) από την αρχή στον κόμβο i, ως ποσοστό των αριθμών των κόμβων στην οντολογία.

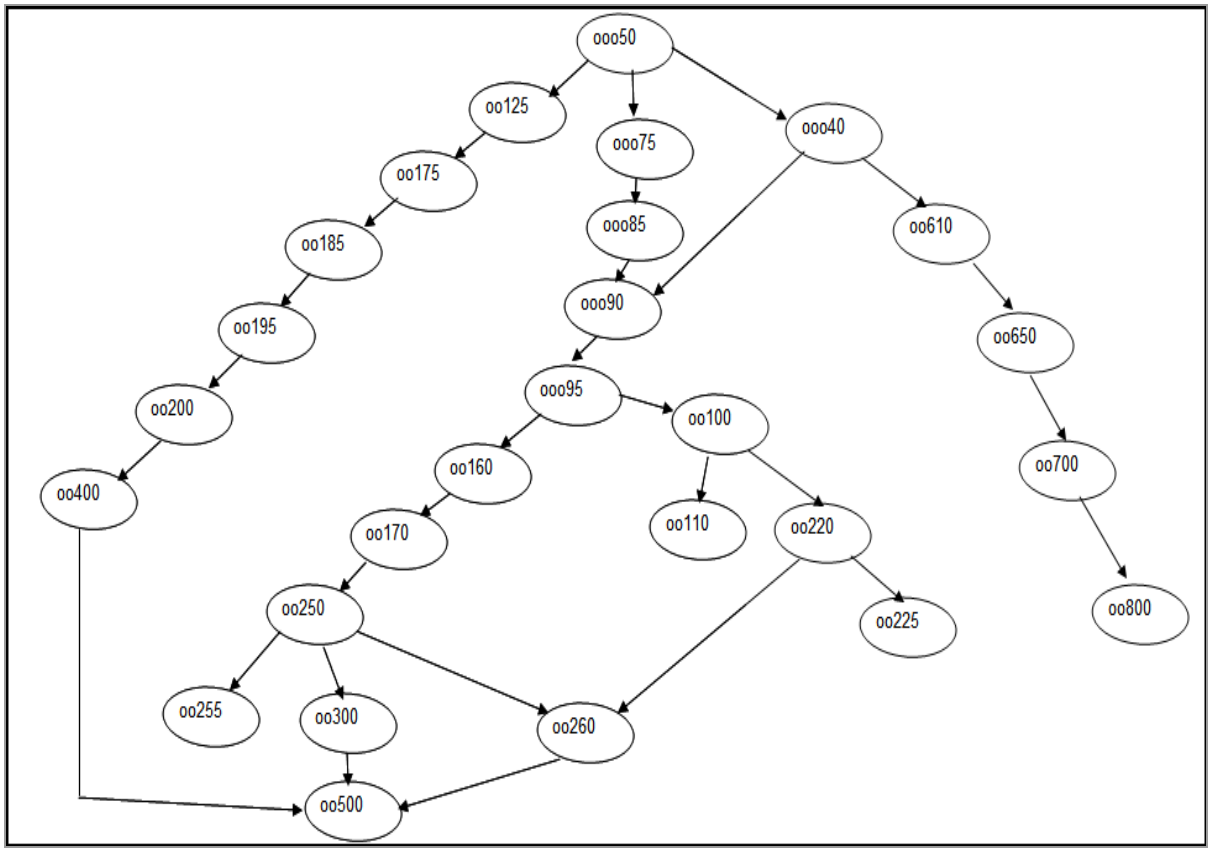
3.2. Πειραματικά Δεδομενα & Αποτελεσματα

3.2.1. Αναγνωριση Σχετιζομενων Κομβων

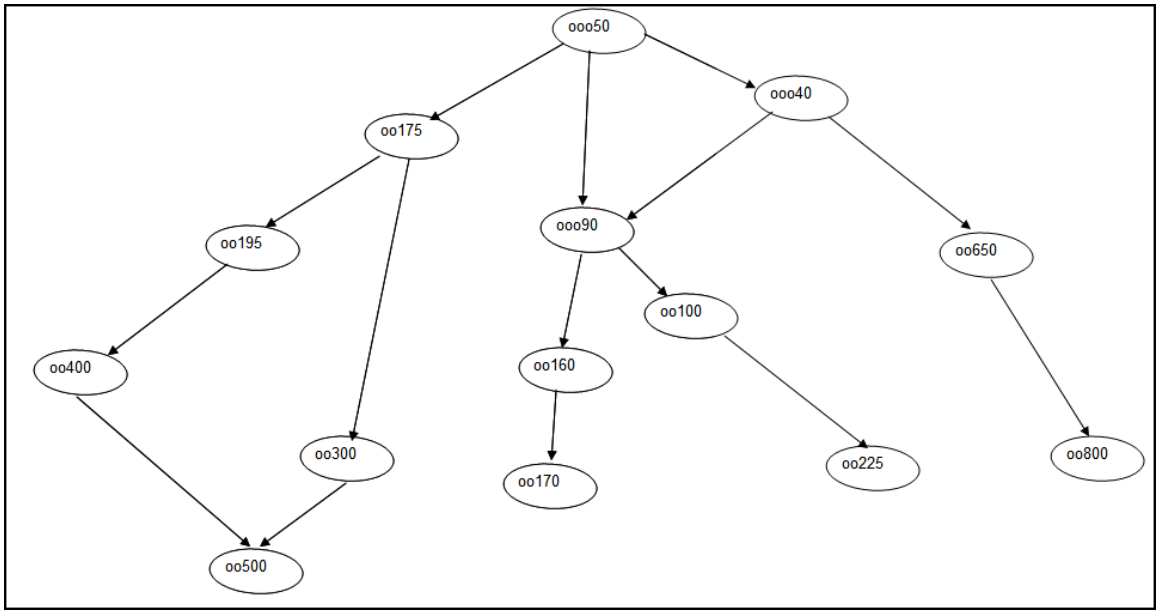
Οι κάτωθι εμπειρικές δοκιμές σκοπεύουν στην εξέταση της ακρίβειας της προτεινόμενης τεχνικής ως προς την αναγνώριση κόμβων - εννοιών. Για τον έλεγχο της ακρίβειας της μεθοδολογίας ONARM, δημιουργήθηκε το παρακάτω πειραματικό πλαίσιο:

Υποθέτουμε την ύπαρξη τριών οντολογιών, O1, O2, O3, έτσι ώστε η O2 να προκύπτει άμεσα από την O1 και η O3 κατ' ευθείαν από την O2. Κατ' αυτόν τον τρόπο η O1 θεωρείται σαν Οντολογία Αναφοράς για τις συγκρίσεις των δύο άλλων. Σαν λεξιλόγιο των κόμβων χρησιμοποιείται αυτό της ευρέως αποδεκτής Gene Ontology.

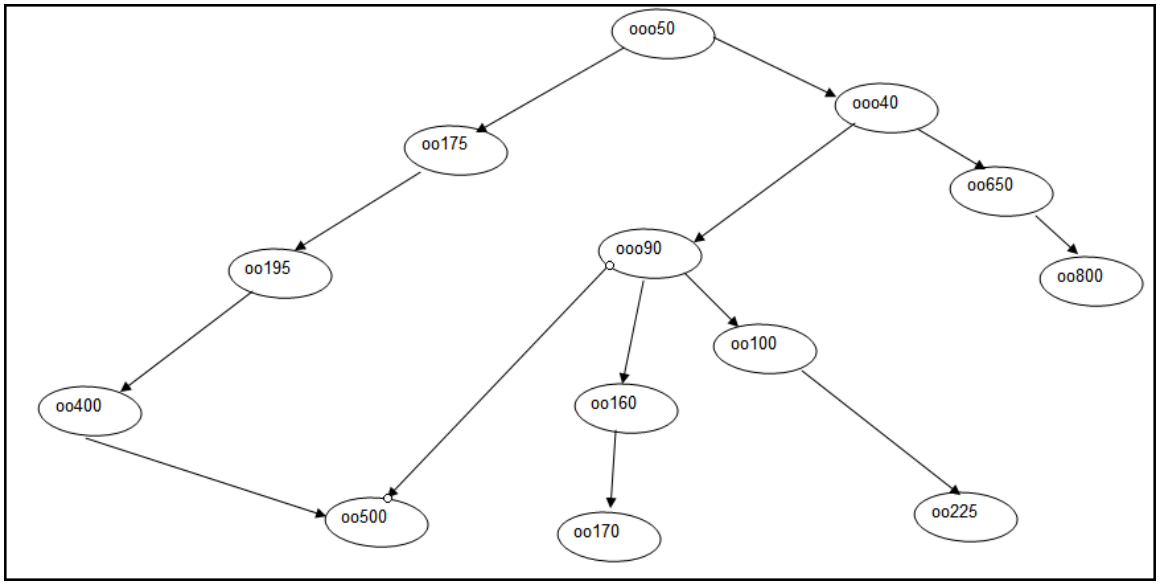
Η Ο1 έχει τη μορφή:



Η Ο2 έχει τη μορφή:



και η O3:



Τα πειράματα που εκτελέστηκαν ήταν συγκρίσεις O2 σε σχέση με O1 και O3 σε σχέση με O1. Σε σχέση με τη σύγκριση και την αντιστοίχηση μεταξύ των οντολογιών O2 και O1, η μεθοδολογία ONARM βρήκε 76 σωστές αντιστοιχίσεις. Οι καλύτερες περιπτώσεις ανακαλύφθηκαν στις περιοχές όπου το support κυμαινόταν μεταξύ 5% - 40% και το confidence μεταξύ 5% - 65%. Παρακάτω παρουσιάζονται σχετικοί πίνακες, σε συνάρτηση με τα s,c. Η ανάλυση για variance και standard deviation, είναι βασισμένη στα scores του Kt, όπως αναλύθηκε στο θεωρητικό υπόβαθρο της μεθοδολογίας ONARM. Στο παρακάτω πίνακα παρουσιάζονται συγκεντρωτικά αποτελέσματα της σύγκρισης O2 και O1 Οντολογιών.

Στους πίνακες αυτούς η στήλη NonZeroVAL δείχνει τις ακριβείς αντιστοιχίσεις ανά περίπτωση:

S	AVG	STDEV	VAR	NonZeroVAL
100	0	0	0	0
95	0	0	0	0
90	0	0	0	0
85	0	0	0	0
80	0	0	0	0
75	0	0	0	0
70	0	0	0	0
65	0	0	0	0
60	0	0	0	0
55	0	0	0	0
50	0	0	0	0
45	0	0	0	0
40	82	69	4,754	13
35	46	50	2,519	10
30	46	50	2,519	10
25	46	50	2,519	10
20	157	174	30,389	20
15	85	125	15,695	7
10	135	204	41,605	10
5	12	30	922	5
			Total:	85

C	AVG	STDEV	VAR	NonZeroVAL
100	0	0	0	0
95	0	0	0	0
90	0	0	0	0
85	0	0	0	0
80	0	0	0	0
75	0	0	0	0
70	0	0	0	0
65	4	17	302	1
60	4	17	302	1
55	4	17	302	1
50	25	51	2643	6
45	24	50	2535	6
40	28	57	3279	6
35	51	75	5682	7
30	57	89	7949	7
25	70	114	12896	8
20	72	119	14051	8
15	83	141	19802	8
10	93	163	26533	8
5	96	163	26539	8
			Total:	75

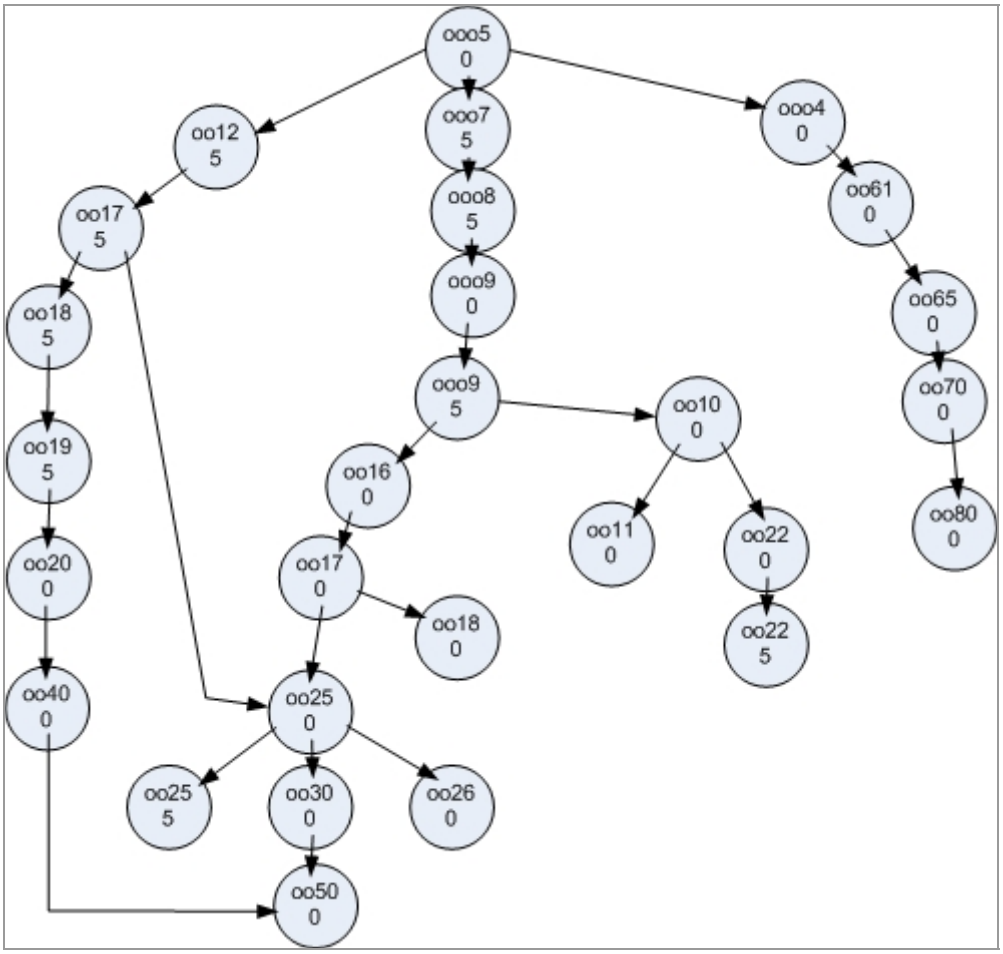
Σχετικά με τις οντολογίες O3 και O1, το ποσοστό ταυτοποίησης είναι 38% ενώ μεταξύ των οντολογιών O2 και O3 είναι 55%. Λόγω του ότι η μεθοδολογία ONARM επιτρέπει την έρευνα και με βάση το λεκτικό κριτήριο, εάν ενεργοποιήσουμε και αυτή τη δυνατότητα, τότε τα αντίστοιχα ποσοστά αυτά αυξάνουν σε 84% με partial matching και 100% λεκτικά.

Παρόμοια, για τις O2 και O1 τα ποσοστά διαμορφώνονται σε 53% και 58% αντίστοιχα.

3.2.2. ΑΥΤΟΜΑΤΗ ΣΥΣΧΕΤΙΣΗ ΣΕ ΣΧΕΣΗ ΜΕ ΑΝΘΡΩΠΙΝΗ ΑΞΙΟΛΟΓΗΣΗ

Τα αποτελέσματα του εμπειρικού ελέγχου στην αξιολόγηση των προσδιορισμένων μέτρων ενδιαφερόντων σε μια εξεταζόμενη οντολογία G_1 παρουσιάζονται στο παρακάτω σχήμα, όπως προέκυψε από εμπειρικές δοκιμές στα πλαίσια της παρούσης Δ.Δ.

Κατασκευάζεται σύμφωνα με την Gene Ontology (http://www.geneontology.org) . Η ονοματολογία των κόμβων είναι συμβατή με την Gene Ontology, έτσι ώστε η πλήρης δομή της Gene Ontology να χρησιμοποιηθεί σε μελλοντικά τεστ αξιολόγησης.



Σχήμα : η εξεταζόμενη οντολογία G1

Οι τιμές για όλα τα παραπάνω μέτρα μαζί με την αξιολόγηση του ειδικού (το 1 δίνεται στον υψηλότερου ενδιαφέροντος κόμβο) παρουσιάζονται στο σχήμα 2. Για την αξιολόγηση των συγκεκριμένων μέτρων, χρησιμοποιήθηκε μια βαθμολογία των κόμβων, η οποία έγινε από άνθρωπο ειδικό στις οντολογίες και στο domain. Για παράδειγμα οι κόμβοι oo100, oo95 και

οο250 είναι μεγάλου ενδιαφέροντος για τον ειδικό, με τον οο100 να είναι ο πιο ενδιαφέρον και έπειτα το οοο95 και ο οο250.

G1	Node	rel_cD%	rel_cl%	rel_b%	mdisbr	nden [2]	in%	out%	n_l(i)%	H
1	οοο50	10.71	96.43	0	-	7	0	100	0.0%	0
2	οοο95	7.14	42.86	0	-	8	33.33	66.67	0.0%	1
3	οο100	7.14	10.71	3.57	0.0025	5	33.33	66.67	0.0%	1
4	οο125	3.57	35.71	7.14	0.01	4	50	50	0.0%	2
5	οο250	10.71	14.29	7.14	0.0059	8	40	60	0.0%	2
6	οο170	7.14	21.43	0	-	7	33.33	66.67	0.0%	4
7	οο175	7.14	32.14	0	-	8	33.33	66.67	0.0%	5
8	οο160	3.57	25	3.57	0.0025	5	50	50	0.0%	5
9	οοο90	3.57	46.43	3.57	0.0115	7	66.67	33.33	0.0%	6
10	οοο40	7.14	64.29	7.14	0.01	5	33.33	66.67	0.0%	7
11	οο185	3.57	14.29	3.57	0.01	4	50	50	0.0%	7
12	οοο85	3.57	50	0	-	4	50	50	0.0%	7
13	οο400	3.57	3.57	0	-	3	50	50	0.0%	8
14	οο195	3.57	10.71	0	-	4	50	50	0.0%	8
15	οοο75	3.57	53.57	7.14	0.01	3	50	50	0.0%	8
16	οο200	3.57	7.14	0	-	4	50	50	0.0%	8
17	οο610	3.57	10.71	3.57	0.0115	4	50	50	0.0%	8
18	οο300	3.57	3.57	7.14	0.0014	4	50	50	0.0%	9
19	οο180	0	0	3.57	0.0018	2	100	0	0.0%	10
20	οο260	0	0	7.14	0.0015	3	100	0	0.0%	10
21	οο650	3.57	7.14	0	-	4	50	50	0.0%	10
22	οο110	0	0	3.57	0.003	2	100	0	0.0%	10
23	οο500	0	0	0	-	4	100	0	0.0%	10
24	οο255	0	0	7.14	0.0015	3	100	0	0.0%	10
25	οο700	3.57	3.57	0	-	3	50	50	0.0%	10
26	οο225	0	0	0	-	2	100	0	0.0%	10
27	οο800	0	0	0	-	2	100	0	0.0%	10
28	οο220	3.57	3.57	3.57	0.003	3	50	50	0.0%	10

Ανωτέρω, πολλά μέτρα δείχνουν ενδιαφέρον με ένα τρόπο που αντανακλά τις πρώτες επιλογές του ειδικού. Επιπλέον, μερικά μέτρα ανακαλύπτουν επιπλέον ενδιαφέροντες κόμβους. Για παράδειγμα, το μέτρο n_l(i)% και το μέτρο out% προσδιορίζουν και τον οο100 και τον οο170 ως ενδιαφέροντες, δίνοντας τις τιμές 14.3% και 66.67% αντίστοιχα. Αυτό το αποτέλεσμα προτάθηκε στον ειδικό για περισσότερα σχόλια, παρ' όλο που ο ίδιος δεν το είχε εντοπίσει. Ο οποίος, το θεώρησε αξιόπιστο σύμφωνα με τα κριτήρια του.

Στα πλαίσια των ανωτέρω, ορίστηκε κατ' αυτόν τον τρόπο, μια νέα παραλλαγή του προβλήματος της ευθυγράμμισης: το πρόβλημα της ευθυγράμμισης μόνο ενδιαφερόντων μερών της οντολογίας. Επίσης ορίστηκε και αξιολογήθηκε ένας αριθμό διαφορετικών μέτρων ενδιαφέροντος (interestingness) μερών των οντολογιών.

Η εφαρμογή τους εντόπισε επιλογές ομοιότητος εννοιών παρόμοιες με αυτές του ανθρώπου ειδικού (expert) με μεγάλη αξιοπιστία.

Επιπρόσθετα, μετά τον εντοπισμό των ενδιαφερόντων μερών, μπορεί να εφαρμοστεί οποιοσδήποτε αλγόριθμος ευθυγράμμισης που προτείνεται στη βιβλιογραφία.

Τα παραπάνω αποσκοπούν σε μια ολοκλήρωση διαφορετικών ζυγισμένων μέτρων, με την ολοκλήρωσή τους σε ένα ενοποιημένο συναρτησιακό μοντέλο ως αποτέλεσμα αυτών: $f(w_1 * rel_cD, \ldots, w_8 * n_I(i))$, όπου το w_i είναι ένα αντίστοιχο βάρος.

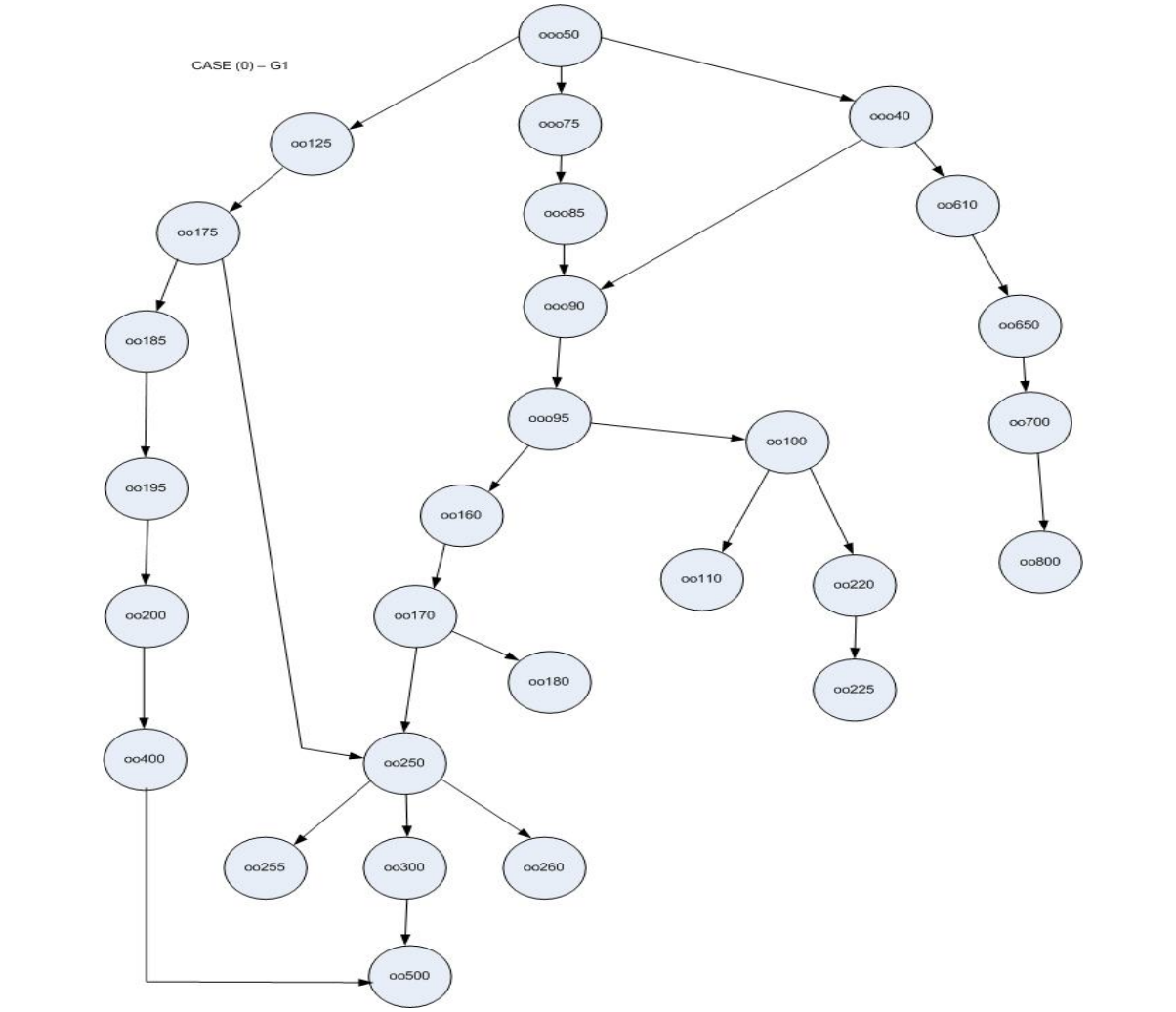
Επιπλέον, η μεθοδολογία λαμβάνει υπόψη την γλωσσολογική ανάλυση των εννοιών που απεικονίζονται από κόμβους σύμφωνα με μια συλλογή κειμένων ή του WWW.

