



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ & ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Ανάλυση, Σύγκριση και Χρήση Οντολογιών Ανθρώπινης Συμπεριφοράς
σε Πραγματικά Δεδομένα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Χρήστου Κοντζίνου

Επιβλέπων: Ασκούνης Δημήτριος
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2016



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ & ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Ανάλυση, Σύγκριση και Χρήση Οντολογιών Ανθρώπινης Συμπεριφοράς
σε Πραγματικά Δεδομένα

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

Χρήστου Κοντζίνου

Επιβλέπων: Ασκούνης Δημήτριος
Αν. Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 24η Οκτωβρίου 2016.

Ασκούνης Δημήτριος
Καθηγητής

Ψαρράς Ιωάννης
Καθηγητής

Δούκας Χάρης
Καθηγητής

Αθήνα, Οκτώβριος 2016



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΗΛΕΚΤΡΙΚΩΝ ΒΙΟΜΗΧΑΝΙΚΩΝ ΔΙΑΤΑΞΕΩΝ & ΣΥΣΤΗΜΑΤΩΝ ΑΠΟΦΑΣΕΩΝ

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό.

Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

Χρήστος Κοντζίνος

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Ηλ.

Υπολογιστών

Copyright © 2016 – Με επιφύλαξη παντός δικαιώματος. All rights reserved

Περίληψη

Ο σκοπός της διπλωματικής εργασίας ήταν η ανάπτυξη οντολογίας που θα μπορεί να καλύψει τις ανάγκες του χώρου των σύγχρονων πηγών δεδομένων (π.χ. κοινωνικά δίκτυα, φορητές συσκευές, έξυπνα ρολόγια κτλ.). Ένας σύγχρονος χρήστης χρειάζεται μία εφαρμογή η οποία τον βοηθάει να καταγράψει τις καθημερινές του δραστηριότητες. Αν τα δεδομένα αναλυθούν και επιστραφούν στον χρήστη σε κατάλληλη μορφή, ο άνθρωπος είναι σε θέση να οργανώσει καλύτερα την καθημερινότητά του. Πράγματι υπάρχει πληθώρα εφαρμογών οι οποίες καταγράφουν και αναλύουν δραστηριότητες ωστόσο κάθε μία από αυτές ασχολείται με μία συγκεκριμένη κατηγορία δραστηριοτήτων. Δεν υπάρχει με άλλα λόγια ένα ενιαίο μοντέλο το οποίο να περιλαμβάνει το σύνολο των δραστηριοτήτων του σύγχρονου χρήστη με στόχο να τις καταγράψει, να τις κατανοήσει και να τις επαναχρησιμοποιήσει για να βγάλει χρήσιμα συμπεράσματα.

Συγκεκριμένα έγινε μελέτη της θεωρίας δραστηριοτήτων (activity theory) και ενός αριθμού οντολογιών σχετικών με την μοντελοποίηση και ανάλυση δεδομένων ανθρώπινης συμπεριφοράς. Από το σύνολο αυτών, επιλέχθηκαν οι X, Y, Z και συνδυάστηκαν σύμφωνα με την τεχνική των «τριών επιπέδων».

Η οντολογία που παράχθηκε, συνδέθηκε και δοκιμάστηκε με πραγματικά δεδομένα, τα οποία συλλέχθηκαν με χρήση σχετικής εφαρμογής, ενώ παράλληλα δοκιμάστηκε η δυνατότητα συλλογιστικής με τη χρήση αυτής της οντολογίας με SPARQL ερωτήματα. Η τελική οντολογία σχεδιάστηκε με χρήση του εργαλείου Protégé ενώ η σύνδεση της οντολογίας με τα πραγματικά δεδομένα έγινε χρησιμοποιώντας τη γλώσσα προγραμματισμού SQL η οποία έχει σχεδιαστεί για τη διαχείριση δεδομένων που βρίσκονται σε σχεσιακές βάσεις. Η μεταφορά των SPARQL ερωτήσεων στην εφαρμογή που αναπτύχθηκε προβάλλει πληροφορίες σχετικά με της δραστηριότητες ενός χρήστη και μπορεί να οδηγήσει σε χρήσιμα συμπεράσματα.

Τέλος, υλοποιήθηκε εφαρμογή χρησιμοποιώντας DJANGO (web framework) στην οποία μεταφέρθηκαν τα queries όπου φαίνονται τα συμπεράσματα των ερωτήσεων που έγιναν στην οντολογία μέσω της SPARQL. Η εφαρμογή αναπτύχθηκε για καλύτερη παρουσίαση των αποτελεσμάτων και τη διευκόλυνση ενός χρήστη ο οποίος θέλει να τα μελετήσει.

Συνολικά, η οντολογία που αναπτύχθηκε μπορεί να γίνει οδηγός και χρήσιμο εργαλείο για την συλλογή πραγματικών δεδομένων από φορητές συσκευές και κοινωνικά δίκτυα επιτρέποντας στους χρήστες να συνδεθούν με τις επαφές τους βάση των κοινών ενδιαφερόντων τους χωρίς να εμφανίζουν τα προσωπικά τους δεδομένα. Τέλος, είναι χρήσιμο το γεγονός ότι υπάρχουν ποικίλα εργαλεία τα οποία είναι συμβατά με τις τεχνολογίες που χρησιμοποιήθηκαν.

Λέξεις Κλειδιά: Θεωρία δραστηριοτήτων, σημασιολογικός ιστός, RDFs, SPARQL, κοινωνική δικτύωση, ανθρώπινες δραστηριότητες

Abstract

The scope of this thesis was the development of an ontology that can cover the needs of modern data sources (e.g. social networks, mobile devices, smart watches etc.). The modern user has need of an application which can help him track his everyday activities. If the data are analyzed and returned to the user in suitable form, the person is able to better organize his everyday life. Although there are many applications that record and analyze activities they deal with a specific category of activities. In other words no model exists that contains the total of a modern user's activities in order to record them, understand them and reuse them to reach useful conclusions. Finally, a web app was designed using DJANGO (web framework) in which the queries were transferred where one can see the results of the queries performed in the ontology using SPARQL.