3.2.3. Συμπεριφορα Βαρων Σε Διαφορετικου Τυπου Οντολογιες

Ακολούθως, έγινε πειραματική ανάλυση για την απάντηση στο ζήτημα της συμπεριφοράς των διαφόρων μέτρων, σε οντολογίες γράφους που προέρχονται από κοινό αρχικό λεξιλόγιο, αλλά με πολύ διαφορετική κατασκευαστική δομή και την επιρροή των βαρών αυτών στην τελειοποίηση της διαδικασίας (optimization).

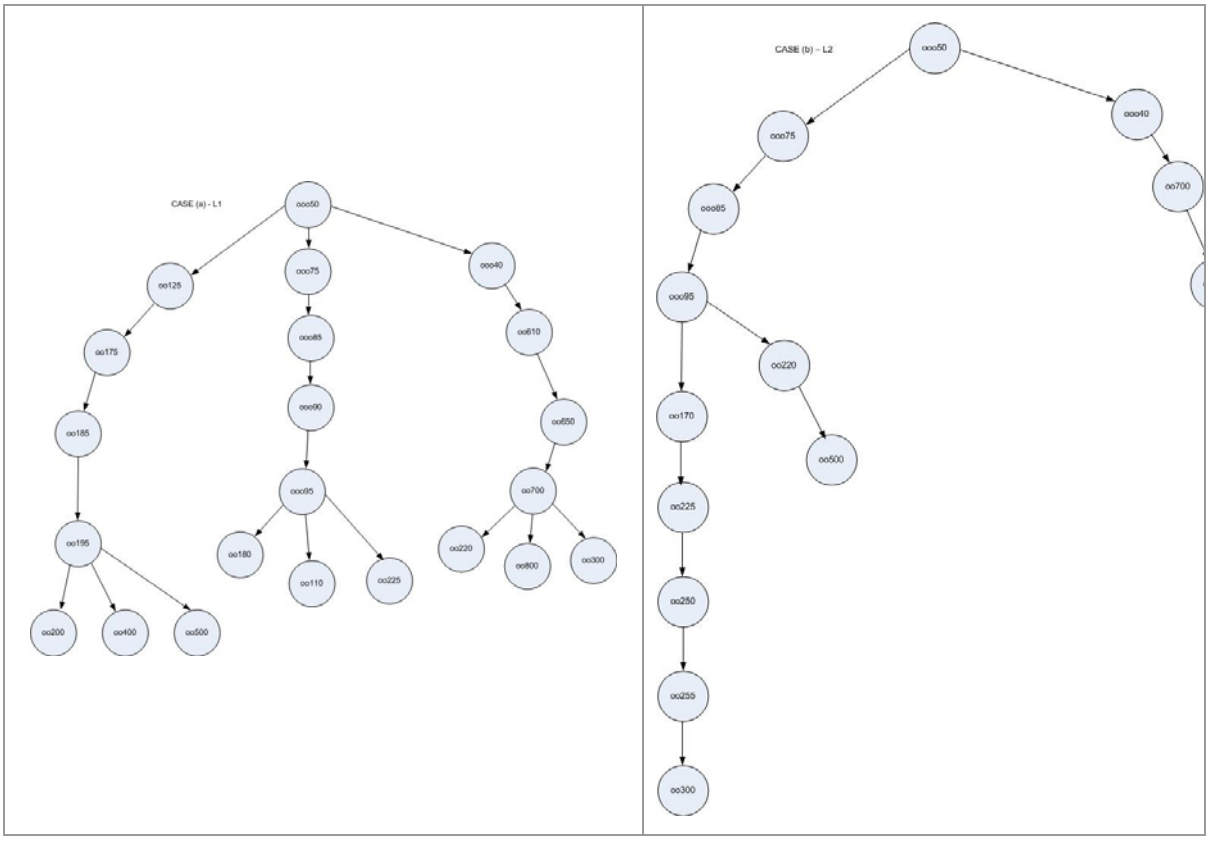
Χρησιμοποιήθηκαν για αυτό το σκοπό δομές και έννοιες από την αρχική οντολογία Gene Ontology και διατηρήθηκε η αρχική ονοματολογία της.

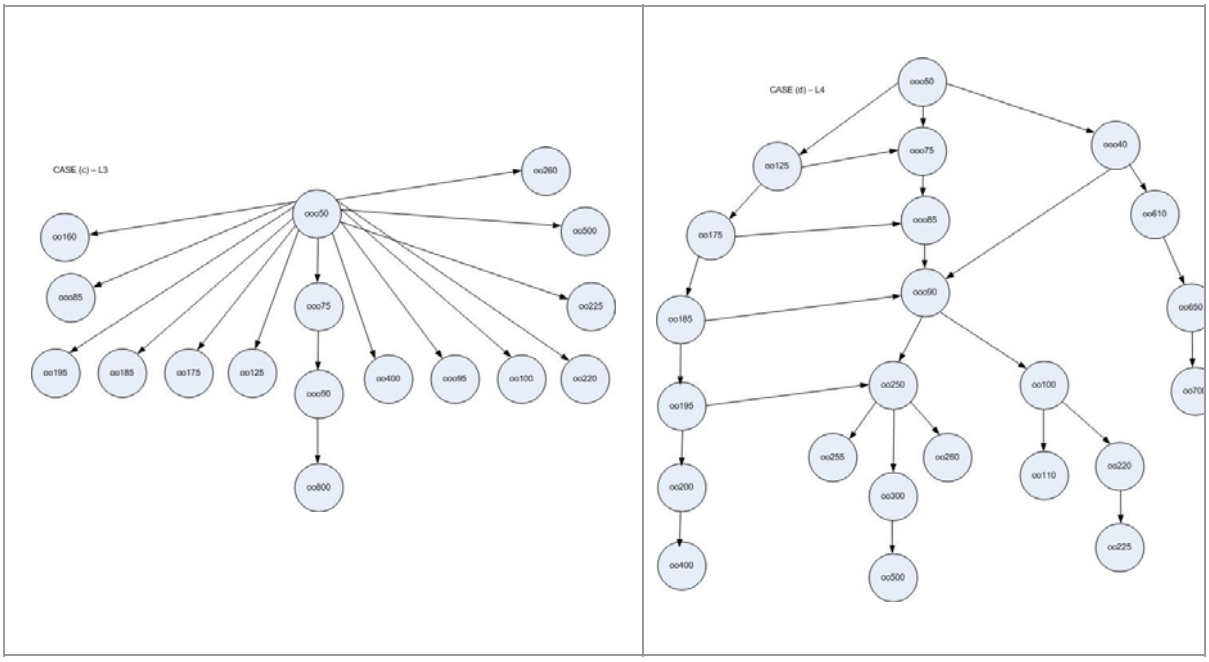
G1 – Αρχική Οντολογία



Από την ανωτέρω αυτή οντολογία, αποκτώνται τέσσερις επί μέρους, με σκοπό τη δημιουργία ακραίων καταστάσεων – γράφων έτσι ώστε να παρατηρηθεί ασφαλέστερα η συμπεριφορά των μέτρων ενδιαφέροντος (interestingness measures) σε όλου τους τύπους των γράφων.

Ακολουθούν τέσσερις οντολογίες L1, L2, L3 και L4. Το κοινό χαρακτηριστικό τους είναι ότι προέρχονται από την ίδια οντολογία G1 που έχει ονομαστεί με τα πρότυπα της gene ontology. Παρατηρούμε ότι στην περίπτωση της L1 αυτή αναπαρίσταται σαν μία ισορροπημένη δενδρική δομή και ως προς το βάθος των επιπέδων της, αλλά και ως προς την οριζόντια διάταξή τους.





Ακολούθως η οντολογία L2, έχει σχεδιαστεί να έχει το κυριότερο βάθος πάνω σε ένα κύριο μονοπάτι, ουσιαστικά δημιουργώντας μία κεκλιμένη δενδρική δομή, μη ισορροπημένη.

Στη συνέχεια, η γραφική αναπαράσταση της οντολογίας L3 αναπαρίσταται σαν ένα δίκτυο τύπου star, όπου υπάρχει ένας κεντρικός κόμβος και κατόπιν οι περισσότεροι υπόλοιποι κόμβοι της αποτελούν κατ' ευθείαν direct children nodes, προσδίδοντας έτσι την ονομασία της «αβαθούς» οντολογίας σε αυτή.

Τέλος, το χαρακτηριστικό της τέταρτης οντολογίας είναι η ύπαρξη πολλαπλής κληρονομικότητας καθώς και η αρκετά ισοβαρής κατανομή των εννοιών κόμβων σε αυτή.

Για το σκοπό αυτό βαθμολογήθηκαν από expert οι κόμβοι των ανωτέρω οντολογιών έπειτα από παρατήρηση με βάση τα δομικά τους χαρακτηριστικά.

L1	Node	rel_cD%	rel_cl%	rel_b%	mdisbr	nden [2]	snk%	src%	n_l%	H
	ooo50	13.64	95.45	0	0	6	0	100	4.55	0
	oo195	13.64	13.64	0	0	5	25	75	13.64	1
	oo700	13.64	13.64	0	0	5	25	75	13.64	1
	ooo95	13.64	13.64	0	0	5	25	75	13.64	1
	oo125	4.55	27.27	9.09	0.0278	3	50	50	13.64	2
	oo175	4.55	22.73	0	0	4	50	50	13.64	2
	oo185	4.55	18.18	0	0	6	50	50	13.64	2
	oo610	4.55	22.73	0	0	4	50	50	13.64	2
	oo650	4.55	18.18	0	0	6	50	50	13.64	2
	ooo40	4.55	27.27	9.09	0.0278	3	50	50	13.64	2
	ooo75	4.55	27.27	9.09	0.0278	3	50	50	13.64	2
	ooo85	4.55	22.73	0	0	4	50	50	13.64	2
	ooo90	4.55	18.18	0	0	6	50	50	13.64	2
	oo110	0	0	9.09	0.0056	2	100	0	40.91	3
	oo180	0	0	9.09	0.0056	2	100	0	40.91	3
	oo200	0	0	9.09	0.0056	2	100	0	40.91	3
	oo220	0	0	9.09	0.0056	2	100	0	40.91	3
	oo225	0	0	9.09	0.0056	2	100	0	40.91	3
	oo300	0	0	9.09	0.0056	2	100	0	40.91	3
	oo400	0	0	9.09	0.0056	2	100	0	40.91	3
	oo500	0	0	9.09	0.0056	2	100	0	40.91	3
	oo800	0	0	9.09	0.0056	2	100	0	40.91	3

Στην τελευταία στήλη του ανωτέρω πίνακα εμφανίζεται η βαθμονόμηση και η κατηγοριοποίηση των κόμβων της οντολογίας L1. Στις υπόλοιπες στήλες εμφανίζεται και η ο υπολογισμός του κάθε μέτρου interestingness, όπως προαναφέρθηκε.

Παρόμοια, σχηματίστηκαν και οι αντίστοιχοι πίνακες για τις οντολογίες L2, L3 και L4.

Κατόπιν κλήθηκε το σύστημα να κάνει συνεχώς βαθμονόμηση (weighting) των δεικτών ενδιαφέροντος που προτάθηκαν στα πλαίσια της τεχνικής ONARM. Σκοπός αυτού ήταν να παρατηρηθούν τα όρια μέσα στα οποία κινούνται τα βάρη αυτά, προκειμένου συνδυαστικά να αποδίδουν αποτελέσματα τα οποία είναι κοντά στις επιλογές του human expert.

Επίσης σκοπός ήταν να παρατηρηθεί και η συμπεριφορά αυτή σε σχέση με την γεωγραφική κατανομή των κόμβων της οντολογίας και τι είδους αντιπροσωπευτικό γράφο αναπαριστούσε.

Δεδομένου ότι υπάρχουν 8 ανεξάρτητοι δείκτες ενδιαφέροντος, ανεδείχθησαν 8 διαφορετικά βάρη αντίστοιχα, w1 – w8. Τα βάρη αυτά καθ' όλη τη διάρκεια της πειραματικής διαδικασίας ήταν διαρκώ μεταβαλλόμενα, έτσι ώστε να προκύψαν όλοι οι δυνατοί συνδυασμοί τους από το 0% -!00%, αυξανομένων ανά 5% και 10%.

Εξετάζουμε τις ακόλουθες L1, L2, L3, L4 περιπτώσεις οντολογιών ως ακολούθως:

Η περίπτωση L1 είναι αυτή που διαθέτει ισορροπημένη μορφή και συμμετρία.

Η περίπτωση L2 είναι αυτή που διαθέτει μη σταθμισμένη (unbalanced) μορφή

Η περίπτωση L3 είναι αυτή που διαθέτει μία τοπολογία χωρίς βάθος αλλά είναι απλωμένη σε ένα πρώτο επίπεδο.

Η περίπτωση L4 είναι αυτή η οποία διαθέτει και τους πιο πολλούς κύκλους.

Τα υπό μελέτη μέτρα ενδιαφέροντος είναι:

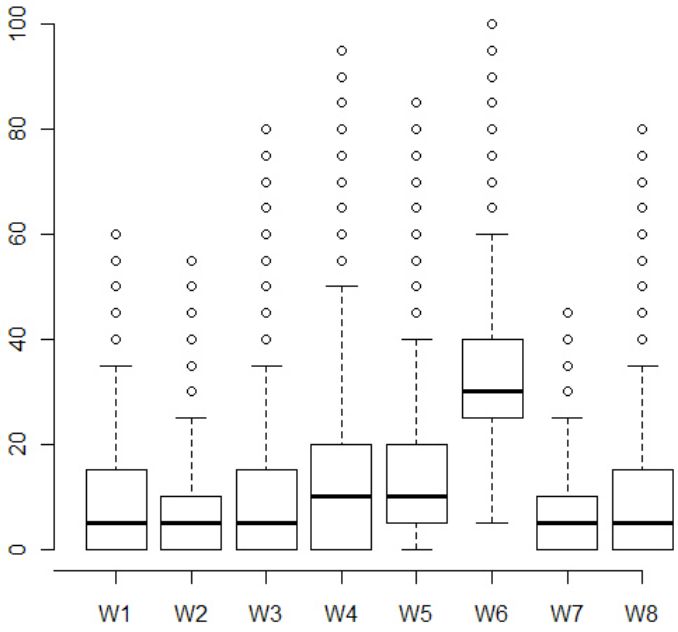
- rel_CD% με αντίστοιχο βάρος w1,
- rel_CI% με αντίστοιχο βάρος w2,
- rel_b% με αντίστοιχο βάρος w3,
- mdistr με αντίστοιχο βάρος w4,
- nden[2] με αντίστοιχο βάρος w5,
- snk% με αντίστοιχο βάρος w6,
- src% με αντίστοιχο βάρος w7,
- n_l(i)% με αντίστοιχο βάρος w8,

Τα αποτελέσματα ανάλυσης των ανωτέρω βαρών w1….w8 αντίστοιχα, έχουν ως ακολούθως:

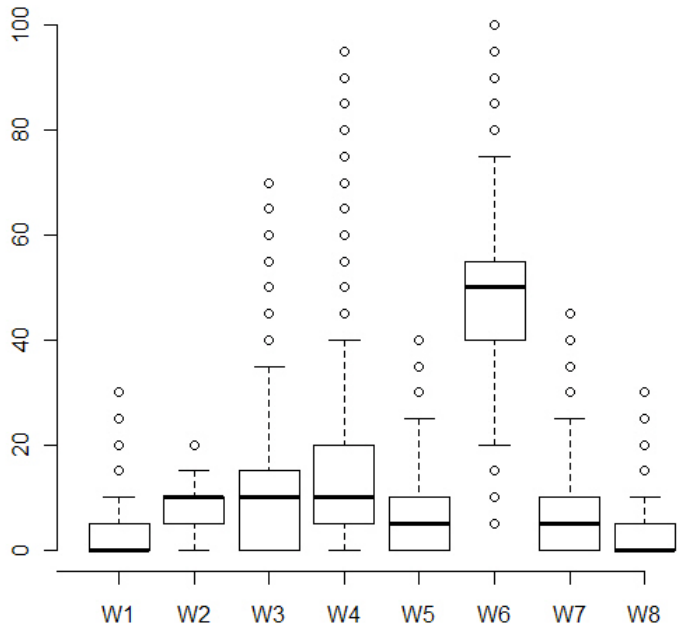
- Στην περίπτωση L1 αναγνωρίστηκαν 21 από τους 22 κόμβους, δηλαδή επιτυχία 95% περίπου. Στην περίπτωση αυτή, παρατηρούμε ότι οι μέσοι όροι των βαρών των μέτρων κυμαίνονται περίπου στα ίδια επίπεδα, εκτός αυτού του w6 ο οποίος βρίσκεται αρκετά ψηλότερα. Ουσιαστικά, στην περίπτωση του γράφου L1, στην ορθή αναγνώριση των 21 από τους 22 κόμβους συμμετέχουν αντίστοιχα και τα οκτώ υπό μελέτη μέτρα.
- Στην περίπτωση L2 αναγνωρίστηκαν 13 από τους 14 κόμβους, δηλαδή επιτυχία 93% περίπου. Παρατηρούμε στην περίπτωση αυτή, ότι οι τιμές του βάρους w6 παρουσιάζουν πολύ υψηλότερο μέσο όρο από τη μέση διακύμανση των μέσων όρων των υπολοίπων βαρών. Επίσης παρατηρούμε ότι η βαρύτητα των w1, w8 τείνει να μηδενισθεί δεδομένου ότι ο μέσος όρος του είναι πολύ κοντά στο 0 και άρα και η συμμετοχή των αντιστοίχων μέτρων στον υπολογισμό της κατηγοριοποιήσεως.

- Στην περίπτωση L3 αναγνωρίστηκαν 16 από τους 17 κόμβους, δηλαδή επιτυχία 94% περίπου. Στην περίπτωση αυτή, παρατηρούμε ότι τα βάρη w3 και w8 ουσιαστικά μηδενίζονται και κατ' αυτόν τον τρόπο τα αντίστοιχα μέτρα δεν συμμετέχουν ουσιαστικά στη διαδικασία. Παρατηρούμε επίσης ότι και πάλι το w6 έχει εύρος διακύμανσης και μέσο όρο αντίστοιχα πολύ υψηλότερα από αυτά των υπολοίπων βαρών.
- Στην περίπτωση L4 αναγνωρίστηκαν 9 από τους 23 κόμβους, δηλαδή ποσοστό 40% περίπου.

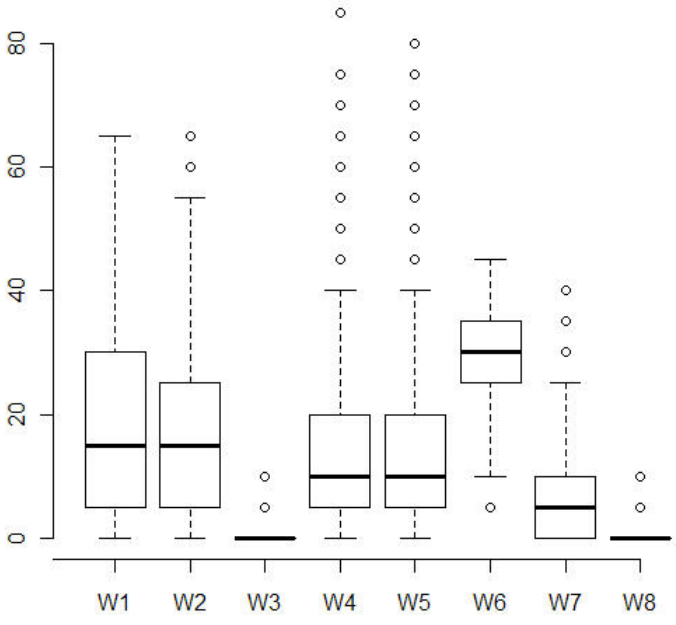
Διαγραμματικά, παρατηρούμε από τα ακόλουθα γραφήματα την συμπεριφορά των βαρώ w1, w2, w3, w4, w5, w6, w7 και w8 σχετικά με την ανίχνευση των ομοίων κόμβων.:



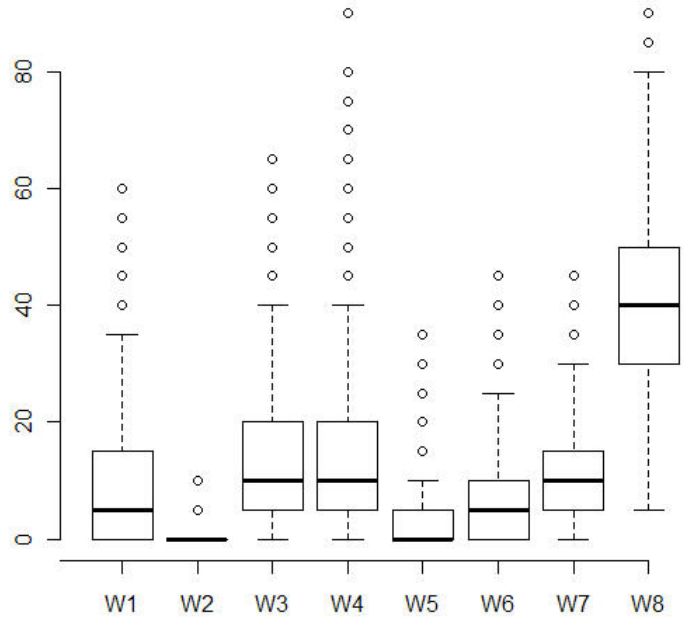
L1


0
20
40
60
80
100
W1
W2
W3
W4
W5
W6
W7
W8

L2


0
20
40
60
80
W1
W2
W3
W4
W5
W6
W7
W8

L3



L4

Δείγμα από το προς στατιστική επεξεργασία για τα ανωτέρω αποτελέσματα είναι το ακόλουθο:

1	W2	W3	W4	W5	W6	W7	W8
0	0	0	0	0	20	0	80
0	0	0	0	0	20	5	75
0	0	0	0	0	25	0	75
0	0	0	0	0	25	5	70
0	0	0	0	0	25	10	65
0	0	0	0	0	30	0	70
0	0	0	0	0	30	5	65
0	0	0	0	0	30	10	60
0	0	0	0	0	30	15	55
0	0	0	0	0	30	20	50
0	0	0	0	0	35	0	65
0	0	0	0	0	35	5	60
0	0	0	0	0	35	10	55
0	0	0	0	0	35	15	50
0	0	0	0	0	35	20	45
0	0	0	0	0	35	25	40

0	0	0	0	0	40	0	60
0	0	0	0	0	40	5	55
0	0	0	0	0	40	10	50
…	……	…	……	……	……..	…..	……

Τα 4 αυτά αρχεία περιέχουν δεδομένα της τάξεως των 140.000 γραμμών, δομημένων όπως παραπάνω τα οποία επεξεργάστηκαν με το στατιστικό εργαλείο R.

3.3. ΣΥΜΠΕΡΑΣΜΑΤΑ

Η ONARM, αξιοποιεί μέτρα δομικής ομοιότητας με σκοπό αυτόματα να προσδιορίσει την αντιστοίχηση και την πιθανή ενοποίηση σε δύο εξεταζόμενες οντολογίες και την πρόταση συσχετιζομένων εννοιών αυτών. Η ONARM βασίζεται στη δομή των οντολογιών αλλά και σε γλωσσικά χαρακτηριστικά τους, μπορεί δε να χειριστεί και τα μεταδεδομένα και την ετερογένεια της κάθε περίπτωσης της κάθε οντολογίας.

Η ONARM μπορεί εύκολα να συμπεριληφθεί σε συστήματα που βασίζονται σε συνδυασμό τεχνικών αντιστοίχησης, ειδικά διότι υπάρχουν μόνο λίγες τεχνικές για τη δομική ομοιότητα ((McGuinness, D.L., Fikes, R., Rice, J., Wilder, S., 2000), Noy, N.F., Musen, M., 2000), (Tatsiopoulos C., Boutsinas B., 2009)), ιδιαίτερα χωρίς ανθρώπινη παρέμβαση.

Τέλος, η ONARM παρουσιάζει μια πολυπλοκότητα χαμηλού χρόνου σε σχέση με τις σχετιζόμενες προσεγγίσεις που προαναφέρθηκαν.

Κεφάλαιο 4. Μελέτη Εφαρμογής της Προτεινόμενης Μεθοδολογίας στον Τουρισμό

4.1. Παρουσιαση Εφαρμογης

Η εφαρμογή Concept-Net επιτρέπει μέσω των προσωπικών φορητών συσκευών (η ύπαρξη και χρήση των οποίων σήμερα είναι ευρέως διαδεδομένη), να απεικονίζουν τα ενδιαφέροντά τους περί τον τουρισμό σε επίπεδο τοπικό, στη συσκευή τους, στο προσωπικό κινητό τηλέφωνο λόγου χάριν ή σε άλλες παρόμοιες συσκευές. Βέβαια, οποιοδήποτε πεδίο εφαρμογής επιλεγεί, αυτό μπορεί να υποστηριχθεί από το σύστημα Concept-Net.

4.1.1. Σκοπος - Γενικη περιγραφη

Για να υλοποιηθεί η ολοκληρωμένη τόσο σε αλγοριθμικό όσο και σε μεθοδολογικό επίπεδο ONARM προσέγγιση, αναπτύχθηκε η εφαρμογή εξαγωγής και διαχείρισης γνώσης Concept-Net. Ο σκοπός της Concept-Net είναι να αποδείξει μέσω υλοποιημένου λογισμικού, σε πραγματικό χρόνο, την εφαρμοσιμότητα και τη λειτουργικότητα της προτεινόμενης μεθοδολογίας ONARM, στον τομέα του τουρισμού (Tatsiopoulos C., Boutsinas B., submitted).

Για το σκοπό αυτό και προκειμένου να γίνει γρήγορα και εύληπτα αποδεκτή η εφαρμογή, με τελικό στόχο την πραγματική αγορά των κινητών συσκευών, υλοποιήθηκε λαμβάνοντας υπ' όψιν έννοιες και γνώση που αφορούν το τουριστικό περιεχόμενο και έννοιες που περιβάλλουν και περιγράφουν αυτό.

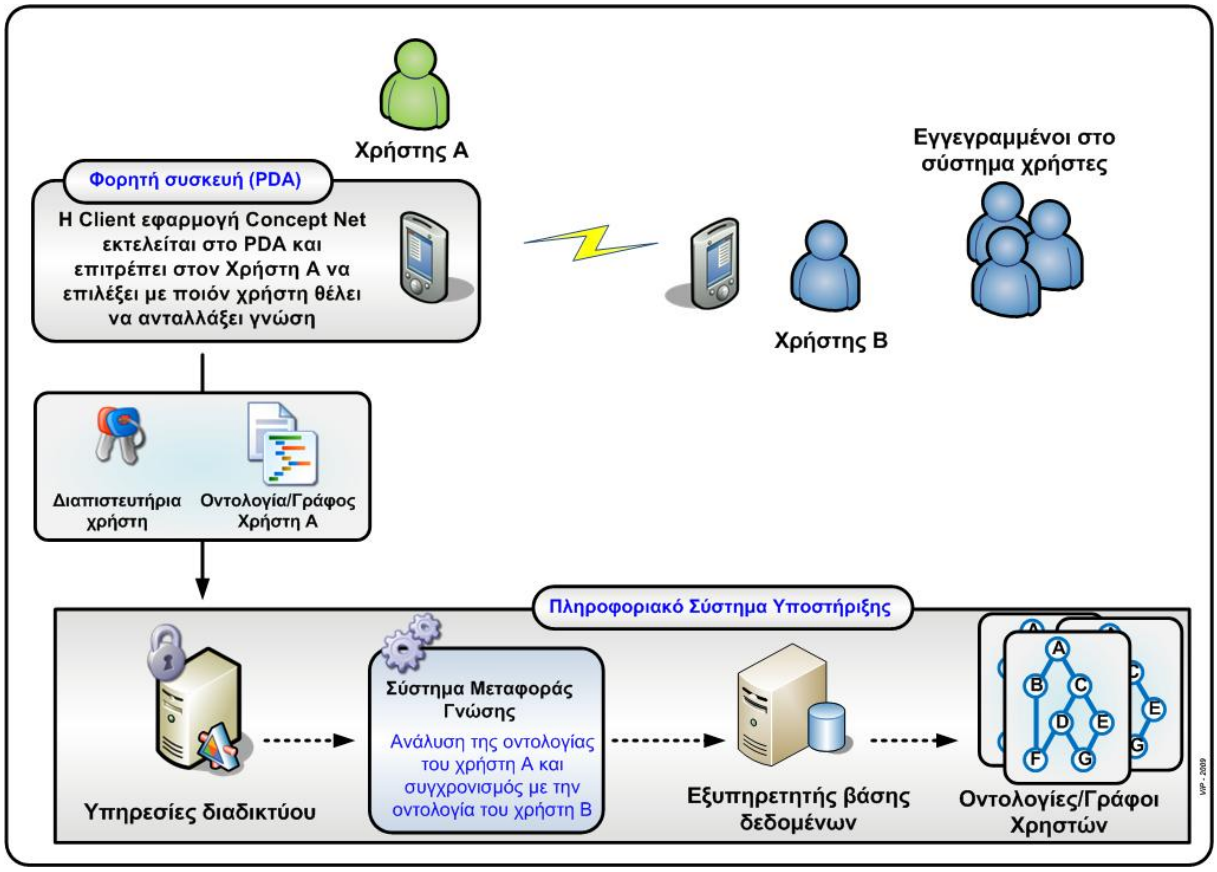
Ως προς τη χρήση του, αν θεωρήσουμε γενικά δύο τέτοιους χρήστες Α και Β, οι οποίοι έχουν περιγράψει γραφικά, με συμβολικό τρόπο, τις έννοιες τους στις φορητές συσκευές τους, τότε αυτοί, είναι σε θέση να συνδέονται στο σύστημα, να μεταφέρονται οι οντολογίες τους στο κεντρικό σύστημα και εκεί να επεξεργάζονται.

Κατ' αυτόν τον τρόπο αναλύονται οι γράφοι-οντολογίες των χρηστών Α και Β και ακολούθως συγχρονίζονται σε κοινά σημεία-έννοιες ενδιαφέροντος. Οι αρχικές οντολογίες των χρηστών Α και Β στη συνέχεια διαφοροποιούνται, με βάση την επεξεργασία του προηγούμενου βήματος και το διαμορφωμένο αντίγραφό τους αποθηκεύεται στη βάση δεδομένων του συστήματος σε τόσα διαμορφωμένα αντίγραφα, όσα επιθυμεί ο εκάστοτε

χρήστης Α και Β. Ακολούθως οι οντολογίες οι οποίες περιγράφονται σε μορφή RDF/XMl καθ' όλη τη διάρκεια της διαδικασίας, αρχίζοντας από την αποθήκευση στις προσωπικές συσκευές έως τελικά και στο Κεντρικό Πληροφορικό Σύστημα Υποστήριξης της μεθοδολογίας ONARM, επιστρέφουν αντίστροφα στους χρήστες των συσκευών Α και Β. Εάν οι χρήστες είναι on-line, οι φορητές συσκευές τους ενημερώνονται άμεσα μετά τον χρόνο επεξεργασίας, αλλιώς ενημερώνονται άμεσα, με το που θα συνδεθούν την επόμενη φορά σε αυτό. Στη συνέχεια ενημερώνεται το τοπικό γραφικό περιβάλλον των προσωπικών φορητών συσκευών των χρηστών Α και Β με τις νέες έννοιες.

Κατ' αυτόν τον τρόπο δημιουργείται ένα δίκτυο κοινών ενδιαφερόντων μεταξύ όλων των εγγεγραμμένων χρηστών του συστήματος.

Συνοπτικά η γενική περιγραφή απεικονίζεται στο παρακάτω διάγραμμα:



4.1.2. ΣΕΝΑΡΙΑ ΧΡΗΣΗΣ - UML ΔΙΑΓΡΑΜΜΑΤΑ

Στο παρακάτω διαγράμματα UML παρουσιάζεται η ροή των διαδικασιών που εκτελούνται στο σύστημα καθώς και ο τρόπος με τον οποίο αλληλεπιδρά ο χρήστης με τα βασικά

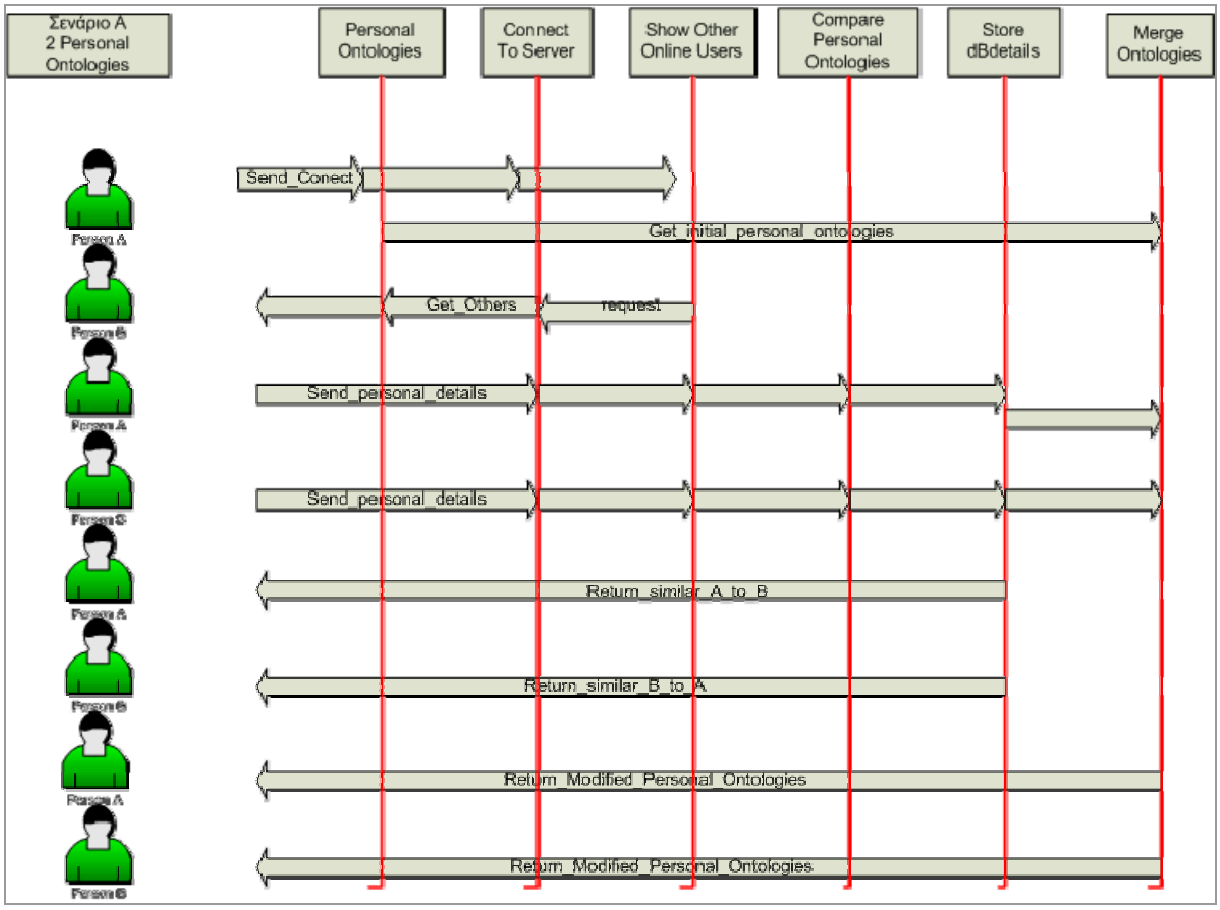
υποσυστήματα του συστήματος CONCEPT NET. Οι διαδικασίες αυτές εκτελούνται με βάση αποστολή μηνυμάτων, από και προς τον χρήστη του συστήματος, προς και από τα διάφορα υποσυστήματα (modules). Τα μηνύματα εμφανίζονται στα διαγράμματα σαν τόξα, ενώ τα υποσυστήματα σαν παραλληλόγραμμα στην κορυφή τους. Οι κόκκινες κάθετες γραμμές υποδηλώνουν την αντιστοίχηση τους καθ' όλη τη διάρκεια της διαδικασίας. Παρακάτω παρουσιάζονται δύο σενάρια χρήσης του συστήματος με τη μορφή UML διαγραμμάτων:

- *Σενάριο 1: User - User*
- *Σενάριο 2: User – System*

4.1.2.1. ΣΕΝΑΡΙΟ 1: USER - USER

Στο σενάριο user – user, θεωρείται ότι υπάρχουν δύο χρήστες του συστήματος Α και Β αντίστοιχα. Ο καθένας από αυτούς έχει την προσωπική του οντολογία, αποθηκευμένη στη προσωπική φορητή συσκευή του. Όταν οι δύο χρήστες Α και Β βρεθούν στον ίδιο χώρο, έτσι ώστε το σύστημα να μπορεί να τους ταυτοποιήσει σε αυτόν, τίθεται σε λειτουργία η μεθοδολογία ONARM, με απώτερο στόχο να επικοινωνήσουν οι δύο χρήστες, με έναν αυτόματο τρόπο και να γίνει ανταλλαγή ενδιαφερόντων με βάση τους προσωπικούς τους γράφους προτιμήσεων και ενδιαφερόντων.

Αν και στο παρακάτω διάγραμμα ροής εμφανίζεται η συμμετοχή δύο μόνων χρηστών στο σύστημα ταυτόχρονα, είναι δυνατόν να συμμετέχουν στη διαδικασία αυτή μεγαλύτερος αριθμός, δημιουργώντας κατ' αυτόν τον τρόπο ένα δυναμικό, διαρκώς μεταβαλλόμενο P2P δίκτυο χρηστών, παρομοίων ενδιαφερόντων και παρόμοιας αντίληψης και οργάνωσης των εννοιών. Τέλος μέσα από το P2P δίκτυο δημιουργείται η δυνατότητα online communities, σε αντίστοιχα γνωσιακά πεδία. Στο παρακάτω διάγραμμα παρουσιάζεται η ροή των διεργασιών κατά το σενάριο 1:



Τα βήματα τα οποία ακολουθούνται με βάση το ανωτέρω διάγραμμα, είναι συνοπτικά: Ο χρήστης Α έρχεται σε επαφή με το σύστημα και αποστέλλει μήνυμα στο σύστημα να του επιτρέψει να συνδεθεί σε αυτό σε πραγματικό χρόνο, να συνδεθεί στον κεντρικό server τους συστήματος και εν τέλει να του εμφανιστεί η λίστα με όλους του online χρήστες. Ακολούθως αποστέλλει τα προσωπικά του στοιχεία που του ζητούνται από το σύστημα καθώς και την προσωπική του οντολογία. Παρομοίως πράττει και ο χρήστης Β. Ακολούθως το σύστημα επιστρέφει στον Α και τον Β υποψήφιους παρόμοιους κόμβους, αναγνωρισμένους σαν σχετιζόμενους. Στο τελευταίο βήμα, επιστρέφει και στους δύο, τις προτάσεις του για την διαμόρφωση των προσωπικών τους οντολογιών με βάση κόμβους – έννοιες, σαν σημεία ενδιαφέροντος.

4.1.2.2. ΣΕΝΑΡΙΟ 2: USER - SYSTEM

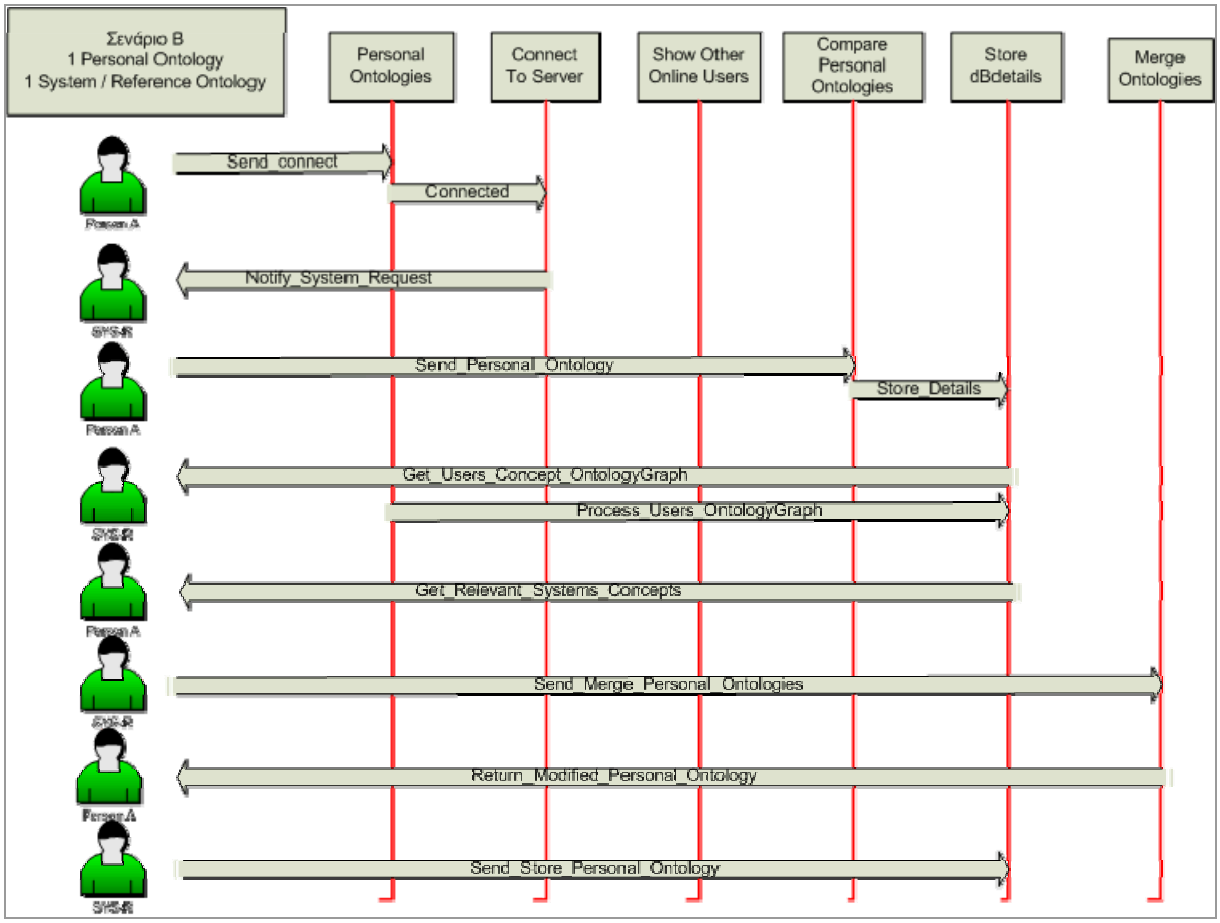
Στο δεύτερο σενάριο USER-SYSTEM, ο χρήστης Α είναι ένας κοινός χρήστης με φορητή συσκευή όπως και στο προηγούμενο σενάριο. Τη θέση του χρήστη Β σε αυτό το σενάριο παίρνει το ίδιο το σύστημα. Σε αυτήν την περίπτωση, αυτό που ουσιαστικά συμβαίνει, είναι ότι ο χρήστης Α εμπλουτίζει το γνωστικό του προφίλ, με βάση ενδιαφέρουσες έννοιες οι οποίες είναι απεικονισμένες σε μία "υπέρ-οντολογία" και ανήκει στο σύστημα, γύρω από το χώρο που τον ενδιαφέρει.

Είναι σημαντικό εδώ να αναφερθεί, ότι το σύστημα είναι σε θέση να κατέχει τέτοιου είδους "υπέρ-οντολογίες", από διαφορετικά γνωστικά πεδία και ανάλογα με το προφίλ του χρήστη, να του προωθεί μόνο σχετική και σχετιζόμενη γνώση.

Και σε αυτό το σενάριο χρήστης, σε γενικές γραμμές, ακολουθείται η ίδια σειρά ενεργειών, δηλαδή ο χρήστης ζητάει να συνδεθεί στο σύστημα και του επιστρέφεται μήνυμα αποδοχής και αναγνώρισης του χρήστη. Κατόπιν ο χρήστης Α στέλνει σε μορφή RDF/XML την προσωπική του οντολογία όπου αυτή επεξεργάζεται και συγκρίνεται με την "υπέρ-οντολογία" του συστήματος. Κατόπιν του επιστρέφονται από το σύστημα οι σχετικές έννοιες με μορφή RDF, αποθηκεύεται η νέα διαμορφωμένη κατάσταση του χρήστη (νέο προφίλ) και ενημερώνεται η φορητή συσκευή του χρήστη, μέσω της διαμορφωμένης οντολογίας.

Στη συνέχεια, το τοπικό γραφικό περιβάλλον του χρήστη Α στη φορητή συσκευή του αναπροσαρμόζεται και καλείται ο χρήστης να αποφασίσει εάν επιθυμεί η όχι, να κρατήσει τις προτεινόμενες αλλαγές και διαμορφώσεις στη προσωπική του οντολογία από το σύστημα.

Στο παρακάτω σχήμα εμφανίζονται οι παραπάνω διαδικασίες, με τη μορφή abstract UML διαγράμματος.



4.1.3. Περιβαλλον Υλοποιησης

Το παρόν πληροφοριακό σύστημα αναπτύχθηκε χρησιμοποιώντας μια αρκετά μεγάλη συλλογή από γλώσσες προγραμματισμού, βιβλιοθήκες και εργαλεία ανάπτυξης που διανέμονται ελεύθερα.

Όσον αφορά την client εφαρμογή που εκτελείται στην φορητή συσκευή, αναπτύχθηκε σε γλώσσα προγραμματισμού C#, η οποία αποτελεί μέρος του .NET Framework 2.0. Ειδικά για την περίπτωση της φορητής συσκευής χρησιμοποιήθηκε το .NET Compact Framework 2.0. Ο πηγαίος κώδικας της client εφαρμογής αναπτύχθηκε στο περιβάλλον ανάπτυξης Visual Studio Express Edition και χρησιμοποιώντας το Windows Mobile 6.0 SDK ώστε να γίνει πλήρης αξιοποίηση των χαρακτηριστικών της φορητής συσκευής και των δυνατοτήτων του λειτουργικού συστήματος της φορητής συσκευής. Αξίζει να σημειωθεί πως κατά την διάρκεια της ανάπτυξης χρησιμοποιήθηκε ο προσομοιωτής συσκευής Microsoft Device Emulator.

Για τις ανάγκες ανάπτυξης του Backend συστήματος, χρησιμοποιήθηκε η γλώσσα προγραμματισμού JAVA, ειδικότερα το JDK 1.6. Ο πηγαίος κώδικας του Backend συστήματος αναπτύχθηκε στο περιβάλλον ανάπτυξης Netbeans IDE 6.5 για Windows XP. Στον πυρήνα του συστήματος, για την ανάλυση των οντολογιών, χρησιμοποιήθηκε το JENA Framework 2.7, όπως αυτό αναπτύχθηκε από την Hewlett-Packard, σε συνδυασμό με τις γλώσσες περιγραφής μεταδεδομένων RDF/RDFS και XML. Επίσης, σε αρχικό στάδιο χρησιμοποιήθηκε και η γλώσσα αναζήτησης μεταδεδομένων SPARQL.

Τέλος, για την δημοσίευση (publishing) των υπηρεσιών διαδικτύου (Web Services) επιλέχθηκε ο Apache Tomcat Server.

4.1.4. Γενικη Αρχιτεκτονικη Συστηματος

4.1.4.1. Εισαγωγη

Ακολούθως παρουσιάζεται η γενική αρχιτεκτονική του συστήματος Concept-Net, που υλοποιεί την μεθοδολογία ONARM, η οποία εκτελείται σε κινητά τηλέφωνα, PDAs και άλλες φορητές συσκευές. Αποτελείται, όπως φαίνεται και στο διάγραμμα, από δύο (2) κύρια υποσυστήματα: Το υποσύστημα Front-End, αυτό το οποίο εμφανίζεται στην φορητή συσκευή του χρήστη και το υποσύστημα Back-End, το υποσύστημα δηλαδή που βρίσκεται κεντρικά και όπου υλοποιείται το αλγοριθμικό τμήμα της μεθοδολογίας ONARM.

Τα δύο υποσυστήματα μπορούν να επικοινωνούν μεταξύ τους μέσω Internet, υλοποιώντας ένα σύνολο υπηρεσιών διαδικτύου (Web Services).