Specifically, the activity theory was studied as well as a number of ontologies related to the modelling and analysis of human behavior data. Of those ontologies, a select number were chosen and concatenated according to the <<three level>> technique.

The designed ontology was tested with real data which were collected using a relevant application. At the same time SPARQL queries were used for reasoning purposes. The final ontology was designed using Protégé whereas the concatenation of the ontology with the real data was performed using the programming language SQL which was designed to handle data in relational databases. The transfer of the SPARQL queries in the web app shows information concerning the activities of a user and can lead to useful conclusions.

Finally, an application was developed using DJANGO (web framework), in which the queries were transferred. The application shows the conclusions of the queries performed in the ontology and was developed for the better presentation of the results and the facilitation of the user who wants to study them.

Overall, the final ontology can be used as a guide for collecting real data from mobile devices and social networks, allowing the users to connect to their contacts according to their common interests without showing their personal data. Finally, of use is the fact that there are many tools which are compatible with the technologies that were used.

Keywords : Activity theory, semantic web, RDFS, SPARQL, Social networking, Human activities

Ευχαριστίες

Με την ολοκλήρωση της διπλωματικής μου εργασίας αισθάνομαι την ιδιαίτερη υποχρέωση να εκφράσω τις θερμότερες ευχαριστίες μου στον επιβλέποντα και Καθηγητή του ΕΜΠ κ. Ασκούνη Δημήτριο, για την επίβλεψη της παρούσας διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Συστημάτων Αποφάσεων & Διοίκησης.

Στη συνέχεια, θα ήθελα να ευχαριστήσω ιδιαίτερω τον συνεργάτη του εργαστηρίου Συστημάτων Αποφάσεων & Διοίκησης και υποψήφιο Διδάκτωρ κ. Αλβέρτη Ιωσήφ για την εξαιρετη συνεργασία και καθοδήγηση σε όλη την πορεία της διπλωματικής μου εργασίας. Η βοήθεια και οι συμβουλές του ήταν καταλυτικής σημασίας για την επιτυχή ολοκλήρωσή της και ελπίζω πραγματικά να μου δοθεί ξανά η ευκαιρία συνεργασίας μαζί του στο μέλλον. Επίσης, θα ήθελα να ευχαριστήσω τον συνεργάτη του εργαστηρίου και υποψήφιο Διδάκτωρ κ. Αρβανιτάκη Ευάγγελο και για τη βοήθειά του σε τεχνικά ζητήματα και θέματα υλοποίησης.

Τέλος, θα ήθελα να ευχαριστήσω την οικογένεια και τους φίλους που με στήριξαν σε πολλά επίπεδα κατά την πολυετή μου πορεία στο Εθνικό Μετσόβιο Πολυτεχνείο, όπως και τους καθηγητές της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλ. Υπολογιστών που με καθοδήγησαν και με εισήγαγαν στον ευρύ και ξεχωριστό κόσμο του ηλεκτρολόγου μηχανικού.

Πίνακας Περιεχομένων

1	Εισαγωγή	14
1.1	Κοινωνικές Θεωρήσεις και Μοντελοποίηση Ανθρώπινων Δραστηριοτήτων	14
1.2	Αντικείμενο Διπλωματικής	15
1.2.1	Συνεισφορά	16
1.2.2	Μεθοδολογία	16
1.3	Οργάνωση Κειμένου	17
2	Θεωρητικό Υπόβαθρο	18
2.1	Θεωρία Δραστηριοτήτων (Activity Theory)	18
2.2	Σημασιολογία και Σημασιολογικός Ιστός	19
2.2.1	Σημασιολογικός Ιστός	19
2.2.2	Σημασιολογικά Μοντέλα	22
2.2.3	RDF , RDF Schema (RDFs) & OWL	24
2.2.4	SPARQL Query Language	29
3	Σχετικές Εργασίες	35
3.1	Κατανοώντας τα Ψηφιακά Αποτυπώματα: Μία Οντολογική Προσέγγιση της Θεωρίας Δραστηριοτήτων (Making Sense of Digital Traces: An Activity Theory Driven Ontological Approach)	35
3.2	Ανάλυση και Μοντελοποίηση Ανθρώπινων Δραστηριοτήτων στα Πληροφοριακά Συστήματα, και Ανάπτυξη Πρότυπης Εφαρμογής για Καταγραφή, Παρουσίαση και Παρακολούθηση Προσωπικών Δραστηριοτήτων	36
3.3	Οντολογία Θεωρίας Δραστηριοτήτων για Διαμοιρασμό Γνώσης σε Συστήματα Υγείας (Activity Theory Ontology for Knowledge Sharing in E-Health)	36
3.4	Χρησιμοποιώντας τη Θεωρία Δραστηριοτήτων για Μοντελοποίηση του Γενικού Πλαισίου Γνώσης (Using Activity Theory to Model Context Awareness)	37
4	Πειραματική Ανάλυση Δεδομένων	39
4.1	Επιλογή Οντολογιών	39
4.1.1	Υψηλό επίπεδο	39
4.1.2	Μεσαίο επίπεδο	40
4.1.3	Χαμηλό Επίπεδο	44
4.1.4	Μοντέλο Activity Tracker	45
4.2	Σύνδεση οντολογιών	46
4.2.1	Σύνδεση υψηλού-μεσαίου επιπέδου	46
4.2.2	Σύνδεση μεσαίου-χαμηλού επιπέδου	47
4.2.3	Σύνδεση Δεδομένων του Activity Tracker	48
4.3	SPARQL Queries	48