Αναλυτικότερα, τα κύρια υποτμήματα (modules) του καθενός εκ των δύο υποσυστημάτων, έχουν ως ακολούθως με βάση τη λειτουργικότητα:

Υποσύστημα Front-End

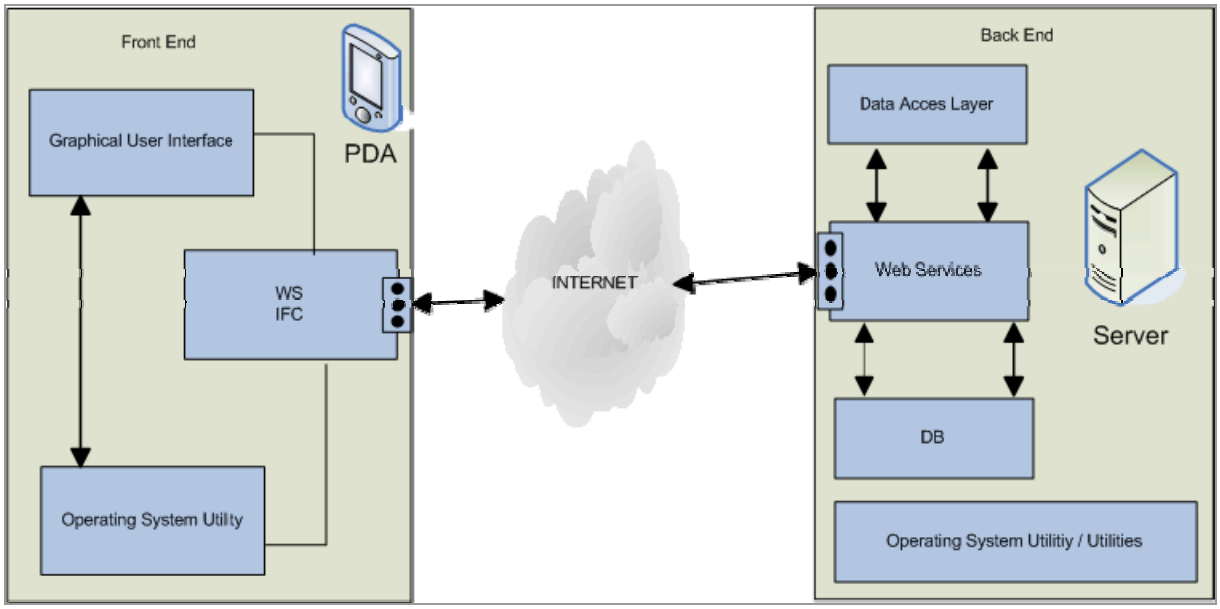
1. Τμήμα Γραφικών Διεπαφών
2. Τμήμα Επικοινωνίας (με υπηρεσίες Web)
3. Τμήμα Λειτουργικότητας Λειτουργικού Συστήματος και άλλων βοηθητικών προγραμμάτων

Υποσύστημα Back-End

1. Τμήμα Πρόσβασης Δεδομένων (data access layer)
2. Τμήμα Επικοινωνίας Υποσυστημάτων (Web Services)
3. Τμήμα Αποθήκευσης Δεδομένων
4. Τμήμα Λειτουργικότητας Λειτουργικού Συστήματος και άλλων βοηθητικών προγραμμάτων

Παρακάτω ακολουθεί πλήρης περιγραφή τόσο του καθενός υποσυστήματος (Front End – Back End) όσο και των τμημάτων του καθενός, όπως αυτά προαναφέρθηκαν ανωτέρω

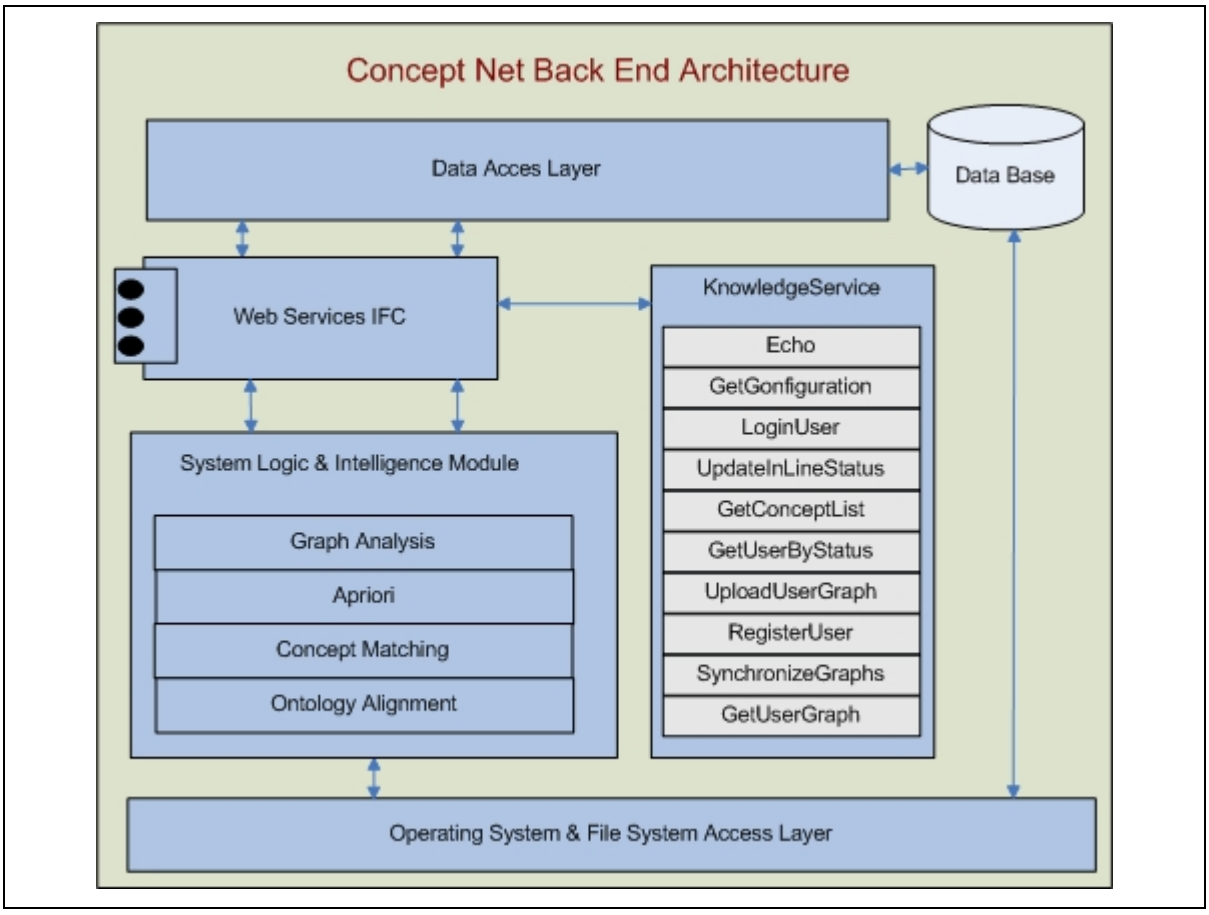
Στο παρακάτω διάγραμμα που ακολουθεί, φαίνεται η αρχιτεκτονική και η διασυνδεσιμότητα των δύο κύριων υποσυστημάτων.



4.1.4.2. ΑΡΧΙΤΕΚΤΟΝΙΚΗ BACKEND

Λειτουργικότητα

Η κυρίως λειτουργικότητα του συστήματος έχει υλοποιηθεί στο υποσύστημα Back- End του οποίου η αρχιτεκτονική φαίνεται στο παρακάτω διάγραμμα



Χωρίζεται στα παρακάτω λειτουργικά τμήματα (modules) ανάλογα με τις διαδικασίες που εκτελούνται σε αυτά

1. Τμήμα Πρόσβασης Δεδομένων (Data Access Layer)
2. Τμήμα Επικοινωνίας (Web Services)
3. Τμήμα Βάσης Δεδομένων
4. Τμήμα Λογικής Συστήματος Concept-Net
5. Τμήμα Λειτουργικότητας Λειτουργικού Συστήματος και άλλων βοηθητικών προγραμμάτων

Το τμήμα πρόσβασης δεδομένων είναι υπεύθυνο για την εκτέλεση των εργασιών στη βάση δεδομένων του συστήματος όπου έχει αποθηκευθεί η γνώση του συστήματος (knowledge DB) καθώς επίσης και όλες οι σχετικές ρυθμίσεις. Πιο συγκεκριμένα, μέσα σε αυτό υλοποιούνται οι συναρτήσεις ερωταποκρίσεων μέσω της λειτουργικότητας DBUtility προς τη βάση.

Η πρόσβαση στα δεδομένα του συστήματος γίνεται αφού πρώτα κληθούν τα Web Services πού είναι υλοποιημένα όλα μαζί στα Knowledge Service, ανάλογα με τη εκάστοτε λειτουργικότητα που πρέπει να εκτελεστεί κάθε φορά. Η knowledge Service υλοποιεί ένα πλήθος από 10 συναρτήσεις οι οποίες της επιτρέπουν να επικοινωνεί η ίδια και μέσω αυτής όλο το Back End υποσύστημα με το εξωτερικό φλοιό, όπου υπάρχει η επικοινωνία με την Client εφαρμογή (Front End).

Εκτός των ανωτέρω, η κύρια λειτουργικότητα της Knowledge Service είναι να αποστέλλει γράφους-οντολογίας στο τμήμα του υποσυστήματος System Logic Graph Analysis, όπου εκτελούνται όλες οι διαδικασίες Similarity, Matching και knowledge Extraction των οντολογιών και των εννοιών. Επίσης η knowledge Service αλληλεπιδρά και με το λειτουργικό σύστημα μέσω του υποσυστήματος που εμπεριέχει τέτοιου είδους utilities και με τα τμήματα settings όπου αποθηκεύονται προτιμήσεις του user (προφίλ) σε files, HDD).

Συνοπτικά, η λειτουργικότητα που υλοποιείται μέσω της αρχιτεκτονικής στο Back End σύστημα, είναι η ακόλουθη:

1. Υποδοχή μηνυμάτων επικοινωνίας από τις φορητές συσκευές των χρηστών
2. Υποδοχή και αποστολή προσωπικών οντολογιών των χρηστών.
3. Επεξεργασία των προσωπικών οντολογιών των χρηστών με βάση τη μεθοδολογία ONARM (Αλγόριθμος Apriori, Concept Matching, Similarity κλπ.)
4. Αποθήκευση στοιχείων χρηστών συστήματος είτε αυτά αφορούν τη χρήση του (login, ….) είτε αφορούν το γνωστικό προφίλ και προτιμήσεις του χρήστη.

Οι τεχνικές επιλογές ως προς την σχεδίαση της αρχιτεκτονικής, είναι τέτοιες, οι οποίες επιτρέπουν εύκολα την αναβάθμιση του συστήματος και την υλοποίησή του σε διαφορετικές τεχνολογικές πλατφόρμες.

4.1.4.3. ΑΡΧΙΤΕΚΤΟΝΙΚΗ BACKEND ΜΕ ΒΑΣΗ CLASS – UML ΔΙΑΓΡΑΜΜΑΤΑ

Τα web service methods αποτελούν ουσιαστικά τις μονές διεπαφές ανάμεσα στην client εφαρμογή που εκτελείται στην φορητή συσκευή και στο Backend σύστημα. Παρακάτω ακολουθεί λεπτομερής περιγραφή αυτών των διεπαφών καθώς και όλων των παραμέτρων εισόδου και εξόδου των.

Στον παρακάτω πίνακα εμφανίζονται οι συγκεκριμένες συναρτήσεις που υλοποιήθηκαν για να υποστηρίξουν την λειτουργικότητα του χρήστη του συστήματος, τόσο σε επίπεδο διαχειριστικό, έτσι ώστε η εφαρμογή να αποτελεί μια ολοκληρωμένη πλατφόρμα λογισμικού όσο και πιο συγκεκριμένα σε επίπεδο υποστήριξης λειτουργικότητος που αφορούν την διαχείριση των εννοιών και των οντολογιών – γράφων που υποστηρίζουν αυτές σαν δομές.

Όνομα Web Method	Echo
Περιγραφή	Η συγκεκριμένη μέθοδος βοηθάει στον έλεγχο της συνδεσιμότητας της Client εφαρμογής με το Backend σύστημα. Ειδικότερα, η μέθοδος Echo, με το να επιστρέφει το μήνυμα το οποίο έχει προηγουμένως λάβει ως είσοδο, σηματοδοτεί πως το σύστημα είναι σε ετοιμότητα.
Παράμετροι Εισόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
Input	String
Παράμετροι Εξόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
Output	String

Όνομα Web Method	GetConfiguration
Περιγραφή	Επιστρέφει μια περιληπτική αναφορά περιγράφοντας όλες τις ενεργές ρυθμίσεις και παραμέτρους με τις οποίες λειτουργεί το Backend σύστημα, π.χ. το όνομα και τη διεύθυνση της βάσης δεδομένων, τους τοπικούς φακέλους αποθήκευσης των οντολογιών/γράφων, την έκδοση του λογισμικού κτλ
Παράμετροι Εισόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
-	-
Παράμετροι Εξόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
ActiveConfiguration	String

Όνομα Web Method	LoginUser
Περιγραφή	Αναλαμβάνει να ταυτοποιήσει τον χρήστη. Έτσι, λαμβάνοντας τα διαπιστευτήρια του χρήστη, που αποτελούνται από το username και το password του, ελέγχει τη βάση δεδομένων, όπου είναι καταχωρημένοι όλοι οι χρήστες του συστήματος, και επιτρέπει ή απαγορεύει την πρόσβαση στον χρήστη.
Παράμετροι Εισόδου	

Όνομα Παραμέτρου	Τύπος Παραμέτρου
Username	String
Password	String
Παράμετροι Εξόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
LoginResult	String

Όνομα Web Method	RegisterUser
Περιγραφή	Η συγκεκριμένη μέθοδος λαμβάνει τα βασικά προσωπικά στοιχεία του νέου χρήστη στην είσοδό της και αναλαμβάνει να τον εγγράψει στη βάση δεδομένων του συστήματος. Στο τέλος της επιστρέφει μια σύντομη αναφορά με το αν η καταχώρηση έγινε με επιτυχία ή παρουσιάστηκε κάποιο σφάλμα.
Παράμετροι Εισόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
Firstname	String
Lastname	String
Username	String
Password	String
Παράμετροι Εξόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
RegistrationResult	String

Όνομα Web Method	UpdateOnlineStatus
Περιγραφή	Το Backend σύστημα διατηρεί ένα επικαιροποιημένο μητρώο με όλους τους συνδεδεμένους χρήστες. Η συγκεκριμένη μέθοδος ενημερώνει το σύστημα για την κατάσταση της σύνδεσης κάποιου χρήστη, π.χ. για το αν συνδέθηκε ή για το αν αποσυνδέθηκε.
Παράμετροι Εισόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
UserID	String
NewStatus	String

Παράμετροι Εξόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
UpdateStatusResult	String

Όνομα Web Method	GetConceptsList
Περιγραφή	Επιστρέφει ένα πίνακα με όλες τις έννοιες που έχουν καταχωρηθεί στο σύστημα.
Παράμετροι Εισόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
-	-
Παράμετροι Εξόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
ConceptsList	Concept[]

Όνομα Web Method	GetUsersByStatus
Περιγραφή	Επιστρέφει ένα πίνακα με όλους τους χρήστες του συστήματος. Επίσης, υπάρχει η δυνατότητα να επιστραφούν μόνο οι συνδεδεμένοι στο σύστημα χρήστες μέσω κατάλληλης παραμέτρου εισόδου.
Παράμετροι Εισόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
Status	String
Παράμετροι Εξόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
UsersList	User[]

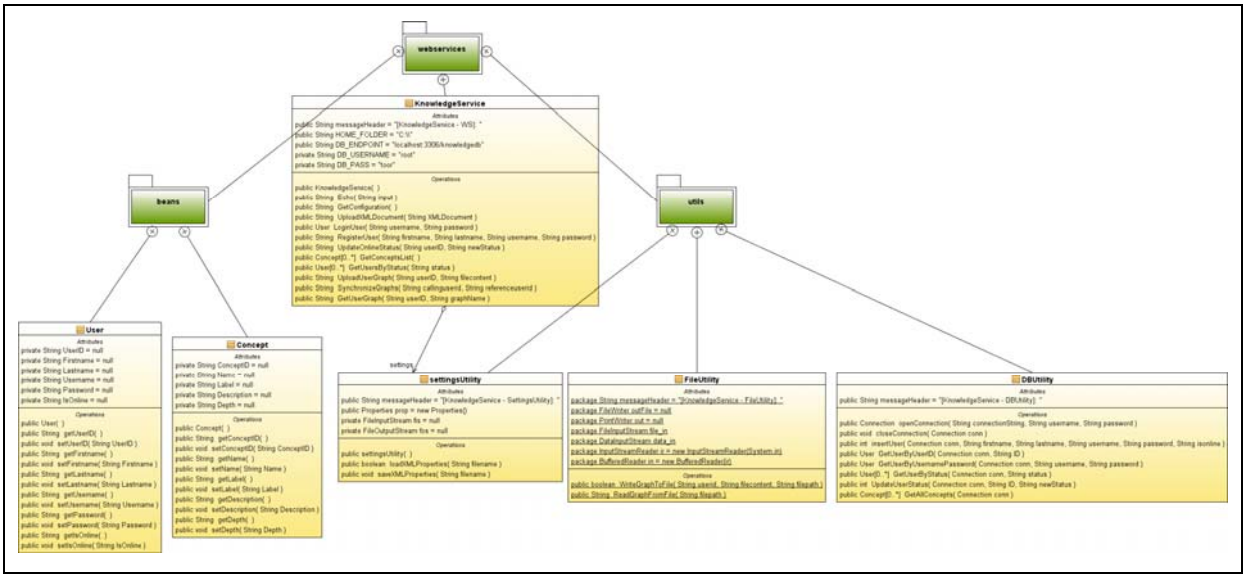
Όνομα Web Method	UploadUserGraph
Περιγραφή	Αναλαμβάνει την αποθήκευση μιας οντολογίας/γράφου ενός χρήστη στο Backend σύστημα. Στην είσοδό της δέχεται, μεταξύ άλλων, τον μοναδικό κωδικό χρήστη καθώς και το όνομα του γράφου ώστε να υπάρχει αναφορά σε ποιον χρήστη ανήκει ο κάθε

	γράφος. Στο τέλος της, η μέθοδος επιστρέφει μια σύντομη αναφορά με το αν η αποθήκευση έγινε με επιτυχία ή παρουσιάστηκε κάποιο σφάλμα.
Παράμετροι Εισόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
UserID	String
FileContent	String
GraphName	String
Παράμετροι Εξόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
UploadResult	String

Όνομα Web Method	SynchronizeGraphs
Περιγραφή	Η συγκεκριμένη μέθοδος υλοποιεί τον αλγόριθμο αναγνώρισης και ταυτοποίησης κόμβων που έχει παρουσιαστεί με λεπτομέρεια σε προηγούμενο κεφάλαιο και αναλαμβάνει να συγκρίνει την οντολογία/γράφο ενός χρήστη με την αντίστοιχη οντολογία/γράφο κάποιου άλλου χρήστη ή ακόμη και με την οντολογία αναφοράς που υπάρχει στο σύστημα. Το αποτέλεσμα αυτής της σύγκρισης είναι μια νέα έκδοση της αρχικής οντολογίας του χρήστη, εμπλουτισμένη με περισσότερη γνώση.
Παράμετροι Εισόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
CallingUserID	String
ReferenceUserID	String
Παράμετροι Εξόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
SynchronizationResult	String

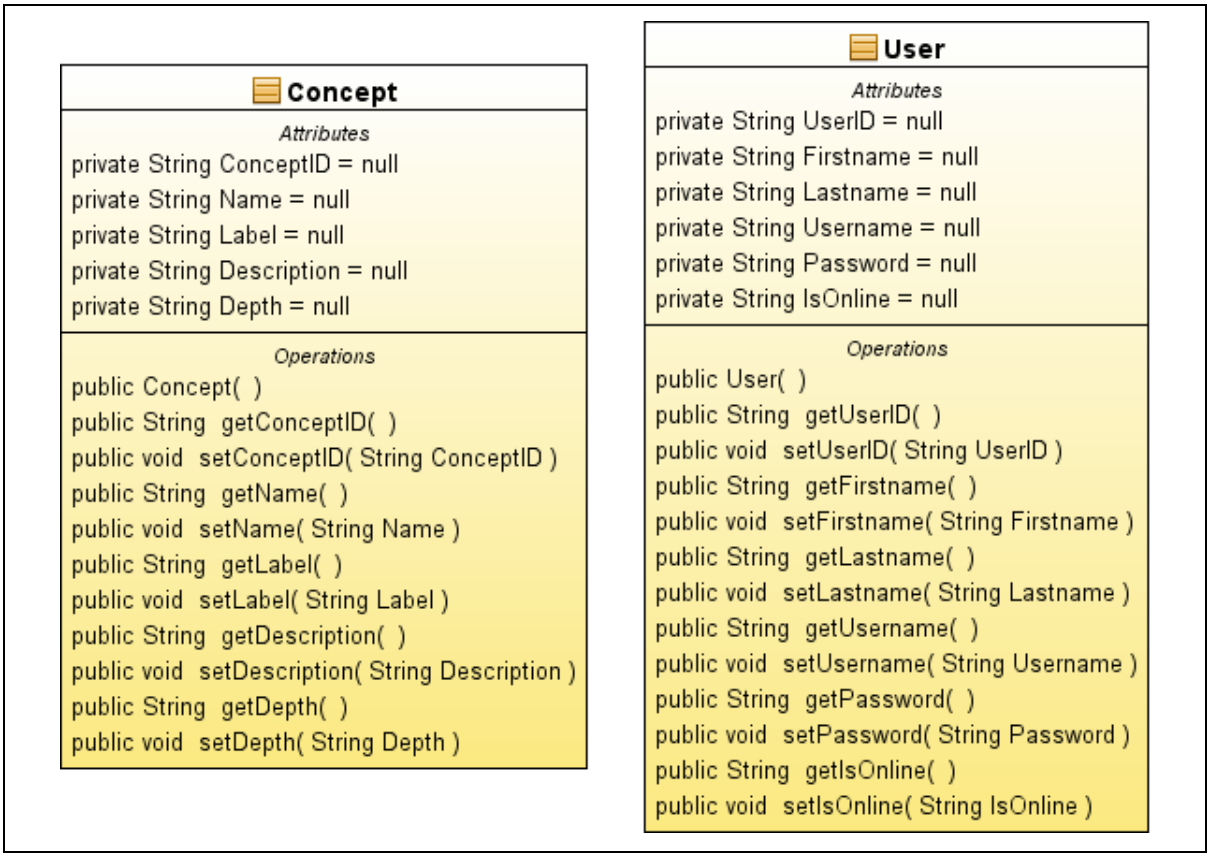
Όνομα Web Method	GetUserGraph
Περιγραφή	Αναλαμβάνει την μεταφορά μιας οντολογίας/γράφου ενός χρήστη από το Backend σύστημα προς την client εφαρμογή.
Παράμετροι Εισόδου	

Όνομα Παραμέτρου	Τύπος Παραμέτρου
UserID	String
GraphName	String
Παράμετροι Εξόδου	
Όνομα Παραμέτρου	Τύπος Παραμέτρου
SynchronizedGraph	String

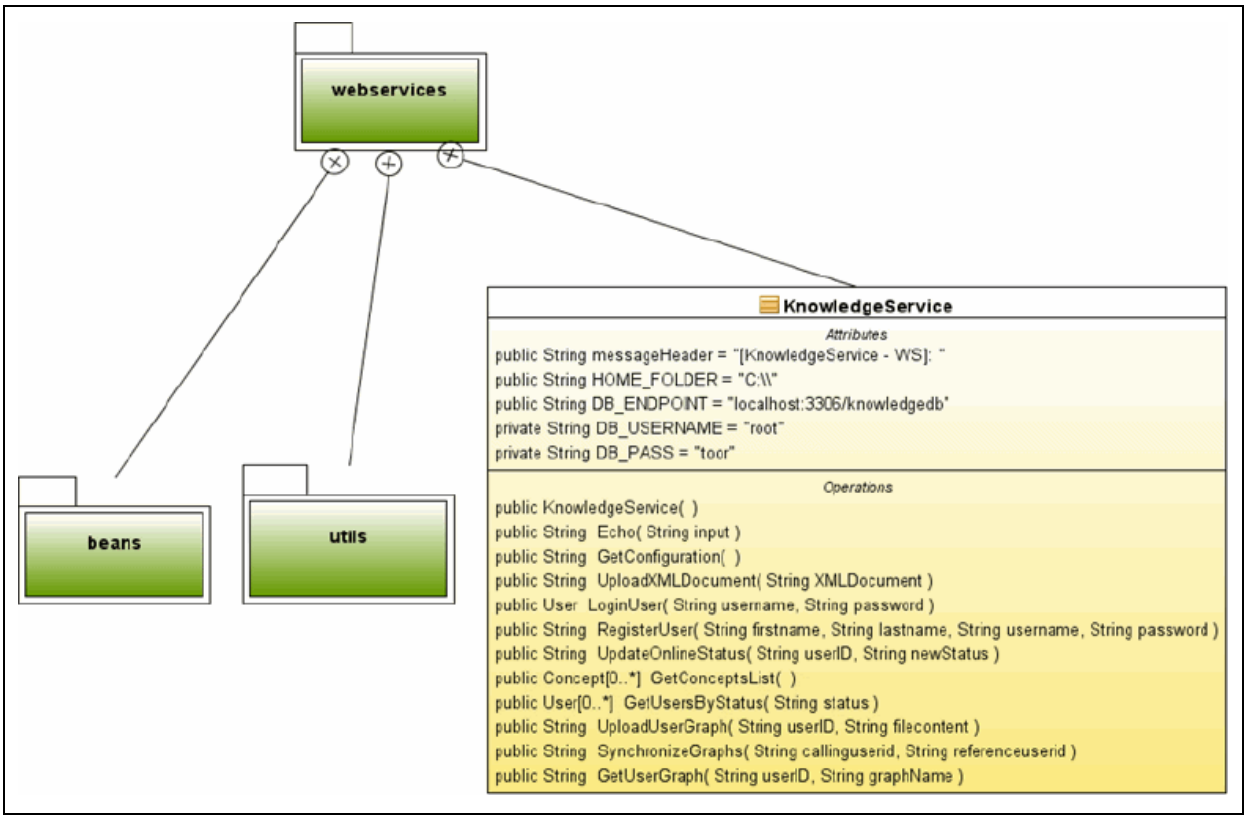


Στα παρακάτω διαγράμματα απεικονίζονται οι δύο (2) κλάσεις των data objects που ανταλλάσσονται στην client εφαρμογή και στο backend σύστημα.