4.3.1	Λίστα δραστηριοτήτων/Χρήστη	48
4.3.2	Κοινές δραστηριότητες των δύο χρηστών	49
4.3.3	Συνολικές καταγραφές συγκεκριμένης δραστηριότητας/χρήστη	50
4.3.4	Συνολικές καταγραφές όλων των δραστηριοτήτων/χρήστη	51
4.3.5	Ποσοστό μίας δραστηριότητας επί των συνολικών καταγραφών	51
4.3.6	Μέσος όρος διάρκειας μίας δραστηριότητας.....	52
4.3.7	Compare activities	53
5	Ανάλυση και Σχεδίαση Συστήματος.....	55
5.1	Απαιτήσεις συστήματος	55
5.2	Αρχιτεκτονική συστήματος	55
5.3	Περιγραφή Λειτουργιών Υποσυστήματος	57
5.3.1	Εγγραφή Χρήστη στην Εφαρμογή με Συμπλήρωση Πεδίων	57
5.3.2	Σύνδεση και Αποσύνδεση του Χρήστη από το Σύστημα	57
5.3.3	Εμφάνιση των Χρηστών και Δραστηριοτήτων του Activity Tracker	57
5.3.4	Εμφάνιση των Χρηστών από την RDFS Βάση Δεδομένων	58
5.3.5	Εμφάνιση των Δεδομένων από τις SPARQL Ερωτήσεις	58
5.3.6	Δημιουργία Redirect Links στα Endpoint URLs του Dydra	58
5.4	Αρχιτεκτονική Συστήματος Πλαισίου DJANGO	58
5.4.1	Οθόνη Start.....	59
5.4.2	Οθόνη Register	60
5.4.3	Οθόνη Login.....	60
5.4.4	Οθόνη Logged in.....	61
5.4.5	Οθόνη Index	62
5.4.6	Οθόνη User Index	62
5.4.7	Οθόνη Semantic Users	63
5.4.8	Οθόνη Common Activities.....	64
5.4.9	Οθόνη Activity Entries	64
5.4.10	Οθόνη Activity Frequency	64
5.4.11	Οθόνη Activity Mean Time	65
5.4.12	Οθόνη Compare Activity.....	66
6	Υλοποίηση Συστήματος.....	67
6.1	Πλατφόρμες και Προγραμματιστικά Εργαλεία.....	67
6.2	Στιγμιότυπα και Καθοδήγηση	68
7	Επίλογος	77

7.1	Σύνοψη και Συμπεράσματα	77
7.2	Μελλοντικές Επεκτάσεις	78
8	Βιβλιογραφία.....	80
9	Παραρτήματα.....	81
9.1	Παράρτημα Ι:Οδηγίες εγκατάστασης	81

Πίνακας Εκθεμάτων

Εικόνα 1.1:Πρόβλημα χώρου διπλωματικής	15
Εικόνα 1.2: Μεθοδολογία διπλωματικής.....	17
Σχήμα 2.1:Μοντέλο Θεωρίας Δραστηριοτήτων(Engestrom 1987).....	19
Σχήμα 2.2:Σημασιολογικό Μοντέλο.....	23
Σχήμα 2.3:Παράδειγμα σημασιολογικού μοντέλου	24
Σχήμα 2.4:Παράδειγμα RDF γράφου	29
Εικόνα 3.1: Μοντέλο της δομής πληροφορίας πλαισίου (context) – Anders Kofod-Petersen, Jorg Cassens(2006). Using Activity Theory to Model Context Awareness	37
Σχήμα 4.1:Διάγραμμα του activity module της AMOn	40
Σχήμα 4.2:Διάγραμμα των κύριων κλάσεων της CoDAMoS	41
Σχήμα 4.3:Σχέσεις της κλάσης service στην CoDAMoS.....	41
Σχήμα 4.4:Σχέση πλατφόρμας-περιβάλλοντος στην CoDAMoS	42
Σχήμα 4.5:Γράφημα της οντολογίας activity streams 2.0.....	44
Σχήμα 5.1:Μοντέλο αρχιτεκτονικής εφαρμογής	56
Εικόνα 6.1:Στιγμιότυπο οθόνης start	68
Εικόνα 6.2:Στιγμιότυπο οθόνης register	68
Εικόνα 6.3:Στιγμιότυπο οθόνης login.....	69
Εικόνα 6.4:Στιγμιότυπο οθόνης invalid_login	69
Εικόνα 6.5:Στιγμιότυπο οθόνης loggedin.....	70
Εικόνα 6.6:Στιγμιότυπο οθόνης index.....	70
Εικόνα 6.7:Στιγμιότυπο οθόνης Activity details.....	71
Εικόνα 6.8:Στιγμιότυπο οθόνης Userindex	71
Εικόνα 6.9:Στιγμιότυπο οθόνης Userdetails	71
Εικόνα 6.10:Στιγμιότυπο οθόνης semantic users.	72
Εικόνα 6.11:Στιγμιότυπο οθόνης Common Activities	72
Εικόνα 6.12:Στιγμιότυπο από οθόνη του endpoint url του commonactivities.....	73
Εικόνα 6.13:Στιγμιότυπο οθόνης Running Entries	73
Εικόνα 6.14:Στιγμιότυπο οθόνης running frequency.....	74
Εικόνα 6.15:Στιγμιότυπο οθόνης running mean time.....	74
Εικόνα 6.16:Στιγμιότυπο οθόνης compare running.....	74
Εικόνα 6.17:Στιγμιότυπο οθόνης compare social media	75
Εικόνα 6.18:Στιγμιότυπο οθόνης status update	75
Εικόνα 6.19:Στιγμιότυπο οθόνης endpoint url του comparestatusupdate	75

1 Εισαγωγή

Στη σημερινή εποχή, με τη ραγδαία εξέλιξη της τεχνολογίας υπάρχει η δυνατότητα καταγραφής των ανθρώπινων δραστηριοτήτων από υπολογιστικά συστήματα και εφαρμογές μέσω της τεχνολογίας η οποία είναι γνωστή ως pervasive computing. Βασίζεται στην ενσωμάτωση μικροεπεξεργαστών σε καθημερινά αντικείμενα όπως για παράδειγμα ρολόγια και τηλέφωνα ώστε να μπορούν να ανταλλάσσουν πληροφορίες μεταξύ τους. Για να μπορέσουν αυτές οι εφαρμογές να επιστρέφουν χρήσιμες πληροφορίες σχετικά με τις δραστηριότητες των χρηστών κρίνεται σημαντική η κατανόηση της ανθρώπινης συμπεριφοράς από μηχανές η οποία μπορεί να οδηγήσει στην καλύτερη εμπειρία του χρήστη όταν χρησιμοποιεί pervasive υπολογιστικά εργαλεία και την αυξανόμενη χρησιμότητα αυτών.