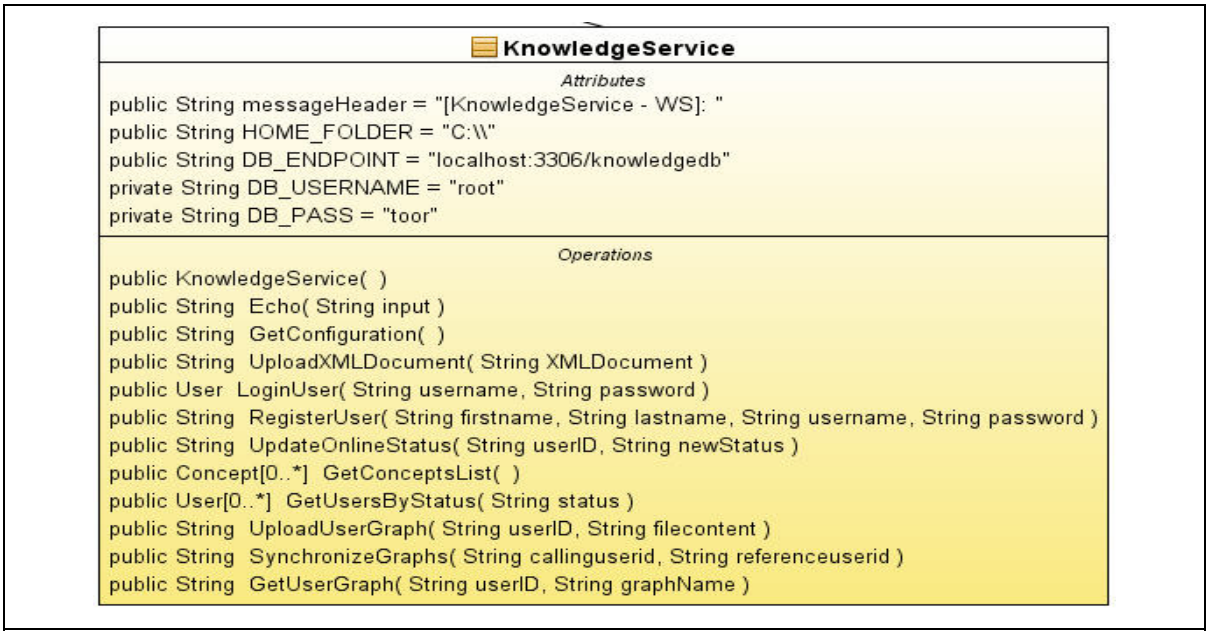
Με το data object Concept περιγράφεται η δομή μίας πλήρους έννοιας όπως αυτή απεικονίζεται στο σύστημα ενώ αντίστοιχα το data object User περιγράφει τη δομή της κλάσης ενός χρήστη, όπως αυτός γίνεται αντιληπτός από το σύστημα.



Στο παρακάτω διάγραμμα φαίνεται η συσχέτιση ανάμεσα στις κύριες κλάσεις του υποσυστήματος των WEB Services.



Στο παρακάτω διάγραμμα φαίνεται αναλυτικά η δομή της κλάσης KnowledgeService, που υλοποιεί όλα τα WEB methods, που παρουσιάστηκαν ανωτέρω.



Στο παρακάτω διάγραμμα φαίνεται αναλυτικά η δομή των κλάσεων που περιλαμβάνονται στο πακέτο Utils. Το πακέτο Utils αποτελείται από 3 διακριτές κλάσεις:



Ειδικότερα, η κάθε κλάση αναλύεται παρακάτω:

DBUtility class

Η class DBUtility υλοποιεί το επίπεδο πρόσβασης στη βάση δεδομένων του συστήματος (Data Access Layer – DAL). Πιο συγκεκριμένα, μέσω συναρτήσεων που περιλαμβάνει, υποβάλλει ερωτήματα προς τη βάση δεδομένων και λαμβάνει τις αντίστοιχες αποκρίσεις από αυτή, με τη μορφή sql queries.

Επίσης στην κλάση αυτή υλοποιείται και η συνολικά διαχείριση λαθών που προκύπτουν από τη αλληλεπίδραση του backend συστήματος με τη βάση δεδομένων.

FileUtility class

Η FileUtility ικανοποιεί τις ανάγκες διαχείρισης (δημιουργίας, ανάγνωσης, διαγραφής, ενημέρωσης) αρχείων, τα οποία πρέπει να καταγράφονται στο file system. Τέτοια αρχεία είναι αυτά που δημιουργούνται για παράδειγμα κατά τη δημιουργία ενός γράφου, που απεικονίζει το σύνολο των εννοιών που ενδιαφέρουν ένα χρήστη του συστήματος. Επίσης σε αυτό τον τύπο αρχείου, βασίζεται και η απεικόνιση των συνολικών γράφων – οντολογιών που χρησιμοποιούνται σαν αναφορές (reference) εννοιών.

Επίσης η κλάση αυτή χρησιμοποιείται και για τις ανάγκες ανάλυσης των υπό εξέταση οντολογιών, εκεί όπου θεωρείται να καταγράφεται πληροφορία η οποία θα είναι διαθέσιμη για περαιτέρω επεξεργασία σε κατοπινά στάδια ανάλυσης και επεξεργασίας αυτών. Για παράδειγμα, η κλάση αυτή χρησιμοποιείται για να καταγράφονται σε αρχεία τα leaf nodes των προς σύγκριση οντολογιών – γράφων. Επίσης σε αρχεία καταγράφονται και όλα τα ενδιάμεσα αποτελέσματα – στάδια του αλγορίθμου Apriori.

Τέλος, στην κλάση αυτή υλοποιείται και η συνολικά διαχείριση λαθών που προκύπτουν από τη αλληλεπίδραση του backend συστήματος με το file system.

SettingsUtility class

Στην κλάση SettingsUtility, υλοποιείται η λειτουργικότητα, η οποία έχει να κάνει με το configuration όλων των υποσυστημάτων του backend συστήματος, έτσι ώστε να είναι δυνατή η απρόσκοπτη λειτουργία του συστήματος, μέσω γενικών ρυθμίσεων, οι οποίες εύκολα μπορούν να μεταβληθούν προκειμένου να τρέξει η εφαρμογή σε άλλο σύστημα. Για παράδειγμα τέτοιες ρυθμίσεις αφορούν σύνδεση με τη βάση δεδομένων, τοποθεσία για την αποθήκευση αρχείων, παράμετροι που αφορούν το optimization των αλγορίθμων που χρησιμοποιούνται κλπ.

Για παράδειγμα το παρακάτω απόσπασμα αρχείου XML, δηλώνει τέτοιες ρυθμίσεις:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <!DOCTYPE properties (View Source for full doctype...)>
- <properties version="1.0">
```

```
<comment>Knowledge Web Services - Settings File</comment>
<entry key="homeFolder">c:\\GraphsRepository</entry>
<entry key="version">0.1</entry>
<entry key="dbEndpoint">localhost:3306/knowledgedb?useUnicode=true&characterEncoding=UTF-8</entry>
<entry key="dbUsername">root</entry>
<entry key="dbPassword">toor</entry>
<entry key="releaseDate">01-04-2009</entry>
</properties>
```

4.1.4.4. ΑΡΧΙΤΕΚΤΟΝΙΚΗ FRONTEND

Λειτουργικότητα

Στο σημείο αυτό παρουσιάζεται η αρχιτεκτονική του τμήματος Front End της εφαρμογής, δηλαδή αυτής που εκτελείται στην φορητή συσκευή του χρήστη.

Η αρχιτεκτονική αποτελείται από τα παρακάτω κύρια τμήματα.

Το τμήμα των διεπαφών του χρήστη (forms) όπου εκεί υπάρχουν όλα τα user interfaces όπως αυτό του Graph Editor το οποίο είναι το γραφικό περιβάλλον στο οποίο ο χρήστης απεικονίζει την προσωπική του οντολογία .

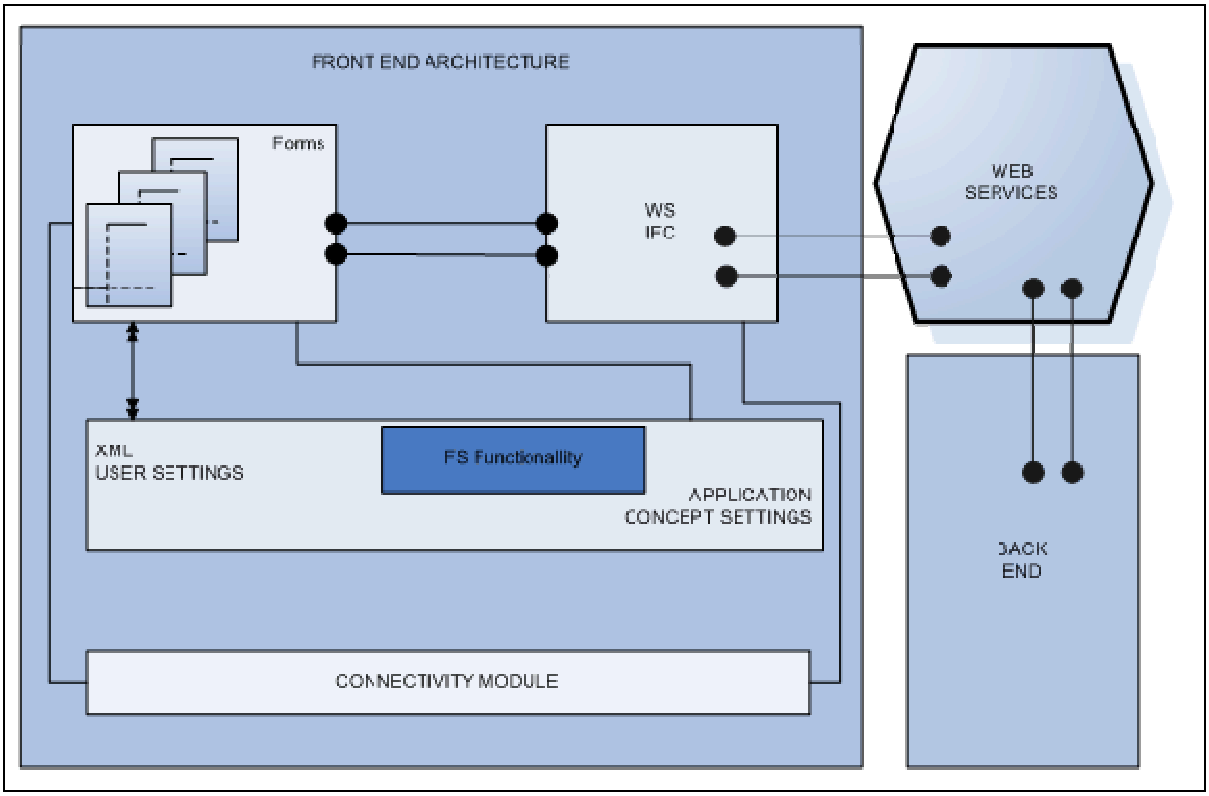
Αυτό έρχεται σε επικοινωνία με το τμήμα των WS (Web Services) το οποίο είναι υπεύθυνο για το επικοινωνιακό κομμάτι της εφαρμογής στη φορητή συσκευή.

Ακολούθως υπάρχει το τμήμα το οποίο είναι υπεύθυνο για την αποθήκευση των προσωπικών οντολογιών με τη μορφή XML στη φορητή συσκευή κάνοντας χρήση του τμήματος Application Concept Settings.

Όλα τα επικοινωνιακά δεδομένα (διευθύνσεις, ports, portals κλπ) διαχειρίζονται από το connectivity τμήμα.

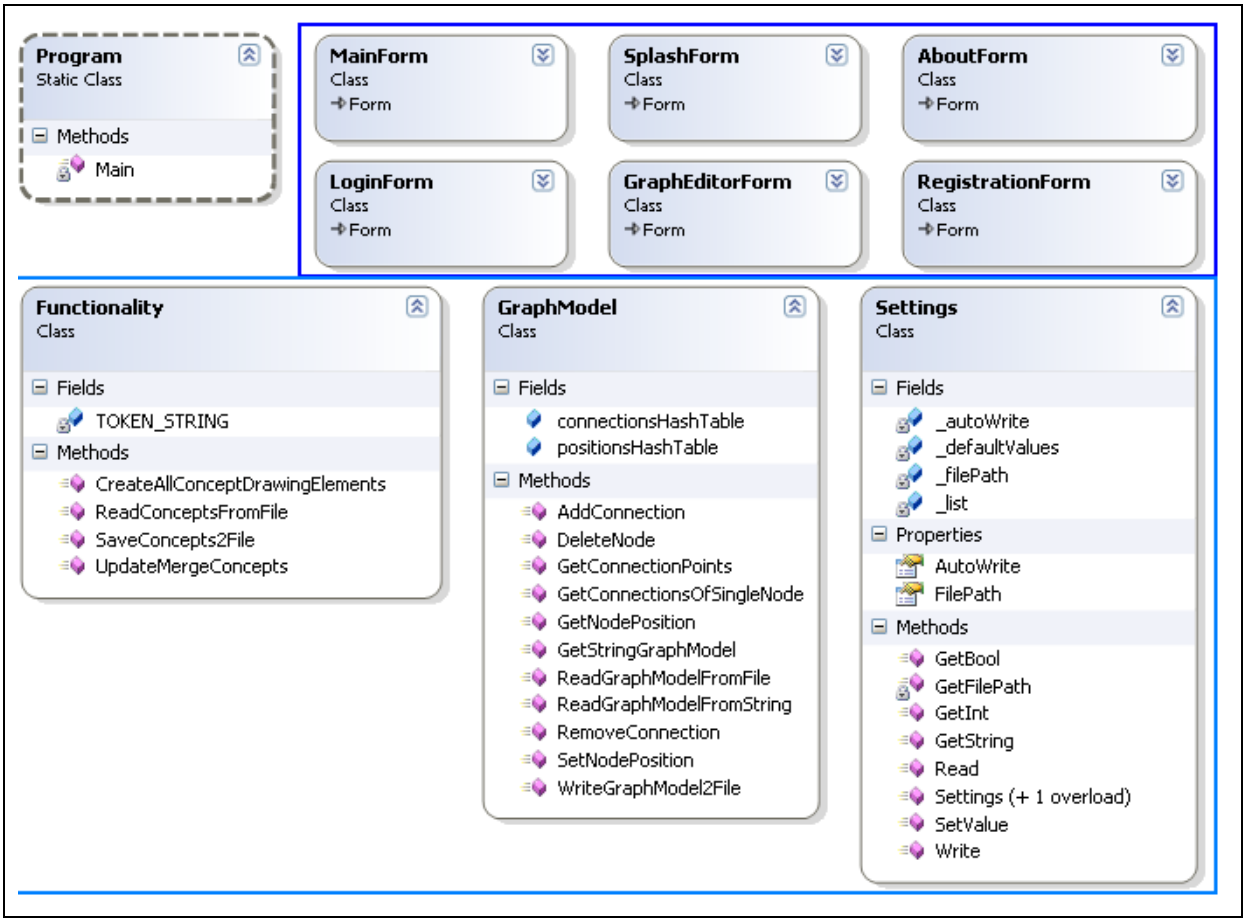
Η εφαρμογή client είναι έτσι σχεδιασμένη έτσι ώστε να χρησιμοποιεί ευρέως διαδεδομένα πρωτόκολλα του Διαδικτύου.

Στο παρακάτω διάγραμμα εμφανίζονται τα τμήματα του υποσυστήματος client με βάση τη λειτουργικότητα της εφαρμογής.


FRONT END ARCHITECTURE
Forms
WS
IFC
WEB
SERVICES
XML
USER SETTINGS
FS Functionality
APPLICATION
CONCEPT SETTINGS
BACK
END
CONNECTIVITY MODULE

4.1.4.5. Αρχιτεκτονικη Frontend Με Βαση CLASS – UML Διαγραμματα

Στο παρακάτω διάγραμμα παρουσιάζεται η αρχιτεκτονική της εφαρμογής clientμε βάση τις λειτουργικές ανάγκες ομαδοποιημένες σε κλάσεις χρηστικότητας, όπως αυτές υλοποιήθηκαν σε μορφή κώδικα.



Program class

Η κλάση Program, αποτελεί το σημείο εισαγωγής στην εφαρμογή που τρέχει στο περιβάλλον client και είναι υπεύθυνη για την αρχικοποίηση αυτής και κατόπιν για την παρακολούθηση της καλής εκτέλεσής της (execution) αυτής.

Settings class

Στην κλάση Settings, υλοποιείται η λειτουργικότητα, η οποία έχει να κάνει με το configuration που έχει να κάνει με την εφαρμογή η οποία τρέχει στη συσκευή client (PDA),

έτσι ώστε να λαμβάνει υπ' όψιν κάθε φορά τις ιδιαιτερότητες της κάθε συσκευής και να λειτουργεί με τον καλύτερο δυνατό τρόπο.

Για παράδειγμα, το παρακάτω απόσπασμα

```
<configuration>
  <appSettings>
    <add key="CurrentVersion" value="0.5" />
    <add key="ConceptsFilepath" value="\Program Files\knowledgeclient\ConceptsList.txt" />
    <add key="CurrentGraphpath" value="\Program Files\knowledgeclient\defaultGraph.txt" />
    <add key="WS_EndPoint" value="http://62.38.191.195:8084/WebServiceLayer/KnowledgeService?wsdl" />
    <add key="AboutMessage" value="Concept Net has been developed as part of PhD thesis of Mr. Christos Tatsiopoulos, under
the pupersision of Prof. Vasilios Voutsinas at the Artificial Intelligence Laboratory, University of Patras, Rio, Greece. All rights
reserved." />
  </appSettings>
</configuration>
```

προβάλλει ρυθμίσεις σχετικά με τη διαθεσιμότητα των σχετιζομένων με τον χρήστη υπηρεσίες, τόπος αποθήκευσης προσωπικών γράφων χρήστη τοπικά, στοιχεία, προτιμήσεις υπό μορφή εννοιών, κλπ

GraphModel class

Η κλάση GraphModel, υλοποιεί μια δομή αποθήκευσης, αναπαράστασης και διαχείρισης του γράφου – οντολογίας στη μνήμη της φορητής συσκευής. Να σημειωθεί ότι η κλάση GraphModel δεν υλοποιεί την γραφική αναπαράσταση του γράφου, η οποία ανατίθεται στην κλάση GraphEditorForm που αναφέρεται και επεξηγείται παρακάτω.

Για παράδειγμα, η κλάση GraphModel, κρατά στη μνήμη τις σχέσεις ανάμεσα στους κόμβους της οντολογίας, τις σχετικές θέσεις των κόμβων μέσα σε αυτή, τις συντεταγμένες τους, κλπ.

Τέλος, στην κλάση αυτή υλοποιείται και η συνολική διαχείριση λαθών που προκύπτουν στα πλαίσια κλήσεων συναρτήσεων της συγκεκριμένης κλάσης.

Functionality class

Στην κλάση Functionality class, υλοποιείται λειτουργικότητα η οποία δεν δύναται να ενταχθεί σε κανένα από τα παραπάνω υποσυστήματα, αλλά είναι απαραίτητη για την ορθή λειτουργία του συστήματος.

Για παράδειγμα, τέτοιου είδους λειτουργικότητα αφορά τη διαχείριση των εννοιών – concepts, αρχικοποίηση στοιχείων των γραφικών διεπαφών του client συστήματος, κλπ.

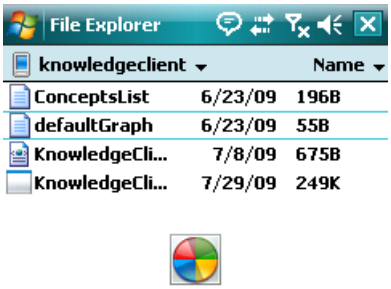
UI Forms Group classes

Στην ομάδα των κλάσεων αυτών, βρίσκονται οι κλάσεις που υλοποιούν το γραφικό περιβάλλον και τις διεπαφές μεταξύ της εφαρμογή και του χρήστη του συστήματος στην φορητή συσκευή (PDA). Οι κλάσεις αυτές, έχουν ως αποτέλεσμα τη δημιουργία διαδραστικών (interactive) φορμών στη συσκευή. Αναλυτικότερα, οι κλάσεις αυτές είναι οι παρακάτω:

- SplashForm: Η φόρμα αυτή είναι η πρώτη που εμφανίζεται με την έναρξη της εφαρμογής και παρουσιάζει διάφορα σχετικά με την εφαρμογή στοιχεία.
- MainForm: Η φόρμα, είναι η κύρια φόρμα με την οποία διαλειτουργεί ο χρήστης. Περιλαμβάνει το κυρίως μενού και προσφέρει τη συνδεσιμότητα στον χρήστη με τις υπόλοιπες φόρμες
- LoginForm: Η φόρμα υπεύθυνη για την ταυτοποίηση και την εισαγωγή του χρήστη στο σύστημα.
- RegistrationForm: Η φόρμα αυτή προσφέρει τη δυνατότητα σε κάποιο νέο χρήστη να εγγραφεί στο σύστημα.
- GraphEditorForm: Η φόρμα αυτή προσφέρει στο χρήστη το διαδραστικό γραφικό περιβάλλον για τη σχεδίαση, διαχείριση και ενημέρωση των γράφων – οντολογιών στη προσωπική φορητή συσκευή του.
- AboutForm; Η φόρμα αυτή παρουσιάζει γενικές πληροφορίες σχετικά με το status της εφαρμογής.

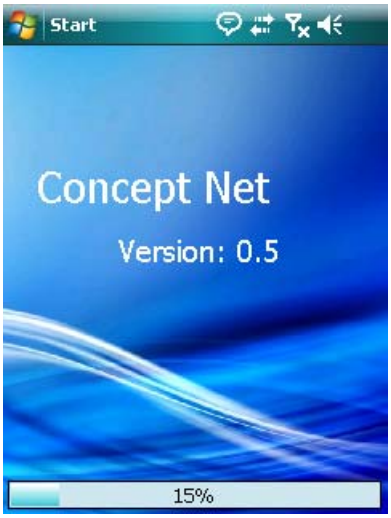
4.1.5. USER INTERFACES - SCREENSHOTS

Παρακάτω παρουσιάζονται τα κυριότερα user interfaces στη φορητή συσκευή του χρήστη της εφαρμογής Concept Net.



Ο χρήστης ξεκινάει την εφαρμογή του στο PDA, μέσα από το file explorer του συστήματος ή επιλέγοντας την αντίστοιχη συντόμευση της εφαρμογής, μέσα από το menu “Programs”.

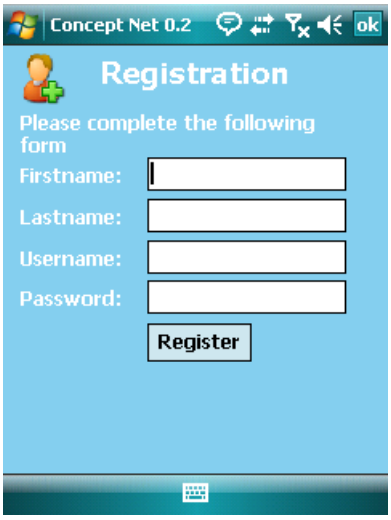




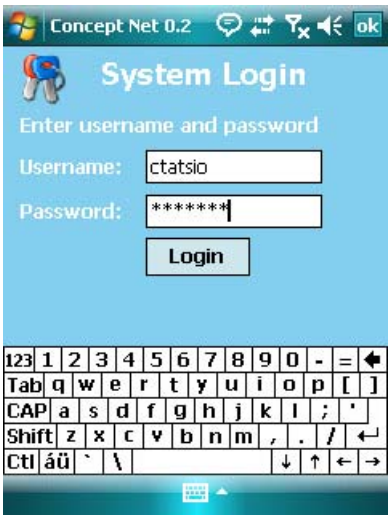
Στη διπλανή εικόνα φαίνεται η αρχική οθόνη (splash – screen) της εφαρμογής η οποία εκτελείται στη φορητή συσκευή του χρήστη. Στο κάτω μέρος διακρίνεται ο δείκτης για τον έλεγχο προόδου φόρτωσης της εφαρμογής.



Στη διπλανή εικόνα φαίνεται η αμέσως επόμενη οθόνη της εφαρμογής. Εμφανίζεται με τη μορφή ενός menu, που δίνει δυνατότητες επί μέρους επιλογών, μέσω εύχρηστων buttons. Οι επιλογές είναι: Login, Join Knowledge Network, Concept Browser, Graph Editor, Settings, About. Επίσης στο κάτω μέρος της οθόνης εμφανίζεται και μία σειρά από λειτουργικότητα, με τη μορφή των tabs, για περισσότερες δυνατότητες πλοήγησης στο σύστημα.



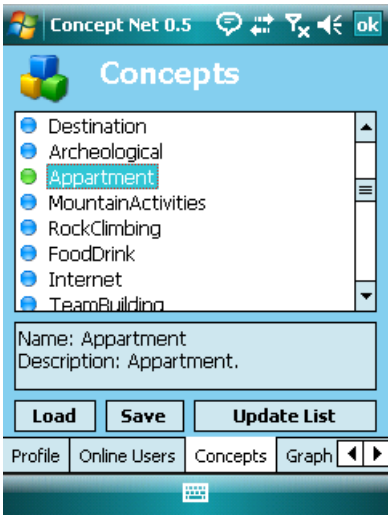
Ακολούθως, ο χρήστης μέσω της επιλογής Join Knowledge Network, εγγράφεται (την πρώτη φορά μόνο), στο σύστημα, συμπληρώνοντας στη φόρμα τα στοιχεία του, όπως φαίνεται και στην διπλανή εικόνα.



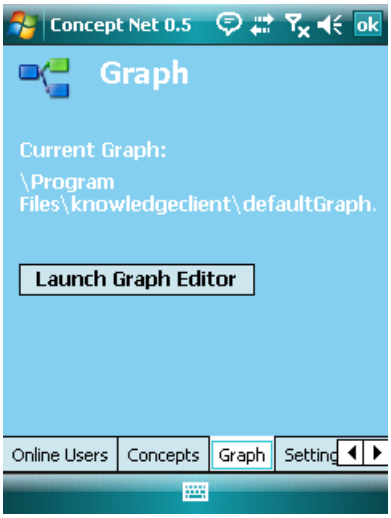
Κατόπιν ο χρήστης μέσω της κατάλληλης φόρμας κάνει login στο σύστημα, εισάγοντας το username και το password του.



Στη συνέχεια, εφ' όσον η διαδικασία ταυτοποίησης είναι επιτυχής, το σύστημα, διαθέτει στο χρήστη το profile του ενημερωμένο. Στο κάτω μέρος της οθόνης της φορητής συσκευής φαίνονται και πάλι οι δυνατότητες περαιτέρω πλοήγησης. Το User Profile χρησιμοποιείται από το σύστημα για διαχείριση προσωπικών στοιχείων σε σχέση με τη χρηστικότητα της εφαρμογής.

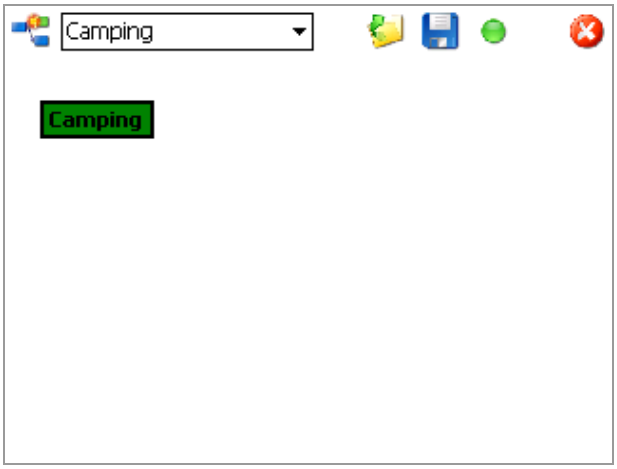


Στη συνέχεια, ο χρήστης πλοηγείται, μέσω ενός Interface, στις έννοιες του domain που τον ενδιαφέρει. Επίσης, μέσω αυτού του γραφικού περιβάλλοντος μπορεί να αποθηκεύσει τα concepts στο τοπικά στη φορητή συσκευή του (PDA) ή να ενημερώσει / επικαιροποιήσει τα concepts του από τον server του.

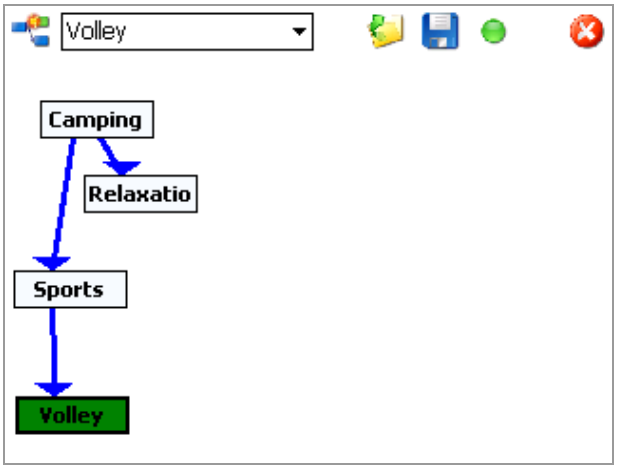


Στη διπλανή οθόνη φαίνεται η λειτουργικότητα της δημιουργίας προσωπικών οντολογιών υπό την μορφή γράφων, τις οποίες δημιουργεί ο χρήστης μέσω ενός γραφικού περιβάλλοντος, απεικόνισης εννοιών, τύπου Graph Editor.

Το περιβάλλον αυτό είναι προσβάσιμο στο χρήστη της φορητής συσκευής, μέσω της κλήσης του αντίστοιχου tab του γενικού περιβάλλοντος διεπαφών.

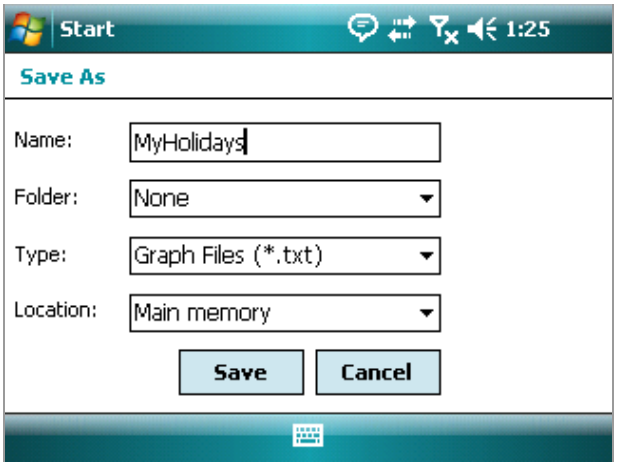


Στη διπλανή εικόνα φαίνεται το κυρίως περιβάλλον του Graph Editor. Η οθόνη του PDA αλλάζει προσανατολισμό διευκολύνοντας την εργασία αυτή. Ο χρήστης μπορεί με τα εργαλεία του περιβάλλοντος αυτού να δημιουργήσει να διαγράψει, μετακινήσει, κόμβους κλπ.

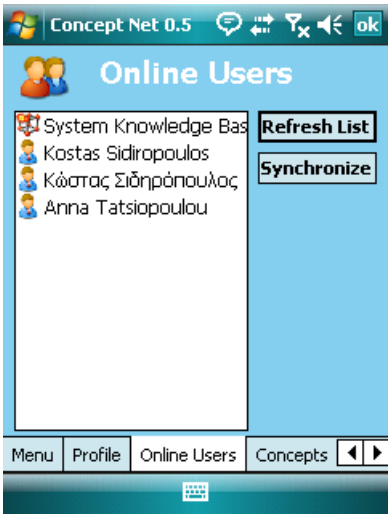


Στη συνέχεια ο Graph Editor, του δίνει τη δυνατότητα να συνδέσει τις έννοιες – κόμβους και να δημιουργήσει γράφους οντολογιών.

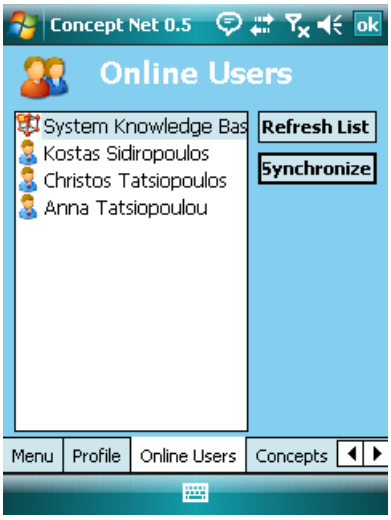
Επίσης, φορτώνοντας γράφους τοπικά σωμένους, μπορεί να επεξεργάζεται αυτούς και σε κατάσταση off-line.



Το περιβάλλον αυτό επιτρέπει την πλήρη διαχείριση των γράφων, όπως φόρτωση, σώσιμο τοπικά στη φορητή συσκευή κλπ. Το περιβάλλον αυτό έχει πρόσβαση στο λειτουργικό σύστημα και μέσω αυτού δίνει διάφορες δυνατότητες στο χρήστη.



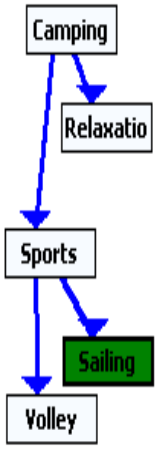
Στη συνέχεια ο χρήστης μέσω της φορητής συσκευής του, συνδέεται σε πραγματικό χρόνο μέσω server, με το δίκτυο χρηστών. Έχει τη δυνατότητα επιλογής ενός εξ αυτών, με τον οποίο θα πραγματοποιηθεί η διαδικασία ανταλλαγής γνώσης μέσω των προσωπικών τους οντολογιών.



Ο χρήστης επίσης, μπορεί να επιλέξει τον “System Knowledge Based User”, για ανταλλαγή γνώσης, με την γενική οντολογία του συστήματος – δικτύου.

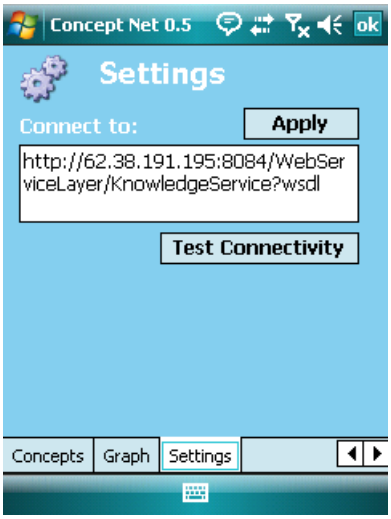
Τα αποτελέσματα της διαδικασίας ανταλλαγής γνώσης, εμφανίζονται στο Graph Editor του καθενός χρήστη, σαν νέα γνώση, με τη μορφή νέων κόμβων.





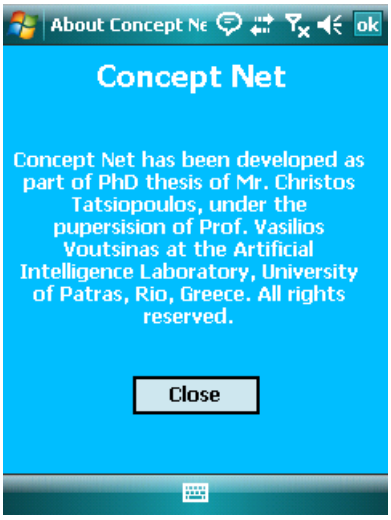
Οι κόμβοι συνδέονται με ήδη υπάρχοντες κόμβους έννοιες, και τους οποίους, προτείνει αυτόματα το σύστημα σαν πιο «όμοιους» με αυτούς.

Στο συγκεκριμένο παράδειγμα, το σύστημα πρότεινε σαν νέα γνώση – έννοια τον κόμβο “Sailing”, προτείνοντας να συνδεθεί με την γενικώτερη έννοια “Sports” Ο χρήστης διατηρεί μέσω του γραφικού περιβάλλοντος της αποδοχής ή μη των προτάσεων του συστήματος.



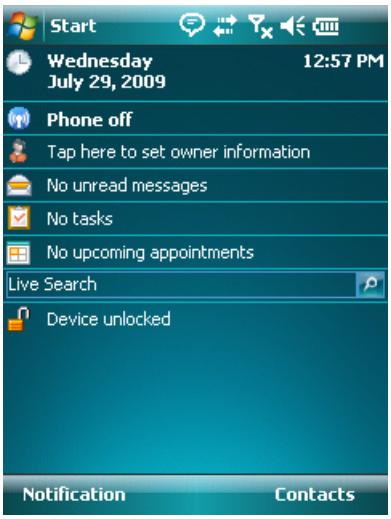
Το γραφικό περιβάλλον δίνει επίσης τη δυνατότητα σύνδεσης σε διαφορετικούς servers οι οποίοι τρέχουν την εφαρμογή, μέσω εισαγωγής της αντίστοιχης IP / port διεύθυνσης.

Επίσης μέσω συγκεκριμένου interface , γίνεται ο έλεγχος διαθεσιμότητας της υπηρεσίας αυτής από τον συγκεκριμένο server.



Στην διπλανή εικόνα εμφανίζεται η οθόνη “About” που εμφανίζει γενικές πληροφορίες για την client εφαρμογή που αναπτύχθηκε και τρέχει σε PDAs, την Concept Net.

Το λειτουργικό σύστημα που επιλέχτηκε, είναι το πιο διαδεδομένο σε φορητές συσκευές PDA και σε κινητά τηλέφωνα υψηλών δυνατοτήτων (smartphones).



Στη διπλανή εικόνα παρουσιάζεται η γενική άποψη του αυτού του λειτουργικού συστήματος, το Microsoft Windows Mobile Edition 6.0, στο οποίο αναπτύχθηκε η εφαρμογή Concept Net, στα πλαίσια της παρούσης διδακτορικής διατριβής.

Λόγω της ευρείας αποδοχής της, προσδίδει πολύ μεγάλες δυνατότητες υιοθέτησής της, τόσο από την υπάρχουσα αλλά και από την μελλοντική βάση χρηστών τέτοιου είδους φορητών συσκευών.

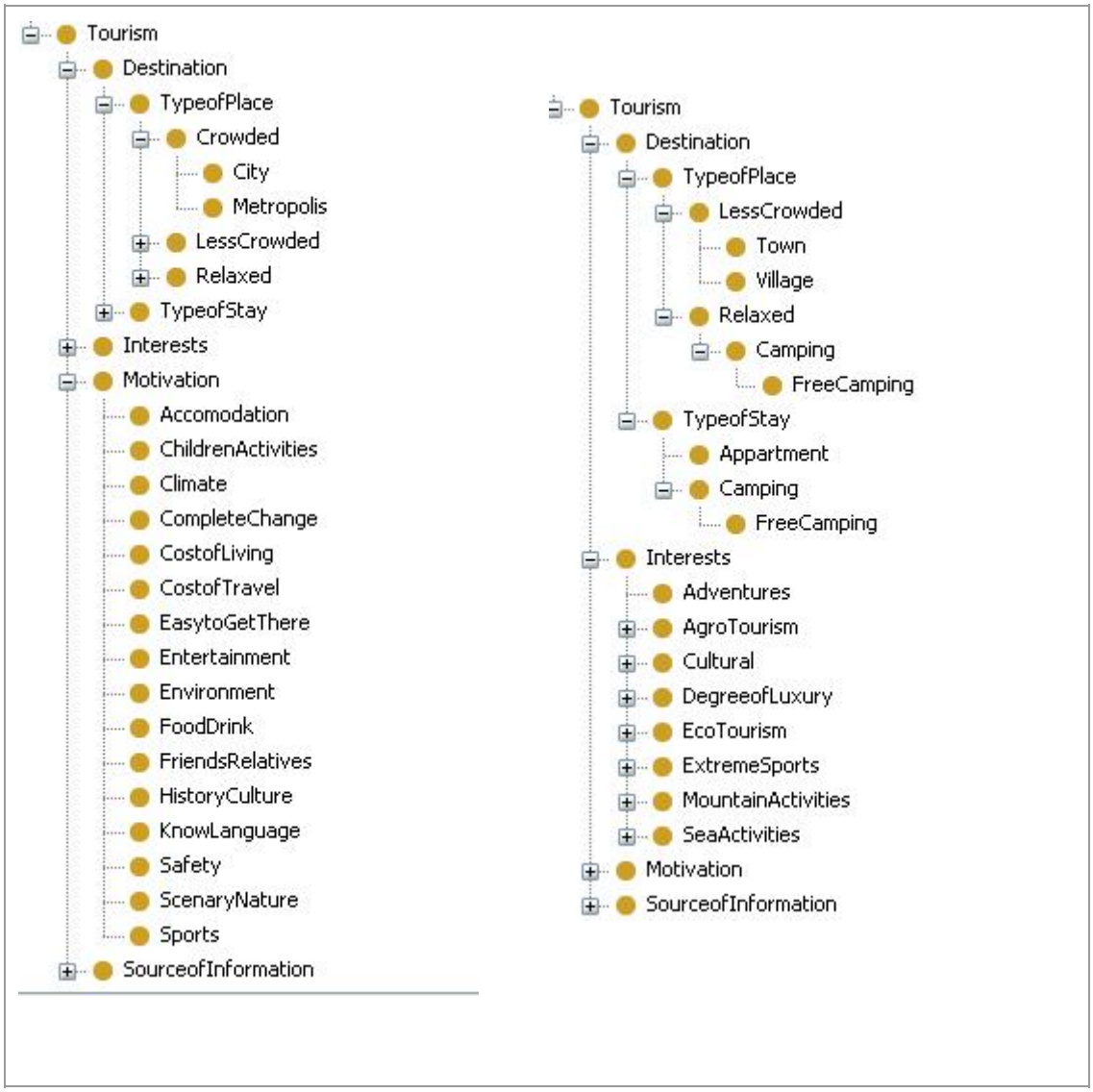
4.1.6. Έλεγχος Ορθης Λειτουργιας Συστηματος (Evaluation)

Η αρχιτεκτονική του Concept Net έχει υλοποιηθεί για απόδειξη των σκοπών των εννοιών, ως ένα πλήρες λειτουργικό πρωτότυπου. Ως πεδίο γνώσης, έχει επιλεγεί το πεδίο των δραστηριοτήτων που είναι σχετικές με τον τουρισμό.

Ο στόχος για το πρωτότυπο του συστήματος ήταν να αναγνωρίζει ανάμεσα σε δύο χρήστες του Concept Net, τους προσωπικούς οντολογικούς, να τους αναλύει χρησιμοποιώντας τη μεθοδολογία ONARM και τελικά να παρουσιάσει τις έννοιες που βρήκε πανομοιότυπες ή παρόμοιες ανά χρήστη.

Αυτός ο τρόπος, σε μια μεταγενέστερη έκδοση του συστήματος, όπως ήδη σχεδιάζεται και παραμένει να ολοκληρωθεί ως ένα λειτουργικό συστατικό, θα είναι η δυναμική και ευφυής ικανότητα του συστήματος να μπορεί να περάσει σχετικό ενδιαφέρον υλικό, τέτοιο όπως δυνητική πληροφορία marketing, διαφήμισης, μηνυμάτων, εστιασμένη σε συγκεκριμένες ομάδες-στόχους.

Στην επόμενη εικόνα δείχνουμε μέρη 2 προσωπικών οντολογιών κάποιων χρηστών, όπως εμφανίζονται στο PROTÉGÉ, οι οποίες έχουν σχεδιαστεί για το δοκιμαστικό τρέξιμο του συστήματος.



Μετά το τρέξιμο του συστήματος, τα αποτελέσματα που λάβαμε ήταν του τύπου:

```
[Onto1]:Tourism -> [Onto2]:Tourism 9,68 [A]
[Onto1]:Motivation -> [Onto2]:SourceofInformation 15,69 [A]
[Onto1]:Interests -> [Onto2]:Interests 9,52 [A]
[Onto1]:Motivation -> [Onto2]:Motivation 30,85 [A]
[Onto1]:Motivation -> [Onto2]:Destination 7,23 [A]
[Onto1]:SeaActivities -> [Onto2]:SeaActivities 18,27 [A]
[Onto1]:Cultural -> [Onto2]:Cultural 10,67 [A]
[Onto1]:Archeological -> [Onto2]:Archeological 8,89 [A]
[Onto1]:Motivation -> [Onto2]:Modern 7,64 [A]
[Onto1]:Events -> [Onto2]:Events 8,33 [A]
[Onto1]:Motivation -> [Onto2]:Exchibitions 7,84 [A]
[Onto1]:Motivation -> [Onto2]:ProductExchibition 11,77 [A]
[Onto1]:TypeofPlace -> [Onto2]:TypeofPlace 11,57 [A]
[Onto1]:LessCrowded -> [Onto2]:LessCrowded 0.00 [L]
[Onto1]:Handicrafts -> [Onto2]:Handicrafts 0.00 [L]
```

```
[Onto1]:Theater -> [Onto2]:Theater 0.00 [L]
[Onto1]:Arcitecture -> [Onto2]:Arcitecture 0.00 [L]
[Onto1]:CostofLiving -> [Onto2]:CostofLiving 0.00 [L]
[Onto1]:AgroTourism -> [Onto2]:AgroTourism 0.00 [L]
[Onto1]:Adventures -> [Onto2]:Adventures 0.00 [L]
[…… More Concepts Follow ……]
```

Από τα παραπάνω αποτελέσματα (μέρος της πλήρους λίστας) παρουσιάζεται ότι το σύστημα του Concept Net βρίσκει αυτές τις συσχετίσεις ανάμεσα στις λίστες εννοιών των 2 χρηστών με δύο τρόπους.

1. Ο δομημένος τρόπος που βασίζεται πλήρως στις επεξεργασίες των γράφων των 2 οντολογιών και
2. Ο δεύτερος που βασίζεται και σε γλωσσολογική προσέγγιση.

Οι δύο διαδικασίες δηλώνονται με [A] και [L] στην παραπάνω σχετική λίστα εννοιών. Για παράδειγμα, η έννοια Motivation από τη λίστα κάποιου χρήστη μπορεί πολλαπλώς να συσχετιστεί με τις έννοιες Motivation, Destination, Modern, Exchibitions, ProductExchibition σε ενός άλλου χρήστη τη λίστα, σημειούμενη με την αντίστοιχη σχέση, για περαιτέρω συσχέτιση και προώθηση πληροφορίας του συστήματος και χρήστη.

Το σύστημα έχει την δυνατότητα παραμετροποίησης, κατά τη διάρκεια της χρήσης του, έτσι ώστε να αναπαραστήσει διάφορα επίπεδα περιορισμών με όρους παραμέτρων υποστήριξης και εμπιστοσύνης. Αυτές οι παράμετροι επιτρέπουν στους αλγόριθμους συσχέτισης να επιστρέφουν περισσότερο ή λιγότερο σχετικά αποτελέσματα, έτσι ώστε η επικοινωνία, η παρουσίαση και η χρήση στις φορητές συσκευές να είναι αποτελεσματικές και διαχειρίσιμες.

Το σύστημα του Concept Net έχει επιτυχώς υλοποιηθεί, μετά από συνεχή εφαρμογή διαφόρων σεναρίων χρήσης, ως ένα λειτουργικό πρωτότυπο για την μεθοδολογία ONARM, επιστρέφοντας σε κάθε περίπτωση γνώση πρόσθετης αξίας στον εκάστοτε χρήστη, για την οποία είναι ενδιαφερόμενος, και της οποίας την ύπαρξη αγνοούσε.

Τα μελλοντικά σχέδια για την εφαρμογή Concept Net περιλαμβάνουν την ολοκλήρωση μιας υπομονάδας (module) για να περνά τη σχετική πληροφορία (διαφήμιση και λοιπά εργαλεία marketing) στο χρήστη, με αυτοματοποιημένο τρόπο.

Επιπλέον, περιλαμβάνει σχέδια για μελλοντικά τεστ και αναθεώρηση μιας αρχιτεκτονικής Concept Net με την ικανότητα υποστήριξης πολλαπλών συσκευών πελατών ταυτόχρονα. Σε αυτά τα πλαίσια θα εξεταστεί η εφαρμογή μιας κατανεμημένης αρχιτεκτονικής. Επίσης, σχετικά με το επικοινωνιακό κομμάτι, θα εξεταστεί η δυνατότητα ανάπτυξης modules για τεχνολογία Bluetooth η / και Wifi τεχνολογίες, με σκοπό την location-based αναγνώριση των χρηστών, σε τοπικά υποδίκτυα.

4.2. ΠΑΡΟΥΣΙΑΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ & ΔΙΑΔΙΚΑΣΙΩΝ ΕΦΑΡΜΟΓΗΣ

Στο παρακάτω απόσπασμα, παρουσιάζονται τα αποτελέσματα της διαδικασίας κατά την οποία πραγματοποιείται η συγχώνευση (merging) νέων κόμβων – εννοιών μιας reference Οντολογίας και μιας προσωπικής οντολογίας. Το απόσπασμα αυτό δημιουργείται σε πραγματικό χρόνο στη κονσόλα του backend συστήματος, δείχνοντας σε ποιό σημείο της διαδικασίας βρισκόμαστε σε κάθε χρονική στιγμή.