1.1 Κοινωνικές Θεωρήσεις και Μοντελοποίηση Ανθρώπινων Δραστηριοτήτων

Τα τελευταία χρόνια έχει αναπτυχθεί πληθώρα οντολογιών οι οποίες είναι σχετικές με την μοντελοποίηση και ανάλυση δεδομένων σχετικών με την ανθρώπινη συμπεριφορά. Ωστόσο δεν έχουν αναλυθεί κάτω από το σωστό πρίσμα κοινωνιολογίας και ανθρωπολογίας ώστε να αποκαλύψουν επιπλέον διαστάσεις σε αυτά τα δεδομένα¹. Από όσο βλέπουμε στην ανάλυση των σχετικών οντολογιών που αναλύουν την πληροφορία πλαισίου, η διάσταση της ανθρώπινης δραστηριότητας στη μοντελοποίηση είναι ανεπαρκής. Χάνουμε επομένως πληροφορία που θα μπορούσε να συνεισφέρει σε πιο χρήσιμα συμπεράσματα. Ειδικότερα στην εποχή μας με την ανάπτυξη της τεχνολογίας και ειδικότερα των κινητών τελευταίας γενιάς σε συνδυασμό με ένα πλήθος εφαρμογών καταγραφής ανθρώπινων δραστηριοτήτων, παρουσιάζεται η ανάγκη δημιουργίας ενός γενικότερου πλαισίου πάνω στο οποίο μπορούν να μοντελοποιηθούν τα δεδομένα ενός σύγχρονου χρήστη των παραπάνω τεχνολογιών. Δεδομένου όμως ότι οι ανθρώπινες δραστηριότητες ποικίλουν εξαιρετικά τόσο σε κατηγοριοποίηση όσο και σε αριθμό οι περισσότερες εφαρμογές καταγραφής δεδομένων στηρίζονται πάνω σε μοντέλα συγκεκριμένα για μία κατηγορία δραστηριοτήτων. Για παράδειγμα υπάρχουν μοντέλα τα οποία περιγράφουν δραστηριότητες ενός ανθρώπου σχετικές με την άθληση, την υγεία, τα κοινωνικά δίκτυα κτλ. Αυτό σημαίνει ότι κάθε εφαρμογή μοντελοποιεί διαφορετικά τα δεδομένα έτσι ώστε να ανταποκρίνεται βέλτιστα στους στόχους της. Η έννοια λοιπόν και το μοντέλο της δραστηριότητας διαφέρει από εφαρμογή σε εφαρμογή δημιουργώντας πρόβλημα συσχέτισης σε περίπτωση που ο χρήστης επιθυμεί να ομαδοποιήσει τα δεδομένα του.

Οι περισσότερες εργασίες οι οποίες είναι σχετικές με την μοντελοποίηση μίας δραστηριότητας μέσω οντολογιών κάνουν αναφορά σε ένα συγκεκριμένο χώρο ανθρώπινων δραστηριοτήτων (υγεία, εργασία, οργάνωση κτλ.) και αρκούνται στην επιλογή κατάλληλης κοινωνικής θεώρησης για τη δημιουργία ενός σημασιολογικού μοντέλου, ικανού να περιγράψει το χώρο με τον οποίο ασχολείται η εργασία. Το πρόβλημα με εφαρμογές οι οποίες χρησιμοποιούν κάποια από τα παραπάνω μοντέλα είναι η διαφοροποίηση στη μοντελοποίηση των δραστηριοτήτων. Αν λοιπόν κάποια εφαρμογή

¹ (Rodríguez N. D., Cuéllar M. P., Lilius J., Calvo-Flores M. D. 2014. A survey on ontologies for human behavior recognition. *Journal ACM Computing Surveys (CSUR)*. Volume 46 Issue 4, April 2014. Article No. 43. doi: 10.1145/2523819).

θέλει να συσχετίσει τα δεδομένα της με μία άλλη πρέπει να δημιουργήσει ένα πλαίσιο μετάφρασης της μοντελοποίησης της δεύτερης με τη δική της. Κάτι τέτοιο ωστόσο δεν είναι πάντα εφικτό και μπορεί να απαιτεί επέκταση του μοντέλου ώστε να μπορεί να ανταποκρίνεται σε διαφορετικούς τύπους δραστηριοτήτων.

Κρίνεται λοιπόν αναγκαία η ύπαρξη ενός γενικότερου πολυεπίπεδου μοντέλου το οποίο έχοντας ως βάση μία γενική κοινωνική θεωρία περιγραφής των ανθρώπινων δραστηριοτήτων μπορεί να περιγράφει την πλειοψηφία των δεδομένων και δραστηριοτήτων τις οποίες επιτελεί ένας χρήστης σε καθημερινή βάση. Η ύπαρξη ενός τέτοιου μοντέλου θα αποτελούσε ισχυρό εργαλείο για εφαρμογές καταγραφής δραστηριοτήτων και θα επέτρεπε σε ένα χρήστη να πραγματοποιεί ομαδοποίηση και σύγκριση των δεδομένων του από διαφορετικές πηγές.



Ανθρώπινες Δραστηριότητες

Εικόνα 1.1: Πρόβλημα χώρου διπλωματικής

Συνοπτικά, όπως φαίνεται και στην Εικόνα 1.1, υπάρχει αυτή τη στιγμή μία απόσταση από όλες τις ανθρώπινες δραστηριότητες που εκτελούνται και αυτές που μπορούν να αποτυπωθούν σαν πληροφορία ή ψηφιακά αποτυπώματα. Φυσικά οι διάφορες σύγχρονες συσκευές, όπως είναι τα έξυπνα ρολόγια, επιτρέπουν την καταγραφή ψηφιακών αποτυπωμάτων και την περαιτέρω ανάλυση αυτών μέσω τεχνικών τηλεματικής. Παράλληλα, οι ίδιοι οι άνθρωποι έχουν επιλέξει να περιγράφουν τις δραστηριότητές τους στα κοινωνικά δίκτυα, και να τις μοιράζονται με άλλους. Παρόλα αυτά, η ενοποίηση όλων αυτών των διαφορετικών μορφών πληροφορίας σχετικά με τις ανθρώπινες δραστηριότητες δεν είναι κάτι που έχει επιτευχθεί ακόμα επιτυχώς, όπως και οι ίδιες οι εταιρίες δεν έχουν καταλήξει σε κάποια συγκεκριμένα πρότυπα που διευκόλυναν την διαλειτουργικότητα.