```
[KnowledgeService - SettingsUtility]: Reading Settings File...
[KnowledgeService - WS]: are up and running..
Oct 1, 2009 12:46:32 PM com.sun.xml.ws.transport.http.servlet.WSServletDelegate <init>
INFO: WSSERVLET14: JAX-WS servlet initializing
Oct 1, 2009 12:46:32 PM org.apache.coyote.http11.Http11Protocol start
INFO: Starting Coyote HTTP/1.1 on http-8084
Oct 1, 2009 12:46:33 PM org.apache.jk.common.ChannelSocket init
INFO: JK: ajp13 listening on /0.0.0.0:8009
Oct 1, 2009 12:46:33 PM org.apache.jk.server.JkMain start
INFO: Jk running ID=0 time=0/78  config=null
Oct 1, 2009 12:46:33 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 4407 ms
Oct 1, 2009 12:46:33 PM org.apache.catalina.core.StandardContext reload
INFO: Reloading this Context has started
Oct 1, 2009 12:46:33 PM com.sun.xml.ws.transport.http.servlet.WSServletDelegate destroy
INFO: WSSERVLET15: JAX-WS servlet destroyed
Oct 1, 2009 12:46:33 PM com.sun.xml.ws.transport.http.servlet.WSServletContextListener contextDestroyed
INFO: WSSERVLET13: JAX-WS context listener destroyed
Oct 1, 2009 12:46:34 PM com.sun.xml.ws.transport.http.servlet.WSServletContextListener contextInitialized
INFO: WSSERVLET12: JAX-WS context listener initializing
[KnowledgeService - SettingsUtility]: Reading Settings File...
[KnowledgeService - WS]: are up and running..
Oct 1, 2009 12:46:36 PM com.sun.xml.ws.transport.http.servlet.WSServletDelegate <init>
INFO: WSSERVLET14: JAX-WS servlet initializing
Oct 1, 2009 12:47:10 PM com.sun.xml.ws.transport.http.servlet.WSServletDelegate destroy
INFO: WSSERVLET15: JAX-WS servlet destroyed
Oct 1, 2009 12:47:10 PM com.sun.xml.ws.transport.http.servlet.WSServletContextListener contextDestroyed
INFO: WSSERVLET13: JAX-WS context listener destroyed
Oct 1, 2009 12:47:11 PM org.apache.catalina.startup.HostConfig checkResources
INFO: Undeploying context [/WebServiceLayer]
Oct 1, 2009 12:47:46 PM com.sun.xml.ws.transport.http.servlet.WSServletContextListener contextInitialized
INFO: WSSERVLET12: JAX-WS context listener initializing
[KnowledgeService - SettingsUtility]: Reading Settings File...
[KnowledgeService - WS]: are up and running..
Oct 1, 2009 12:47:50 PM com.sun.xml.ws.transport.http.servlet.WSServletDelegate <init>
INFO: WSSERVLET14: JAX-WS servlet initializing
```

```
Oct 1, 2009 12:47:50 PM org.apache.catalina.core.StandardContext start
INFO: Container org.apache.catalina.core.ContainerBase.[Catalina].[localhost].[/WebServiceLayer] has
already been started


[KnowledgeService - WS]: LoginUser operation - input:
[KnowledgeService - WS]:                          Username: ctatsio
[KnowledgeService - WS]:                          Password: ctatsio
[KnowledgeService - DBUtility]: Connected To Database:
localhost:3306/knowledgedb?useUnicode=true&characterEncoding=UTF-8
[KnowledgeService - DBUtility]:  >> 0 Tatsiopoulos
[KnowledgeService - DBUtility]: Retrieved 1 user(s)
[KnowledgeService - DBUtility]: Connection Closed


[KnowledgeService - WS]: UpdateOnlineStatus operation - input:
[KnowledgeService - WS]:                          UserID: 3
[KnowledgeService - WS]:                          New Status: 1
[KnowledgeService - DBUtility]: Connected To Database:
localhost:3306/knowledgedb?useUnicode=true&characterEncoding=UTF-8
[KnowledgeService - DBUtility]: User Updated into Database -  Updated Rows: 1
[KnowledgeService - DBUtility]: Connection Closed


[KnowledgeService - WS]: GetConceptsList operation
[KnowledgeService - DBUtility]: Connected To Database:
localhost:3306/knowledgedb?useUnicode=true&characterEncoding=UTF-8
[KnowledgeService - DBUtility]: Retrieved 89 concepts(s)
[KnowledgeService - DBUtility]: Connection Closed


[KnowledgeService - WS]: GetUsersByStatus operation
[KnowledgeService - DBUtility]: Connected To Database:
localhost:3306/knowledgedb?useUnicode=true&characterEncoding=UTF-8
[KnowledgeService - DBUtility]: Retrieved 3 user(s)
[KnowledgeService - DBUtility]: Connection Closed



[KnowledgeService - WS]: UploadUserGraph operation into: c:\\GraphsRepository - input:
[KnowledgeService - WS]:                          UserID: 3
[KnowledgeService - WS]:                          Graph Name: Graph
[KnowledgeService - WS]:                          File Content:
C:25:74
C:25:88
C:88:96
P:25:[15:20]
P:74:[70:64]
P:88:[2:66]
P:96:[8:115]

[KnowledgeService - WS]:                          Min. Support: 15
[KnowledgeService - WS]:                          Min. Confidence: 20
[KnowledgeService - WS]: Getting Concepts List From DB...
[KnowledgeService - DBUtility]: Connected To Database:
localhost:3306/knowledgedb?useUnicode=true&characterEncoding=UTF-8
[KnowledgeService - DBUtility]: Retrieved 89 concepts(s)
[KnowledgeService - DBUtility]: Connection Closed
[KnowledgeService - WS]: Commencing 1st phase of analysis ...
@ Initiating Ontology Analysis @
Reification Style: Standard
Node: Volley
Node: Camping
Node: Relaxation
Node: Sports
Opening: Camping
Child of Camping: Sports
Child of Camping: Relaxation
Child of Sports: Volley
---  Iterating Hash Map  ---
Camping - 1
Relaxation - 3
Volley - 4
Sports - 2
Leaf Reached - Volley. Adding Path to List...
Leaf Reached - Relaxation. Adding Path to List...
 - Camping - Sports - Volley
 - Camping - Relaxation
--- Translating Paths to Mapped Numbers
--- Translating Leaves to Mapped Numbers
1, 2, 4,
1, 3,
 ---   Remapping Nodes  ---
---------  Header  ----------
1        2        3        4
Camping        Sports        Relaxation        Volley
0        1        2        3
```

```
---------------------------
---  Setting Matrix  ---
-----------------------------------------
Writing results to: c:\\GraphsRepository\3_Graph_Leaves.txt
-----------------------------------------
Volley : 4
Relaxation : 3
-----------------------------------------
Writing Header to: c:\\GraphsRepository\3_Graph_Header.txt
-----------------------------------------
@ Ontology Analysis Terminated @


@ Initiating Apriori Analysis @
Starting Apriori Process on [ c:\\GraphsRepository\3_Graph_Header.txt ]....
=========  Configuration ============
Number of Transactions: 2
Minimum Support(%)    : 15.0
Minimum Confidence(%) : 20.0
====================================

-> Starting Combinations of : [4] items by [2]
Merging ...


-> Starting Combinations of : [4] items by [3]
Merging ...

-----------------------------------------
Writing results to: c:\\GraphsRepository\3_Graph_AprioriResults.txt
-----------------------------------------
Finished Processing
@ Apriori Analysis Terminated @


[KnowledgeService - WS]: SynchronizeGraphs operation  - input:
[KnowledgeService - WS]:                          UserID: 3
[KnowledgeService - WS]:                          Reference UserID: Reference
[KnowledgeService - WS]: Getting Concepts List From DB...
[KnowledgeService - DBUtility]: Connected To Database:
localhost:3306/knowledgedb?useUnicode=true&characterEncoding=UTF-8
[KnowledgeService - DBUtility]: Retrieved 89 concepts(s)
[KnowledgeService - DBUtility]: Connection Closed
[KnowledgeService - WS]: Commencing 2nd phase of analysis ...
 -- DEBUG -- 3_Graph


@ Initiating ARC Analysis @
Starting ARC Process....
=========== ARC Configuration ===========
C = 1.5
Thresshold = 0.0
b = 1.5
Linguistiq Matching = false
=========================================
====== Reading Leaf Nodes File #1: c:\\GraphsRepository\Reference_Graph_Leaves.txt ======
== Reading Leaf Nodes File #2: c:\\GraphsRepository\3_Graph_Leaves.txt ===
== Reading file #1: c:\\GraphsRepository\Reference_Graph_AprioriResults.txt ==
== Reading file #2: c:\\GraphsRepository\3_Graph_AprioriResults.txt ===
== Reading mapping file #1: c:\\GraphsRepository\Reference_Graph_Header.txt ==
== Reading mapping file #2: c:\\GraphsRepository\3_Graph_Header.txt ===
====   Calculating ARC table  ==========
@ ARC Analysis Terminated @


@ Initiating Merging @
=============================================
 Reading Ontology File: c:\\GraphsRepository\Reference_Graph.rdfs
=============================================
 * Node Loaded: Crowded
 * Node Loaded: Arcitecture
 * Node Loaded: Recomendation
 * Node Loaded: OrganizedCamping
 * Node Loaded: Environment
 * Node Loaded: Internet
 * Node Loaded: Interests
 * Node Loaded: Entertainment
 * Node Loaded: SeaDiving
 * Node Loaded: WineTourism
 * Node Loaded: Monasteries
 * Node Loaded: Metropolis
 * Node Loaded: Medium
 * Node Loaded: LessCrowded
 * Node Loaded: EasytoGetThere
 * Node Loaded: FreeCamping
 * Node Loaded: CompleteChange
 * Node Loaded: MagazineNewspaper
```