1.2 Αντικείμενο Διπλωματικής

Η παρούσα εργασία έχει ως βασικό στόχο την κατασκευή ενός γενικού μοντέλου περιγραφής ανθρώπινων δραστηριοτήτων με την επιλογή και σύνθεση των κατάλληλων οντολογιών. Τα παραδείγματα εφαρμογών που προαναφέρθηκαν στηρίζονται σε μοντέλα τα οποία είναι σχετικά με ένα περιορισμένο υποσύνολο ανθρώπινων δραστηριοτήτων οι οποίες ανήκουν σε μία κατηγορία. Για την εκπόνηση της διπλωματικής ωστόσο, ήταν αναγκαία η κατασκευή ενός μοντέλου το οποίο μπορεί να περιγράψει το σύνολο των

ανθρώπινων δραστηριοτήτων ελαχιστοποιώντας τις απώλειες, αλλά να είναι σε θέση να περιγράψει ικανοποιητικά κάθε τύπο δραστηριότητας. Με άλλα λόγια επιχειρήθηκε να βρεθούν οντολογίες και να συνδεθούν για την δημιουργία μίας τελικής οντολογίας η οποία θα μπορεί να ανταποκρίνεται σε κάθε είδος δραστηριότητας ανεξαρτήτως κατηγορίας και τύπου. Σκοπός του τελικού μοντέλου ήταν η σύνδεσή του με πραγματικά δεδομένα τα οποία λήφθηκαν από την εφαρμογή Activity Tracker με σκοπό να επιχειρηθεί να δοκιμαστεί η δυνατότητα συλλογιστικής ώστε να επικυρωθεί η εγκυρότητα και η ορθότητα της οντολογίας.

Η επόμενη απόφαση αφορούσε την γλώσσα που θα χρησιμοποιηθεί για το κομμάτι της συλλογιστικής καθώς και ο τύπος των ερωτήσεων που θα γίνουν στην τελική οντολογία. Έπρεπε δηλαδή να απαντηθεί η ερώτηση σχετικά με το ποιες πληροφορίες μπορεί να ζητήσει ο χρήστης ενός τέτοιου γενικότερου μοντέλου σχετικά με τις δραστηριότητές του και τις δραστηριότητες άλλων χρηστών με τους οποίους είναι συνδεδεμένος με κάποιο τρόπο.

Το τελευταίο κομμάτι είχε σχέση με την επιλογή του κατάλληλου πλαισίου παρουσίασης των αποτελεσμάτων. Αν και η εφαρμογή που επιλέχθηκε για την πραγματοποίηση ερωτήσεων πάνω στην οντολογία παρουσιάζει τα αποτελέσματα των ερωτήσεων κρίθηκε αναγκαία η ανάπτυξη εφαρμογής στην οποία ένας χρήστης θα μπορεί να δει αυτά τα δεδομένα σε μορφή πιο εύκολη για ανάγνωση. Καθορίστηκαν αρχικά οι λειτουργίες που πρέπει να επιτελεί μία τέτοια εφαρμογή, καθώς και η βέλτιστη αρχιτεκτονική σχεδίαση του συστήματος και των υπομονάδων του. Με βάση την αρχιτεκτονική αυτή, επιλέχθηκε η κατάλληλη γλώσσα προγραμματισμού και η πλατφόρμα η οποία θα μπορούσε να επιφέρει τα καλύτερα αποτελέσματα στη σχεδίαση του συστήματος.

1.2.1 Συνεισφορά

Εποπτικά, η συγκεκριμένη διπλωματική βοήθησε στην εξερεύνηση των δυνατοτήτων ανάπτυξης σημασιολογικών εφαρμογών για την καταγραφή των ανθρώπινων δραστηριοτήτων, με στόχο την περαιτέρω χρησιμοποίηση αυτής της πληροφορίας. Πιο συγκεκριμένα, η συνεισφορά της διπλωματικής συνοψίζεται στα εξής τέσσερα σημεία:

1. Μελετήθηκαν και παρουσιάστηκαν οντολογίες δραστηριοτήτων και εργασίες μοντελοποίησης ανθρώπινων ενεργειών και συμπεριφοράς.
2. Πραγματοποιήθηκε σύνδεση οντολογιών για την κατασκευή ενός τελικού μοντέλου ικανού να περιγράψει τις ανθρώπινες δραστηριότητες έχοντας ως θεωρητική βάση τη θεωρία δραστηριοτήτων.
3. Πραγματοποιήθηκαν ερωτήσεις πάνω στην τελική οντολογία χρησιμοποιώντας την προγραμματιστική γλώσσα SPARQL με σκοπό την εμφάνιση σχέσεων ανάμεσα στους χρήστες και τις δραστηριότητές τους.
4. Παρουσιάστηκαν οι απαιτήσεις ενός συστήματος εμφάνισης των τελικών αποτελεσμάτων. Στη συνέχεια αναπτύχθηκε εφαρμογή η οποία όντας συνδεδεμένη με την οντολογία εμφανίζει τα αποτελέσματα των ερωτήσεων που προαναφέρθηκαν.