```
 * Node Loaded: Town
 * Node Loaded: Camping
 * Node Loaded: AgroTourism
 * Node Loaded: Relaxation
 * Node Loaded: Village
 * Node Loaded: Gliding
 * Node Loaded: BioSystemsObservation
 * Node Loaded: DegreeofLuxury
 * Node Loaded: SeaShore
 * Node Loaded: Climate
 * Node Loaded: City
 * Node Loaded: Low
 * Node Loaded: Relaxed
 * Node Loaded: Adventures
 * Node Loaded: Ballet
 * Node Loaded: Modern
 * Node Loaded: Food
 * Node Loaded: Sports
 * Node Loaded: High
 * Node Loaded: SourceofInformation
 * Node Loaded: Ski
 * Node Loaded: NotofInterest
 * Node Loaded: CostofTravel
 * Node Loaded: Cruises
 * Node Loaded: ProductExchibition
 * Node Loaded: Advertisiment
 * Node Loaded: Tourism
 * Node Loaded: CostofLiving
 * Node Loaded: HarvestingTourism
 * Node Loaded: Events
 * Node Loaded: EcoTourism
 * Node Loaded: Ruins
 * Node Loaded: Fishing
 * Node Loaded: Other
 * Node Loaded: ExtremeSports
 * Node Loaded: Volley
 * Node Loaded: FoodDrink
 * Node Loaded: Museums
 * Node Loaded: Appartment
 * Node Loaded: SeaActivities
 * Node Loaded: Hotel
 * Node Loaded: TeamBuilding
 * Node Loaded: MountainActivities
 * Node Loaded: BeachSports
 * Node Loaded: Accomodation
 * Node Loaded: KnowLanguage
 * Node Loaded: TypeofStay
 * Node Loaded: Archeological
 * Node Loaded: RockClimbing
 * Node Loaded: ScenaryNature
 * Node Loaded: Buildings
 * Node Loaded: ChildrenActivities
 * Node Loaded: Cultural
 * Node Loaded: Concerts
 * Node Loaded: HistoryCulture
 * Node Loaded: Anniversaries
 * Node Loaded: Exchibitions
 * Node Loaded: Motivation
 * Node Loaded: TypeofPlace
 * Node Loaded: RopeResque
 * Node Loaded: Theater
 * Node Loaded: Safety
 * Node Loaded: Religious
 * Node Loaded: FriendsRelatives
 * Node Loaded: Book
 * Node Loaded: Destination
 * Node Loaded: Handicrafts
 * Node Loaded: Sailing
 * 86 Nodes Loaded
==========================================
 Reading Ontology File: c:\\GraphsRepository\3_Graph.rdfs
==========================================
 * Node Loaded: Volley
 * Node Loaded: Camping
 * Node Loaded: Relaxation
 * Node Loaded: Sports
 * 4 Nodes Loaded
@ Merging Terminated @
```

Κεφάλαιο 5. Διερεύνηση Πεδίων Εφαρμογής

5.1. Προσωποποιημενες Διαδικασιες Μαθησης

Ένα πεδίο εφαρμογής της παρούσης είναι αυτό που αφορά τις διαδικασίες προσωποποιημένου περιεχομένου μέσω profiling σε συστήματα e-Learning, με αντικειμενικό σκοπό την έξυπνη συσχέτιση και διανομή εκπαιδευτικού περιεχομένου και αναγκών εκπαιδευομένων μέσω προφίλ τους.

Το σύστημα Concept-Net, με το είναι ένα ευφυές σύστημα, είναι σε θέση όπως έχει παρουσιαστεί, να εντοπίζει τα ενδιαφέροντα των χρηστών του. Η δυνατότητά του αυτή μπορεί να εφαρμοσθεί σε ένα ηλεκτρονικό σύστημα μάθησης. Πιο συγκεκριμένα, χρησιμοποιούνται σήμερα συστήματα που υποστηρίζουν πρότυπα περιγραφής με μεταδεδομένα, των modules τα οποία αποτελούν είτε ένα σύνολο μαθημάτων (courses) είτε τμημάτων αυτών. Επίσης, υπάρχουν στα συστήματα αυτά και περιγεγραμμένες συσχετίσεις μεταξύ των υπαρχόντων προαπαιτούμενων, των πιθανών αναμενόμενων προσόντων τα οποία θα αποκτήσει μετά την παρακολούθησή τους ο χρήστης και πιθανών δοκιμασιών για τον έλεγχο αυτών.

Το ζήτημα που προκύπτει είναι ότι υπάρχει ανάγκη ύπαρξης συγκεκριμένου λογισμικού, το οποίο θα είναι σε θέση να διαλειτουργεί με τα προαναφερθέντα συστήματα μάθησης και ταυτόχρονα να είναι σε θέση να ομαδοποιεί τις γνωσιακές ανάγκες των χρηστών.

Σε αυτό το σημείο η προτεινόμενη μεθοδολογία ONARM, είναι σε θέση αφού περιγραφούν τα αντικείμενα ενδιαφέροντος για τον προς κατάρτιση η / και εκπαίδευση χρήστη, να συσχετίσει και να προτείνει:

(α) άλλους πιθανούς χρήστες οι οποίοι έχουν αντίστοιχες γνωσιακές ανάγκες,

(β) να προσδιορίσει με περισσότερο ακρίβεια τον γνωστικό χώρο και τα γνωστικά αντικείμενα τα οποία πρέπει να παρακολουθήσει ο ενδιαφερόμενος και

(γ) να δημιουργεί δυναμικά δίκτυα ανθρώπων – χρηστών του συστήματος, με αντικειμενικό σκοπό την αλληλοβοήθειά τους στην κατανόηση των παρομοίων αντικειμένων τους και πέρα από τις τυποποιημένες διαδικασίες μάθησης.

Είναι άξιο σημείωσης εδώ, ότι η γνώση που μεταφέρεται κατ΄ αυτόν τον τρόπο από χρήστη σε χρήστη, μέσω ενός dedicated Peer-To-Peer (p2p) δικτύου ομοίων ενδιαφερόντων, αναγκών,

επιπέδων και ικανοτήτων, θα αποτελεί μία υπηρεσία πολύ μεγάλης πρόσθετης αξίας για το σύνολο του συστήματος, η οποία δεν υποστηρίζεται από τρέχοντα συστήματα.

5.2. ΣΤΟΧΕΥΜΕΝΗ ΔΙΑΦΗΜΙΣΗ

Η εφαρμογή CONCEPT-NET που είναι βασισμένη στη μεθοδολογία ONARM, μπορεί να εφαρμοστεί και στην περίπτωση της στοχευμένης διαφήμισης. Ως γνωστόν, η στοχευμένη διαφήμιση είναι αυτή που επιτυγχάνει το μικρότερο κόστος στον διαφημιζόμενο, δεδομένου ότι απευθύνεται κατ' ευθείαν στο κοινό που προσδοκάται να αγοράσει το συγκεκριμένο προϊόν ή υπηρεσία. Η στοχευμένη διαφήμιση χρησιμοποιεί τη κατάτμηση με βάση χαρακτηριστικά του πιθανού αγοραστικού κοινού, με βάση χαρακτηριστικά όπως γεωγραφικά (περιφέρεια επαρχία, πόλη, κλίμα, κλπ), δημογραφικά χαρκτηριστικά (ηλικία, φύλο, οικογενειακή κατάσταση, εισόδημα, επάγγελμα, κοκ), ψυχολογικούς παράγοντες και προφίλ ως προς την αγορά κλπ.

Η προτεινόμενη εφαρμογή λειτουργεί ακριβώς σε αυτόν τον χώρο, δεδομένου ότι μπορούν να απεικονιστούν στις προσωπικές οντολογίες των χρηστών, τόσο οι προτιμήσεις τους που ορίζουν το προφίλ τους, όσο και συσχετίσεις μεταξύ προφίλ χρηστών και κεντρικού συστήματος, πάντα με την συγκατάθεσή στους.

Πρακτικά αυτό σημαίνει, ότι αν ένας χρήστης έχει μία δεδομένη προσωπική οντολογία – προφίλ, και βρίσκεται σε ένα εμπορικό χώρο το οποίο συνδρομητικά συμμετέχει στο σύστημα, ο χρήστης τότε δέχεται προσωποποιημένη πληροφόρηση - διαφήμιση για προϊόντα και υπηρεσίες που διατίθενται στο συγκεκριμένο εμπορικό χώρο (π.χ. mall, πολυκαταστήματα, κλπ) και έχουν σχέση τόσο με το προφίλ του αλλά και με τις ανάγκες του εκείνη τη στιγμή. Υποθέτοντας ότι ο χρήστης αυτός σε μια κατοπινή στιγμή ταξιδεύει και βρίσκεται σε έναν πολυσύχναστο χώρο αναμονής αεροδρομίου για παράδειγμα, το σύστημα ανιχνεύει άλλους παρόντες χρήστες παρομοίου προφίλ και αναγκών, και εφόσον έχει δοθεί η συγκατάθεση (permission) στις συσκευές από τους ιδιοκτήτες τους, αυτές αυτόνομα, επικοινωνούν αυτόματα και προωθείται η υπάρχουσα πληροφορία – διαφήμιση στον επόμενο χρήστη, επιτυγχάνοντας κατ΄ αυτόν τον τρόπο τον μέγιστο βαθμό στόχευσης και επιτυχίας στο διαφημιστικό μήνυμα που διακινείται και άρα την ελαχιατοποίηση του κόστους του.

Κεφάλαιο 6. Συμπεράσματα & Προτάσεις για Μελλοντική Έρευνα

Μέσω της παρούσας Δ.Δ. προτείνεται (α) μία μεθοδολογία για την εξαγωγή δεδομένων και συσχέτισής τους η οποία ονομάζεται μεθοδολογία ONARM. Επίσης, (β) υλοποιείται η ανωτέρω μεθοδολογία σε μία ολοκληρωμένη εφαρμογή λογισμικού την εφαρμογή Concept Net.

Η μεθοδολογία ONARM κινείται στο χώρο της έξυπνης εξόρυξης δεδομένων από τον τομέα της τεχνητής νοημοσύνης data mining. Ειδικότερα εξειδικεύεται σε δεδομένα με τη μορφή εννοιών, οι οποίες είναι απεικονισμένες σε δενδρικές μορφές – γράφους, με το σχήμα των οντολογιών. Η μεθοδολογία ONARM είναι σε θέση να αξιοποιήσει τέτοιου είδους προσωπικούς γράφους χρηστών και να τους συσχετίζει. Απώτερος σκοπός της συσχέτισης αυτής είναι εύρεση κοινών ενδιαφερόντων απεικονισμένων με τη μορφή εννοιών και κατόπιν η συσχέτιση τους με γνώση που βρίσκεται απεικονισμένη στο σύστημα σε σχέση με αυτό το περιεχόμενο. Κατ' αυτόν τον τρόπο, η μεθοδολογία ONARM εντάσσεται επίσης και στον ευρύτερο τομέα του Knowledge Acquisition, Knowledge Management και Knowledge Distribution. Ειδικότερα ως προς τη διανομή γνώσης γίνεται χρήση ευφυών τεχνικών όπως οι έξυπνοι ενδιάμεσοι (Intelligent Agents).

Η μεθοδολογία ONARM αποτελείται από 6 διακριτά στάδια. Το πρώτο στάδιο είναι η περιγραφή με μορφή γραφικής απεικόνισης μια οντολογίας η οποία απεικονίζει τα ενδιαφέροντα ενός χρήστη σε μία γνωστική περιοχή. Στο πρώτο στάδιο, η οντολογία επίσης διαβιβάζεται από το σύστημα ανάλυσης.

Στο δεύτερο στάδιο αναπτύσσεται η τρέχουσα οντολογία του χρήστη στην φυσική μνήμη του υπολογιστή. Σε αυτή τη φάση η οντολογία με τη μορφή δενδρικής μορφής αριθμείται με τον αλγόριθμο Breadth First Search (BFS). Ακολουθεί στο τρίτο στάδιο η εφαρμογή του αλγόριθμου Depth First Search (DFS) για να προκύψουν αποτελεσματικά αριθμημένοι οι κόμβοι και οι ιδιότητες τους.

Στο τέταρτο στάδιο εφαρμόζεται η τεχνική των κανόνων συσχέτισης με τη μορφή των αλγόριθμων Apriori. Σκοπός αυτού του σταδίου είναι να προκύψει η ισχυρότερη λίστα

κανόνων οι οποίοι αναπαριστούν τις πιο ενδιαφέρουσες συσχετίσεις μεταξύ κόμβων – εννοιών της δεδομένης οντολογίας.

Στο πέμπτο στάδιο, οι κόμβοι της οντολογίας οι οποίοι έχουν ονοματιστεί με αριθμούς από το δεύτερο στάδιο, επαναφέρονται στην αρχική τους ονοματολογία.

Στο έκτο στάδιο της μεθοδολογίας ONARM χρησιμοποιούνται δύο οντολογίες οι οποίες έχουν επεξεργαστεί μέχρι το $5^{ο}$ στάδιο και οι δύο, ανεξάρτητα. Κατ' αυτόν τον τρόπο εισάγονται δύο σετ (λίστες ισχυρών κανόνων) και με βάση τον κεντρικό αλγόριθμο της μεθοδολογίας ONARM που αναπτύχθηκε στα πλαίσια της παρουσίασης, προκύπτει σαν αποτέλεσμα η συσχέτιση κόμβων – εννοιών των δύο προσωπικών οντολογιών των δύο ανεξαρτήτων χρηστών.

Στα πλαίσια επίσης της παρουσίασης όπως προαναφέρθηκε υλοποιήθηκε η ανωτέρω περιγραφείσα μεθοδολογία ONARM. Η υλοποίηση αυτή βασίζεται σε ένα σύστημα λογισμικού το οποίο είναι της μορφής client – server. Ουσιαστικά, στα πλαίσια του συστήματος Concept Net θεωρούνται οι χρήστες οι οποίοι χρησιμοποιούν φορητές συσκευές και απεικονίζουν τα ενδιαφέροντά τους, με εύκολο τρόπο, με τη μορφή οντολογιών σαν δενδρικές μορφές εξειδίκευσης.

Το επόμενο στάδιο του συστήματος είναι ότι όταν το σύστημα βρει ότι υπάρχουν χρήστες που κινούνται στον ίδιο γνωσιακό χώρο, έχοντας παρόμοια η ίδια ενδιαφέροντα, τίθεται σε λειτουργία με τρόπο διαφανή ως προς τον χρήστη (transparent to the user) και του προωθεί με έξυπνο και δυναμικό τρόπο σχετιζόμενη γνώση που είναι καταγεγραμμένη στο σύστημα με τη μορφή εννοιών. Οι έννοιες αυτές προέρχονται είτε από άλλους συνδεδεμένους χρήστες εκείνη τη στιγμή, είτε καταγεγραμμένες στο Knowledge Repository της εφαρμογής.

Το σύστημα λογισμικού Concept Net αναπτύχθηκε για να αποδείξει την εφαρμοσιμότητα και την χρηστικότητα της προταθείσας μεθοδολογίας ONARM.

Στην υλοποίηση που έγινε στα πλαίσια της διατριβής εκτελέσθηκαν σενάρια σε περιεχόμενο με βάση τα ενδιαφέροντα ατόμων πάνω στον τομέα του τουρισμού. Η εφαρμογή εκτελέσθηκε με επιτυχία και αντηλλάγησαν σχετικά δεδομένα τα οποία δεν ήταν εμφανή στους ατομικούς χρήστες του συστήματος, αλλά τα οποία ήταν καταγεγραμμένα σε άλλους χρήστες του κεντρικού συστήματος.

Το λογισμικό Concept Net που αναπτύχθηκε υλοποιήθηκε με βάση πρότυπα ανοικτής αρχιτεκτονικής. Το κομμάτι της επικοινωνίας υλοποιήθηκε με βάση πρωτόκολλα Internet και ειδικότερα υπηρεσιών διαδικτύου που καλούνται με έξυπνο τρόπο (Intelligent WEB Services).

Στα πλαίσια του ελέγχου της μεθοδολογίας ONARM και προκειμένου να προκύψουν έξυπνοι τρόποι διαχείρισης, συσχέτισης και προώθησης ενδιαφέροντος περιεχομένου ανά χρήστη, προτάθηκαν και υλοποιήθηκαν οκτώ (8) μέτρα Mi ενδιαφέροντος κόμβων. Ακολούθως προτάθηκε ένα γενικευμένο μοντέλο $f(wi, mi)$ με σκοπό την πρόβλεψη των γνωσιακών αναγκών του χρήστη έχοντας ως δεδομένο και μόνο τις γραφικές αναπαραστάσεις - γράφους των ενδιαφερόντων του. Το wi όπου i: 1,2... 8, ένα για κάθε προτεινόμενο μέτρο mi.

Εκτελέσθηκαν αναλυτικοί έλεγχοι (tests) σε οντολογίες με τελείως διαφορετικά χαρακτηριστικά (structural comparison), όπου σε ορισμένες από τις οποίες, εξευρέθηκε ότι ορισμένα Wi βάρη, ανεξαρτήτως περιεχομένου και μόνο λόγω της "γεωγραφικής" δομής των γράφων ελαχιστοποιούνται ως προς την συνδρομή τους, διευκολύνοντας κατά πολύ και μειώνοντας το υπολογιστικό κόστος της υλοποίησης της μεθοδολογίας συνολικά.

Η μεθοδολογία ONARM εξετάστηκε επίσης και σε οντολογίες τύπου benchmarking (test ontologies) που χρησιμοποιούνται για αυτό το σκοπό από τα ευρέως γνωστά sites OAEI και Gene Ontologies.

Το υπολογιστικό κόστος της προτεινόμενης μεθοδολογίας είναι το καλύτερο μεταξύ άλλων προτεινόμενων μεθοδολογιών.

Λόγω της ανοιχτής αρχιτεκτονικής που έχει υιοθετηθεί η υλοποίηση εύκολα μεταφέρεται και σε άλλους γνωσιακούς χώρους, αρκεί να αλλαχθεί η κεντρική οντολογία σε μόνον ένα (1) module του συστήματος. Ως επέκταση της μεθοδολογίας καθεαυτής προτείνεται να διερευνηθεί η παραλληλοποίηση της, δηλαδή η ανάπτυξη παράλληλων διεργασιών, σε διαδικασίες parallel processing για να αντιμετωπισθούν εμπόδια τύπου bottleneck από την ταυτόχρονη πολλαπλή χρήση του συστήματος, σε πολλά ταυτόχρονα γνωσιακά πεδία.

Επίσης, στον τομέα της επικοινωνίας των χρηστών προτείνεται η ένταξη modules τα οποία θα υποστηρίζουν την αυτόματη ανίχνευση χρηστών με βάση το proximity, δημιουργώντας έτσι location based γνωσιακά δίκτυα χρηστών.

Τέτοιου είδους άμεσα υλοποιήσιμα πρωτόκολλα είναι τα πρωτόκολλα Bluetooth, Wifi κλπ , των οποίων η υλοποίηση είναι πέραν των πλαισίων της παρούσης.

Συμπερασματικά, η παρούσα Δ.Δ. τόσο με την μεθοδολογία της σε θεωρητικό επίπεδο όσο και την υλοποίηση της σε πρακτικό επίπεδο, μέσω του λογισμικού Concept Net, προτείνει ένα καινούριο μοντέλο στην ανταλλαγή εννοιών, αυτή της "Αγοράς εννοιών" (Concept Marketplace), συνεισφέροντας κατ' αυτόν τον τρόπο, στη διαχείριση και διάχυση με αυτοματοποιημένο τρόπο, ανεξαρτήτως συγκεκριμένου περιεχομένου, της γνώσης μέσω της σημασιολογίας (semantics) και των δομών (graphs) αναπαράστασης αυτής. Μελλοντικά στον τομέα αυτό η παρούσα μεθοδολογία θα είχε μεγάλο ερευνητικό ενδιαφέρον να συσχετισθεί με περαιτέρω έρευνα στο τομέα των έξυπνων διαμεσολαβητών και ειδικώτερα αυτών που ονομάζονται autonomus trading agents. Σε αυτό τον τομέα έχει μεγάλο ερευνητικό ενδιαφέρον η κατά δυναμικό τρόπο, αυτοματοποιημένα και αυτόνομα από τον εκάστοτε χρήστη, εξεύρεση των καλύτερων οικονομικών προσφορών για αυτόν, εκτελώντας ανεξάρτητα ηλεκτρονικές διαμεσολαβήσεις για την επίτευξη των καλυτέρων οικονομικών προσφορών, σε πολλαπλές πλατφόρμες ηλεκτρονικού εμπορίου (negotiating trading agents in e-marketpaces), με άμεση εφαρμογή αυτού σε ηλεκτρονικές πλατφόρμες e-procurement (ηλεκτρονικών δημοπρασιών).

ΓΛΩΣΣΑΡΙ

Μεταπληροφορία, μεταδεδομένα: Τα μεταδεδομένα είναι δομημένα και κωδικοποιημένα δεδομένα τα οποία περιγράφουν χαρακτηριστικά πληροφοριακών οντοτήτων, αποσκοπώντας στην ταύτιση, αναγνώριση, ανακάλυψη, αξιολόγηση και διαχείριση των οντοτήτων που περιγράφονται.
Ant system: Συστήματα στον τομέα τεχνητής νοημοσύνης τα οποία προσπαθούν να μοντελοποιήσουν την ευφυΐα που χρησιμοποιούν οι οργανισμοί στη φύση, που βρίσκονται σε σμήνη (έντομα), ως προς την συνεργατικότητα για την εξεύρεση του ιδανικότερου μονοπατιού από τους χώρους τροφής έως την αποικία τους.
F-Measure: Στατιστικό μέτρο που χρησιμοποιείται στην περιοχή του information retrieval, και εξετάζει το precision και recall για την έρευνα, κατηγοριοποίηση και κατηγοριοποίηση ερωτήσεων και των αποδόσεών τους. Ο τύπος υπολογισμού είναι: $F = 2 \cdot (\mathrm{precision} \cdot \mathrm{recall})/(\mathrm{precision} + \mathrm{recall}).$
Reasoner (Protégé): Λογισμικό το οποίο είναι σε θέση να εξάγει λογικά συμπεράσματα, από ένα σύνολο δεδομένων ή αξιωμάτων.
RSS: Ο όρος RSS προέρχεται από το αγγλικό Really Simple Syndication. είναι ένα format ανταλλαγής περιεχομένου βασισμένο σε γλώσσα XML. Είναι ένας νέος τρόπος να ενημερώνεται ο χρήστης του Ίντερνετ για γεγονότα και νέα από άλλους χρήστες ή και κανάλια πληροφορίας. Η πληροφορία μέσω του RSS έρχεται στον υπολογιστή του χρήστη Online.
Swarm system, Swarm intelligence: Ανήκει στον τομέα της εξελικτικής υπολογιστικής (Evolutionary computation). Η ανάπτυξή τους προέκυψε από τη μελέτη των έμβιων οργανισμών και αφορούν σε έννοιες όπως του πληθυσμού, της μετάλλαξης και της φυσικής επιλογής (επιβίωση του πιο προσαρμοσμένου) για την ακριβέστερη επίλυση ενός προβλήματος. Οι μέθοδοι αυτοί μπορούν να διακριθούν περαιτέρω σε εξελικτικούς αλγόριθμους (evolutionary algorithms) και σε νοημοσύνης σμήνους (swarm intelligence), όπως πχ οι αλγόριθμοι που προσομοιάζουν τη συμπεριφορά μίας κοινωνίας μυρμηγκιών.

WSDL: Ο όρος WSDL προέρχεται από το Web Services Description Language και είναι μία γλώσσα τύπου XML, που επιτρέπει την περιγραφή ενός μοντέλου WEB services. Το πρότυπο προτάθηκε από τις εταιρίες IBM, Microsoft και Ariba και έχει γίνει αποδεκτό από το W3C.
XML: Προέρχεται από τα αρχικά των eXtensible Markup Language και είναι ένα σύνολο κανόνων βάσει των οποίων κωδικοποιούνται ηλεκτρονικά κείμενα. Μετά την υιοθέτησή του απο το W3C η χρήση του επεκτάθηκε και για την κωδικοποίηση άλλων δεδομένων. Το πρότυπο XML αποτελεί ένα από τα βασικότερα πρότυπα ανοικτού λογισμικού και έχει υιοθετηθεί από πολλές εμπορικές εφαρμογές, όπως το ευρέως διαδεδομένο MS-Office.
XML Schema: Αποτελεί πρότυπο του W3C σχετικά με τον τρόπο κωδικοποίησης σε XML, για αφηρημένες – γενικές συλλογές μταδεδομένων τα οποία αποτελούν τα στοιχεία του XML Schema. Τα αρχεία τα οποία ορίζουν XML Schema συνήθως έχουν κατάληξη .XSD, όπως ορίζεται στις προδιαγραφές RFC 3023 (http://tools.ietf.org/html/rfc3023)

ΒΙΒΛΙΟΓΡΑΦΙΑ

A. C. Ramos, J. Gensel, M. Villanova-Oliver, and H. Martin. (2005). Adapted information retrieval in web information systems using pumas. In Workshop on Agent-Oriented Information Systems (AOIS).

A. Doan, J. Madhavan, P Domingos, and A. Halevy, Learning to map between ontologies on the semantic web, Proceedings of the 11th International World Wide Web Conference (WWW 2002), 2002.

A. Firat and S. Madnick, Knowledge Integration to Ontological Heterogeneity: Challenges from Financial Information Systems, in 23rd International Conference on Information Systems (ICIS), Barcelona, Spain, 2001.

A. Gal, A. Segev, C. Tatsiopoulos, K. Sidiropoulos, and P. Georgiades: "Agent Oriented Data Integration". 7th International Bi-conference Workshop on Agent-Oriented Information Systems (AOIS-2005), pp. 98-108. Lecture Notes in Computer Science, Springer 2005.

A. M. Ouksel and A.P. Sheth. Semantic interoperability in global information systems: A brief introduction to the research area and the special section. SIGMOD Record, 28(1):5–12, March 1999.

A. Maedche, and S. Staab, Semi-automatic engineering of ontologies from texts, Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering (SEKE 2000), 2000, pp. 231-239.

A. Maedche, B. Motik, N. Silva, and R. Volz, MAFRA A MApping FRAmework for Distributed Ontologies, Lecture Notes in Computer Science, Knowledge Engineering and Knowledge Management: Ontologies and the Semantic Web, Volume 2473/2002, pp. 69-75, 2002 .

A. Pretschner, and S. Gauch, Ontology based personalized search, Proc. 11th IEEE Intl. Conf. on Tools with Artificial Intelligence, 1999, pp. 391-398.

A. Steth and S. Larson. Federated database schemas for managing distributed heterogenous and autonomous databases. ACM Computing Surveys, 23(3): 183-230, September 1990.

Agrawal R., Srikant R., (1994), Fast Algorithms for Mining Association Rules, IBM Research Center, Proceedings of the 20th VLDB Conference Santiago, Chile, 1994

Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., and Verkamo, A.I. Fast Discovery of Association Rules. In Proceedings of Advances in Knowledge Discovery and Data Mining. 1996, 307-328.

Alexander Maedche and Steffen Staab, Comparing Ontologies – Similarity Measures and a Comparison Study, March 2001, University of Karlsruhe.

B. Boutsinas, T. Papastergiou, On Clustering Tree Structured Data with Categorical Nature, Pattern Recognition, Elsevier Science, 41, 2008, pp. 3613– 3623.

B. Omelayenko, Integration of product ontologies for B2B marketplaces: a preview SIGecom Exch., 2(1), 2001.

Bock, J., Topor, R., Volz, R.: Ontology Merging using Answer Set Programming and Linguistic Knowledge. Proceedings of OAEI 2007, 316–320 (2007).

Boutsinas, B., Papastergiou, T.: On clustering tree structured data with categorical nature.Pattern Recognition, Elsevier Science, 41, 3613–3623, (2008).

Bundy A., Silver B. and Plummer D. (1985), An Analytical Comparison of Some Rule-Learning Programs, Artificial Intelligence 27, pp 137-181

C. Batini, M. Lenzerini, and S. Navathe. A comparative analysis of methodologies for database schema integration. ACM Computing Surveys, 18(4):323–364, December 1986.

C.A. Knoblock, S. Minton, J.L. Ambite, N. Ashish, I. Muslea, A. Philpot, and S. Tejada. The Ariadne approach to web-based information integration. International Journal of Cooperative Information Systems (IJCIS), 10(1-2):145–169, 2001.

Chalupksy, H.: OntoMorph: a translation system for symbolic knowledge. Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR-2000), (2000).

Chara Kostakis , Μέθοδοι Εισαγωγής και Επίδραση των Νέων Τεχνολογιών και της Πληροφορικής σε μονάδες Υγείας – Διδακτορική Διατριβή, Πανεπιστήμιο Πατρών, 2007.

Cheng Hian Goh. Representing and Reasoning, about Semantic Conflicts in Heterogeneous Information Sources. MIT, 1997.

Chinchor N., Robinson P. (1997) MUC-7 Information Extraction Task Definition (version 5.1). Message Understanding Conference Proceedings.

Christos Tatsiopoulos, Basilis Boutsinas, Konstantinos Sidiropoulos, On Aligning Interesting Parts of Ontologies, Proceedings of International Joint Conference on Knowledge Discovery and Ontology Development (KEOD), Madeira, Portugal, 6-8 Oct 2009, pp.: to be announced.

Clark, P, and Niblett, T.,The CN2 induction algorithm. Machine Learning. 3, pp.: 261-283, 1989.

Cunningham H. (2000) Information Extraction: a User Guide. Research Memo. University of Sheffield, 2000.

D. Aumueller, H.H. Do, S. Massmann, and E. Rahm, Schema and Ontology Matching with COMA++, In Proceedings of SIGMOD (Demonstration), 2005.

D. Barret, L. Clarke, P. Tarr, and A. Wise. A framework for event-based software integration. ACM Transactions on Software Engineering and Methodology, 5(4), 1996.

D.L. McGuinness, R. Fikes, J. Rice, S. Wilder, An Environment for Merging and Testing Large Ontologies, Proceedings of the Seventh International Conference on Principles of Knowledge, Representation and Reasoning (KR2000), Breckenridge, Colorado, 2000.

Doan, A., Madhavan, J., Domingos, P., Halevy, A.: Learning to map between ontologies on the semantic web. Proceedings of the 11th International World Wide Web Conference (WWW 2002), (2002).

E. Compatangelo, and H. Meisel, Reasonable support to knowledge sharing through schema analysis and articulation, Neural Computing & Applications, vol 12, pp 129-141, 2004.

E. Hovy, Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses, Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC), Granada, Spain,1998.

E. Rahm, and P.A. Bernstein, A Survey of Approaches to Automatic Schema Matching VLDB Journal, 10(4), 2001.

F. Chabris, Artificial Intelligence & Turbo Pascal, Heuristic Search pp 138-139, Christopher Dow Jones – Irwin, Homewwod, Illinois, 1987, ISBN: 0-87094-963-2

F. Giunchiglia, M. Yatskevich, P. Avesani, P. Shvaiko, A Large Scale Dataset for the Evaluation of Ontology Matching Systems, The Knowledge Engineering Review Journal (KER) issue 24, 2009.

F. Giunchiglia, P.Shvaiko, M. Yatskevich, S-Match: an Algorithm and an Implementation of Semantic Matching, In Proceedings of ESWS, LNCS 3053/2004, pp. 61-75, 2004.

Feigenbaum, E.A., Expert Systems in the 1980s., In Bond A. (ed.), State of the Art in Machine Intelligence. Pergamon- Infotech. Maidenhead, 1981.

G. L. Zuniga, Ontology: Its Transformation From Philosophy to Information Systems, FOIS'01, Ogunquit, Maine, USA, October 2001

G. Modica, A. Gal, and H. Jamil. The use of machine-generated ontologies in dynamic information seeking. In C. Batini, F. Giunchiglia, P. Giorgini, and M. Mecella, editors, Cooperative Information Systems, 9th International Conference, CoopIS 2001, Trento, Italy, September 5-7, 2001, Proceedings, volume 2172 of Lecture Notes in Computer Science, pages 433–448. Springer, 2001.

G. Stumme, and A. Maedche, Ontology Merging for Federated Ontologies on the Semantic Web, Proceedings of the International Workshop for Foundations of Models for Information Integration (FMII-2001), pp. 16-16, 2001.

Giunchiglia, F., Shvaiko, P., Yatskevich, M.: S-Match: an Algorithm and an Implementation of Semantic Matching. Proceedings of 1st European Semantic Web Symposium, (2004).

Gruber T., Toward principles for the design of ontologies used for knowledge sharing. International Journal of Human-Computer Studies, 43, pp.907-928, 1995

Guangfei Yang, Kaoru Shimada, Shingo Mabu, Kotaro Hirasawa and Jinglu Hu, (2007) IEEE Congress on Evolutionary Computation (CEC 2007)

H. Chalupksy, OntoMorph: a translation system for symbolic knowledge, Proceedings of the 17th International Conference on Knowledge Representation and Reasoning (KR-2000), 2000.

H. Wache, T. V¨ogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann, and S. Hubner, Ontology-Based Integration of Information -A Survey of Existing Approaches, in IJCAI Workshop: Ontologies and Information Sharing, Seattle, WA, 2001.

H. Wache, T. Vogele, U. Visser, H. Stuckenschmidt, G. Schuster, H. Neumann and S. Hubner, Ontology-Based Information Integration: A Survey, The BUSTER Project, Intelligent Systems Group, Center for Computing Technologies, Center for Computing Technologies, University of Bremen, 2001

Hovy, E.: Combining and Standardizing Large-Scale, Practical Ontologies for Machine Translation and Other Uses. Proceedings of the 1st International Conference on Language Resources and Evaluation (LREC), Granada, Spain, (1998).

http://www.hpl.hp.com/semweb/doc/tutorial/RDF_API/index.html

http://www.w3.org/DesignIssues/Principles.html .

http://www.w3.org/TR/vcard-rdf

J. Euzenat, P.Valtchev, Similarity-based Ontology Alignment in OWL-Lite, Proc. of ECAI, pp. 333-337, 2004.

J. Madhavan, P.A. Bernstein, and E. Rahm. Generic schema matching with Cupid. In Proceedings of the International conference on very Large Data Bases (VLDB), pages 49–58, Rome, Italy, September 2001.

J. R. Quinlan, (1987), Generating Production Rules from Decision Trees, Artificilal Intelligence Laboratory, MIT, Proceedings of the 10th IJCAI

J. S. Park, M. S. Chen, P. S. Yu, An Effective Hash-Based Algorithm for Mining ssociation rules, in Proceedings of ACM SIGMOD Int. Conf. Management of Data, 1995, pp. 175–186.

J. Tang, J. Li, B. Liang, X. Huang, Y. Li, K. Wang, Using Bayesian decision for ontology mapping Journal of Web Semantics, 4 (4), 2006, pp. 243-262.

J.Park, and S. Ram, Information Systems Interoperability: What Lies Beneath?, ACM Transactions on Information Systems (TOIS), Volume 22 , Issue 4 (October 2004), pp. 595 – 632, 2004.

Jurafsky D., Martin J.H. (2000) Speech and Language Processing. Prentice Hall. ISBN 0-13-0950-69-6.

Kalfoglou Y., Schorlemmer M., (2002), Ontology Mapping, The State of the Art

Kalfoglou, Y., Schorlemmer, M.: Information-flow-based ontology mapping. Lecture Notes in Computer Science, 2519, 1132-1151, (2002).

Kalfoglou, Y., Schorlemmer, M.: Ontology Mapping: the State of the Art. The Knowledge Engineering Review, 18(1), (2003).

Kiu, C.-C., Lee, C.S.: Ontology Mapping and Merging through OntoDNA for Learning Object Reusability. Educational Technology & Society, 9(3), 27–42, (2006).

Knuth, Donald E. (1997), The Art Of Computer Programming Vol. 1. 3rd ed., Boston: Addison-Wesley, ISBN 0-201-89683-4

L. Palopoli, L.G. Terracina, and D. Ursino. (2000) The system DIKE: Towards the semiautomatic synthesis of cooperative information systems and data warehouses. In PADBIS-DASFAA, pages 108–117.

M. Ehrig, and S. Staab, (2003) QOM Quick Ontology Mapping, Proceedings of the International Conference on Data Mining, ICDM-2003, IEEE Press .

M. Ehrig, and Y. Sure, (2005) Ontology Mapping - An Integrated Approach, Professional Knowledge Management , LNCS vol. 3782/2005, pp 560-569.

M. N. Huhns, and M.P. Singh, (1997) Ontologies for agents, IEEE Internet Computing 1(6), pp. 81-83.

M. S. Lacher, and G. Groh, , (2001) Facilitating the exchange of explicit knowledge through ontology mappings, Proceedings of the 14th International FLAIRS Conference.

MacQueen, J.B., (1967). Some Methods for Classification and Analysis of Multivariate Observations. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. Volume I (Statistics), pp.: 281-297.

McGuinness, D.L., Fikes, R., Rice, J., Wilder, S. (2000). An Environment for Merging and Testing Large Ontologies. Proceedings of the Seventh International Conference on Principles of Knowledge, Representation and Reasoning (KR2000), Breckenridge, Colorado, 483–493.

N. Choi, I.Y. Song, and H. Han, (2006). A Survey on Ontology Mapping, ACM SIGMOD Record archive, Volume 35 , Issue 3 (September 2006) , pp 34-31.

N. F. Noy, and M. Klein, Ontology evolution: Not the same as schema evolution, 2002, Also as: Smi-20020926, University of Stanford, Stanford Medical Informatics. Knowledge and Information Systems (In Press).

N. F. Noy, and M. Musen, (1999). SMART: Automated support for ontology merging and alignment, Proceedings of the 12th Workshop on Knowledge Acquisition, Modelling and Management (KAW99),

N. F. Noy, and M. Musen, (2000). PROMPT: Algorithm and tool for automated ontology merging and alignment, Proceedings of the 17th National Conference on Artificial Intelligence (AAAI00),.

N. F. Noy, and M. Musen, PROMPTDIFF: A fixed-point algorithm for comparing ontology versions, Proceedings of the 18th National Conference on Artificial Intelligence (AAAI02), 2002, pp. 744-751.

N. Guarino, Formal Ontology in Information Systems, Proceedings of the First International Confernece (Fois'98), June 6-8 Trento, Italy, by, International Conference on Principles of Knowledge Representation , 1998

Noy, N.F., Klein, M., (2004). Ontology evolution: not the same as schema evolution. Knowledge and Information Systems, 6(4), 428–440.

Noy, N.F., Musen, M. (2000). PROMPT: algorithm and tool for automated ontology merging and alignment. Proceedings of the 17th National Conference on Artificial Intelligence and 12th Conference on Innovative Applications of Artificial Intelligence, 450–455.

Noy, N.F., Musen, M. (2002). PROMPTDIFF: a fixed-point algorithm for comparing ontology versions. Proceedings of the 18th National Conference on Artificial Intelligence (AAAI02), 744-751.

P. Mitra, and G. Wiederhold, (2002) Resolving Terminological Heterogeneity In Ontologies, Proceedings of the ECAI02 workshop on Ontologies and Semantic Interoperability.

P. Shvaiko, and J. Euzenat, (2005) A Survey of Schema-based Matching Approaches, Journal on Data Semantics (JoDS), IV, LNCS 3730, pp. 146-171.

Prasad, S., Peng, Y., Finin, T. (2002). Using Explicit Information To Map Between Two Ontologies. Proceedings of the Workshop on Ontologies in Agent Systems, Bologna, Italy, Vol. 66.

R. J. Miller, M.A. Hern`andez, L.M. Haas, L.-L. Yan, C.T.H. Ho, R. Fagin, and L. Popa. (2001) The Clio project: Managing heterogeneity. SIGMOD Record, 30(1):78–83.

S. Bergamaschi, S. Castano, M. Vincini, and D. Beneventano. (2001) Semantic integration of heterogeneous information sources. Data & Knowledge Engineering, 36(3).

S. Brin, R. Motwani, J.D. Ullman, S. Tsur, (1997). Dynamic Itemset Counting and Implication Rules for Market Basket Data, in: Proceedings of ACM SIGMOD Int. Conf. Management of Data, , pp. 255–264.

S. C. Sundararajan, S. Sankarlal, and A. Kumar. Inca 2005. (Investor network collaborative architecture) a method in the madness of wall street. In Workshop on Agent-Oriented Information Systems (AOIS).

S. Prasad, Y. Peng, and T. Finin (2004). Using Explicit Information To Map Between Two Ontologies, International Conference on Autonomous Agents, Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 2 table of contents, New York, pp. 846 - 853 ,

S.Kaza, and H. Chen, (2008). Evaluating Ontology Mapping Techniques: An Experiment in Public Safety Information Sharing, Decision Support Systems archive, Volume 45 , Issue 4 (November 2008), pp. 714-728.

Safavian, S.R. and Landgrebe, D., (1991). A survey of decision tree classifier methodology. IEEE Trans. Systems, Man & Cybernetics. 21 (3), pp.: 660-674.

Schmitt, I., Saake, G. (1998). Merging inheritance hierarchies for database integration. Proceedings of CoopIS98, IEEE Computer Science Press, 322-331.

Shadbolt N., et. al., (2006), The Semantic WEB Revisited, Computer Magazine, IEEE Computer Society

Spiliopoulos, V., Valarakos, A.G., Vouros, G.A., Karkaletsis, V. (2007). Learning Subsumption Relations with CSR: A Classificationbased Method for the Alignment of Ontologies. Proceedings of OAEI 2007, 321–325.

Stumme, G., Maedche, A. (2001). Ontology Merging for Federated Ontologies on the Semantic Web. Proceedings of the International Workshop for Foundations of Models for Information Integration, Italy.

T. Berners-Lee, Weaving the Web, Harper, 1999.

Tang, J., Li, J., Liang, B., Huang, X., Li, Y., Wang, K. (2006). Using Bayesian decision for ontology mapping. Journal of Web Semantics, 4(4), 243-262.

Tatsiopoulos C., Boutsinas B., Automatic Knowledge Exchanging between Tourists via Mobile Devices, Submitted

Tatsiopoulos C., Boutsinas B., On Exchanging Ontologies, Submitted

Tatsiopoulos, C., Boutsinas, B. (2009). Ontology Mapping Based on Association Rule Mining, Proceedings of 11th International Conference on Enterprise Information Systems, vol 3, pp. 33-40, Milan, Italy.

U.M. Fayyad, G. Piatetsky-Shapiro and P.Smyth, (1996). Advances in Knowledge Discovery and Data Mining, AAAI Press/MIT Press.

Virtual Vineyards: http://www.inc.com/magazine/19960615/1966.html

W. Hu, N. Jian, Y. Qu, Y. Wang, GMO (2005). A graph matching for ontologies, Proceedings of K-CAPWorkshop on Integrating Ontologies, pp. 41-48.

W. Hu., and Y. Qu, Falcon-AO (2008). A practical ontology matching system, Web Semantics: Science, Services and Agents on the World Wide Web archive, Volume 6 , Issue 3 (September 2008), pp. 237-239, .

W. P. Birmingham, E. H. Durfee, T. Mullen, and M. P. Wellman (1995). The distributed agent architecture of the university of Michigan digital library. In AAAI Spring Symposium on Information Gathering.

Y. Kalfoglou and M. Schorlemmer, (2003). Ontology Mapping the State of the Art, The Knowledge Engineering Review 18(1).

Y. Kalfoglou, and M. Schorlemmer, (2002). Information-flow-based ontology mapping, Lecture Notes in Computer Science 2519, pp. 1132-1151.

Y. Kusumura, Y. Hijikata, and S. Nishida. (2004). Text mining agent for net auction. In ACM Symposium on Applied Computing.