1.2.2 Μεθοδολογία

Η μεθοδολογία που ακολουθήθηκε περιλαμβάνει έναν αριθμό από σειριακά βήματα όπως φαίνεται και στην εικόνα 1.2, τα οποία περιλαμβάνουν καταρχάς την έρευνα του χώρου και τη μελέτη σχετικών εργασιών. Έπειτα, έχοντας επιλέξει τις κατάλληλες οντολογίες

προχώρησα στη σύνθεση της τελικής οντολογίας και τη σύνδεσή της με πραγματικά δεδομένα. Στη συνέχεια προχώρησα στη διαδικασία συλλογιστικής χρησιμοποιώντας προγραμματιστική γλώσσα ερωτήσεων. Τέλος ανέπτυξα την εφαρμογή έχοντας επιλέξει τις παραμέτρους υλοποίησής της και πραγματοποιήσα διαδικασία αξιολόγησης.



Εικόνα 1.2: Μεθοδολογία διπλωματικής

1.3 Οργάνωση Κειμένου

Στο κεφάλαιο 2 παρουσιάζεται το θεωρητικό υπόβαθρο το οποίο έπρεπε να μελετηθεί για την εκπόνηση της διπλωματικής. Στο κεφάλαιο 3 παρουσιάζονται παρεμφερείς εργασίες στον τομέα των δραστηριοτήτων πολλές από τις οποίες επηρέασαν και την παρούσα. Το κεφάλαιο 4 ασχολείται με την παρουσίαση των οντολογιών που επιλέχθηκαν για την κατασκευή του μοντέλου, τη σύνδεση αυτών και τις ερωτήσεις που έγιναν στην οντολογία με χρήση της γλώσσας SPARQL. Στο κεφάλαιο 5 καθορίζονται οι απαιτήσεις του συστήματος, παρουσιάζονται οι λειτουργίες που πρέπει να επιτελεί και γίνεται ανάλυση της αρχιτεκτονικής του συστήματος που αναπτύχθηκε. Στο κεφάλαιο 6 γίνεται αναφορά στα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν για την σύνδεση των οντολογιών, την πραγματοποίηση ερωτήσεων και την ανάπτυξη της εφαρμογής. Το κεφάλαιο 7 αναλύει τα συμπεράσματα που προέκυψαν από την εργασία και προτείνει μελλοντικές επεκτάσεις και τέλος το κεφάλαιο 8 παρουσιάζει τη βιβλιογραφία που χρησιμοποιήθηκε.

2 Θεωρητικό Υπόβαθρο

Σε αυτό το κεφάλαιο θα γίνει περιγραφή της θεωρίας η οποία έπρεπε να μελετηθεί για την εκπόνηση της εργασίας.

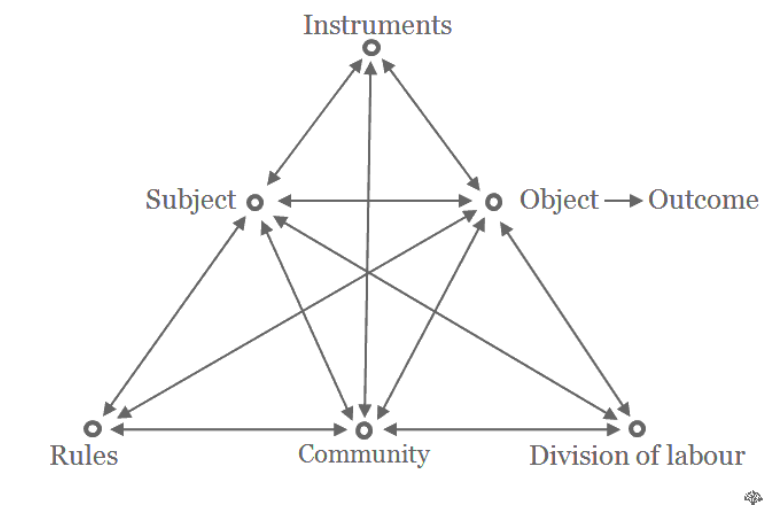
2.1 Θεωρία Δραστηριοτήτων (Activity Theory)

Η θεωρία δραστηριοτήτων(Activity Theory) είναι ο όρος που χρησιμοποιούμε για μία σειρά από κοινωνικές θεωρήσεις και έχει τις απαρχές της στο Ινστιτούτο Ψυχολογίας της Μόσχας από τους ερευνητές Lev Semjonovich Vygotsky, Alexander Romanovich Luria και Alexei Nikolaevich Leont'ev.Τη μεγαλύτερη επίδραση όσον αφορά σε προ-Σοβιετικές αναπτύξεις στον τομέα των Activity Theories έχει η διατύπωση της θεωρίας δραστηριοτήτων του Leont'ev η οποία είναι μάλλον κοινωνικο-επιστημονική και οργανωτική παρά ψυχολογική έρευνα.

Η θεωρία δραστηριοτήτων είναι ένα ισχυρό περιγραφικό εργαλείο σκοπός του οποίου είναι να κατανοήσει τη σχέση ανάμεσα στην ανθρώπινη συνείδηση και τη δραστηριότητα(activity).Μία δραστηριότητα αναλαμβάνεται από κάποιο υποκείμενο (subject),που μπορεί να είναι είτε κάποιος άνθρωπος είτε μία συλλογική δομή(οργάνωση,ομάδα κτλ),το οποίο δουλεύει πάνω σε κάποιο αντικείμενο(object) για να επιτύχει μία επιθυμητή έξοδο(outcome) σύμφωνα πάντα με κάποιο κίνητρο(motivation).Το αντικείμενο της δραστηριότητας μπορεί να αναφέρεται και σε κάποιον άνθρωπο(η οποία μάλιστα είναι και η σύνηθης περίπτωση) και όχι απαραίτητα σε κάποιο άψυχο αντικείμενο.Για την εκπλήρωση της εξόδου το υποκείμενο επιστρατεύει εργαλεία τα οποία μπορεί να είναι εξωτερικά,όπως ένα τσεκούρι ή ένας υπολογιστής,ή εσωτερικά όπως ένα σχέδιο ή η ανθρώπινη εμπειρία.Η έννοια των εσωτερικών εργαλείων που δημιουργούνται από την ανθρώπινη νοημοσύνη ακολουθεί την αρχή ότι οι άνθρωποι αποτελούν το ίδιο το εργαλείο ανάμεσα στους ίδιους(subject) και το αντικείμενο(object)(Wells, 2002).Ο Engestrom το 1987 ανέπτυξε ένα πιο διευρυμένο μοντέλο της θεωρίας δραστηριοτήτων στο οποίο προσθέτει και έναν ακόμα παράγοντα την κοινωνία δηλαδή όλους όσους συμμετέχουν έμμεσα ή άμεσα στη δραστηριότητα(«αυτούς που μοιράζεται το ίδιο αντικείμενο»).Έπειτα πρόσθεσε κανόνες που αφορούν τη μεσολάβηση μεταξύ του υποκειμένου και της κοινωνίας,και τη διαίρεση της εργασίας ανάμεσα σε αντικείμενο και κοινωνία.Οι κανόνες(rules) καλύπτουν ρητούς και αυτονόητους κώδικες,συμβάσεις και κοινωνικές σχέσεις μέσα σε μία κοινότητα.Η διαίρεση της εργασίας(division of labor),αναφέρεται στην ιεραρχική δομή της δραστηριότητας και τον διαμοιρασμό των δραστηριοτήτων ανάμεσα σε όλους τους παράγοντες(actors) του συστήματος. Επομένως η θεωρία δραστηριοτήτων περιλαμβάνει την αντίληψη ότι η δραστηριότητα είναι αποτέλεσμα του κοινωνικού περιεχομένου ή ειδικότερα της κοινότητας.Το μοντέλο του Engestrom για τη θεωρία δραστηριοτήτων φαίνεται παρακάτω στο Σχήμα 2.1.

Ας δούμε όμως τη θεωρία δραστηριοτήτων όπως φαίνεται στο μοντέλο του Engestrom μέσα από το παράδειγμα ενός αγώνα μπάσκετ.Οι παίχτες(object) παίζουν με σκοπό να νικήσουν(object->outcome).Χρησιμοποιούν εξωτερικά εργαλεία(μπάλα,καλάθια) αλλά και εσωτερικά εργαλεία(εξυπνάδα,εμπειρία,αίσθηση του παιχνιδιού) και οφείλουν να υπακούν στους κανόνες του παιχνιδιού οι οποίοι είναι καλά ορισμένοι.Στην κοινότητα γύρω από το παιχνίδι μπορεί να ανήκουν άτομα(προπονητής, πρόεδρος της ομάδας) και σύνολα(φίλαθλοι).Η διαίρεση της εργασίας είναι εύκολο να κατανοηθεί καθώς το μπάσκετ

είναι ομαδικό άθλημα και κάθε παίχτης έχει συγκεκριμένο ρόλο και αρμοδιότητες για την επίτευξη της νίκης.



Σχήμα 2.1: Μοντέλο Θεωρίας Δραστηριοτήτων (Engeström 1987)

Μία σημαντική διάκριση στη θεωρία δραστηριοτήτων βρίσκεται ανάμεσα στις έννοιες της δράσης, δραστηριότητας και διαδικασίας (action, activity, operation). Η διάκριση αυτή υποθέτει ότι οι άνθρωποι πραγματοποιούν δράσεις οι οποίες από μόνες τους δεν συνεισφέρουν απαραίτητα με άμεσο τρόπο στην εκπλήρωση του σκοπού της δραστηριότητας. Αυτό σημαίνει ότι οι πράξεις πραγματοποιούνται μόνο επειδή βρίσκονται στο πλαίσιο κάποιας διαδικασίας. Παρόλα αυτά, συλλογικά, οι δράσεις (τις οποίες αναλαμβάνουν άτομα ή ομάδες) οδηγούν στην ικανοποίηση του κινήτρου μέσα από την κατανόηση του στόχου και τυπικά έχουν νόημα μόνο μέσα στο γενικό πλαίσιο μίας κοινόχρηστης δραστηριότητας. Κάτω από το επίπεδο των δράσεων βρίσκονται οι διαδικασίες οι οποίες μπορεί να είναι συνειδητές κατά την περίοδο εκμάθησής τους αλλά συνήθως καταλήγουν να είναι υποσυνείδητες (ρουτίνες) με το πέρασμα του χρόνου καθώς η ευχέρεια του ατόμου με τη διαδικασία αυξάνεται. Συνεπώς μία δραστηριότητα αποτελείται από έναν αριθμό δράσεων και κάθε δράση μπορεί να απαρτίζεται από αρκετές διαδικασίες. Ένα κλασικό παράδειγμα για κατανόηση των τριών επιπέδων της θεωρίας δραστηριοτήτων είναι η κατασκευή ενός σπιτιού (δραστηριότητα), η οποία μπορεί να περιλαμβάνει την κατασκευή των θεμελίων, την επισκευή της οροφής κτλ (δράσεις) και τη χρήση εργαλείων (σφυρί, πριόνι), την τοποθέτηση των τούβλων κτλ (διαδικασίες). Ενδιαφέρον παρουσιάζει το γεγονός ότι τα τρία επίπεδα της δραστηριότητας δεν είναι πάντα σταθερά. Κάποιες δραστηριότητες μπορεί να γίνουν αργότερα δράσεις ενώ οι δράσεις μπορεί να γίνουν διαδικασίες.

2.2 Σημασιολογία και Σημασιολογικός Ιστός

2.2.1 Σημασιολογικός Ιστός

Ως σημασιολογία συνήθως ορίζεται ο κλάδος της γλωσσικής επιστήμης που εξετάζει την σημασιολογική δομή μίας γλώσσας. Στον τομέα της πληροφορικής ωστόσο ο όρος semantics αναφέρεται στη σημασία μίας γλώσσας προγραμματισμού, προμηθεύει δηλαδή τους

κανόνες για την ερμηνεία μίας σύνταξης.Οι κανόνες αυτοί δεν παρέχουν άμεσα το νόημα της σύνταξης αλλά θέτουν κάποιους περιορισμούς στο τι δηλώνει αυτή η σύνταξη.Επιπλέον ο όρος semantics αναφέρεται σε συγκεκριμένους τύπους δεδομένων(semantic data) οι οποίοι χαρακτηρίζονται από τη χρήση κατευθυνόμενων γράφων,οι κορυφές των οποίων αναπαριστούν έννοιες ή οντότητες και οι ακμές δηλώνουν σχέσεις μεταξύ τους.Τα σημασιολογικά μοντέλα θα περιγραφούν με μεγαλύτερη λεπτομέρεια στη συνέχεια αφού οριστεί και εξηγηθεί η έννοια του σημασιολογικού ιστού.

Αρχικά θα παραθέσω κάποιες πληροφορίες για τον παγκόσμιο ιστό όπως τον γνωρίζουμε σήμερα.Κάποτε η ιδέα ενός ιστού πληροφοριών ήταν μία τεχνική ιδέα προσβάσιμη μόνο από λίγους ανθρώπους που είχαν το κατάλληλο επίπεδο εκπαίδευσης σε αυτόν τον τομέα. Στη σημερινή κοινωνία ωστόσο,οι περισσότεροι άνθρωποι είναι οικείοι με την ιδέα του Internet καθώς ο ιστός περιέχει πληροφορίες για κάθε ζήτημα από διάφορες πηγές.Βασική έννοια στην ιδέα του παγκόσμιου ιστού είναι η ιδέα μίας ανοιχτής κοινότητας.Ο καθένας μπορεί να συνεισφέρει πληροφορίες στο σύνολο,οι οποίες είναι ορατές από όλους τους χρήστες.Μπορεί να ειπωθεί ότι ο ιστός πληροφοριών είναι ένας ζωντανός οργανισμός ο οποίος αναπτύσσεται ανάλογα με τα ενδιαφέροντα και τις συνεισφορές της κοινότητας που τον χρησιμοποιεί.Αυτό καταδεικνύει τη χρησιμότητά του παρά το γεγονός ότι πλέον είναι ένα συνονθύλευμα από διαφορετικές αναλύσεις,παρουσιάσεις και περιλήψεις οποιουδήποτε θέματος.Παρόλα αυτά,αν κάποιος έχει την υπομονή και την κριτική σκέψη που χρειάζεται μπορεί να βρει βοήθεια για οποιαδήποτε ερώτηση ή θέμα τον ενδιαφέρει. Πώς μπορούμε λοιπόν να κάνουμε την ιδέα του Internet πιο ολοκληρωμένη και συνεπή όσον αφορά της πληροφορίες που υπάρχουν σε αυτό;Η απάντηση είναι με τον σημασιολογικό ιστό.

Ο σημασιολογικός ιστός(Web 3.0) είναι μία επέκταση του σημερινού ιστού που θα φέρει δομή στο ουσιαστικό περιεχόμενο των ιστοσελίδων.Ο όρος εφευρέθηκε από τον Tim Berners-Lee, δημιουργό του παγκόσμιου ιστού(World Wide Web) και διευθυντή του World Wide Web Consortium(W3C), το οποίο επιβλέπει την ανάπτυξη των προτάσεων για τα στάνταρ του σημασιολογικού ιστού.Η λογική πίσω από το semantic web είναι ότι η δημοσιευμένη πληροφορία θα περιέχει μετα-δεδομένα,τα οποία θα είναι κοινά για όλους, θα μπορούν να κατανοούνται και από μηχανές,οι οποίες θα βοηθήσουν στην καλύτερη συλλογή και επεξεργασία τους. Μερικά από τα πεδία στα οποία αναμένεται να έχει την μεγαλύτερη επίδραση είναι στην υγεία, στην παιδεία και στις επιχειρήσεις. Υπάρχουν ήδη πολλές προσπάθειες από εταιρίες, ερευνητές και μη κερδοσκοπικές οργανώσεις για να παράγουν πρότυπα οντολογιών, κυρίως για τα παραπάνω πεδία, για να υπάρχουν κοινές γλώσσες και περισσότερα δεδομένα τα οποία να μπορούν να συνδυαστούν για καλύτερα αποτελέσματα. Στην υγεία, γίνεται προσπάθεια για τη δημιουργία ενοποιημένων γλωσσών ιατρικής ορολογίας και υπηρεσίες που θα βοηθάνε το ιατρικό προσωπικό και θα κατευθύνουν τους καταναλωτές σε αξιόπιστες πληροφορίες υγείας σχετικά με την κατάστασή τους. Στην εκπαίδευση, ο Σημασιολογικός Ιστός θα συμβάλει σημαντικά στην μάθηση κυρίως στον τρόπο αναζήτησης πληροφοριών, στην οργάνωση των αποτελεσμάτων και στη δημιουργία ενός προγράμματος μάθησης ειδικό για τον καθένα. Στον επιχειρηματικό τομέα, θα υπάρχει καλύτερη οργάνωση των εταιριών, καλύτερες εμπειρίες για τους χρήστες στις διαδικτυακές αγορές και καλύτερος συντονισμός μεταξύ διαφορετικών εταιριών. Στην καθημερινότητά θα υπάρχουν επιδράσεις του Web 3.0 στα

κοινωνικά δίκτυα και εικονικές κοινότητες. Θα υπάρχουν εφαρμογές, οι οποίες θα δίνουν περισσότερες, πιο έμπιστες, πληροφορίες και θα διευκολύνουν σημαντικά τις διαδικτυακές δραστηριότητες.

Ας δούμε όμως την πρακτική εφαρμογή των παραπάνω μέσα από ένα παράδειγμα. Ας υποθέσουμε ότι κάποιος χρήστης του Internet κάνει μία έρευνα σχετικά με το ηλιακό μας σύστημα και βρίσκει μία σελίδα με αρκετές λεπτομέρειες για όλα τα αντικείμενα που περιλαμβάνει: αστέρες, πλανήτες, δορυφόρους, αστεροειδείς και κομήτες κάθε ένα από τα οποία έχει τη δικιά του σελίδα με πλούσιο φωτογραφικό υλικό και λεπτομερείς πληροφορίες. Στην κορυφή κάθε σελίδας αναγράφεται η κατηγορία του αντικειμένου, για παράδειγμα στη σελίδα της Γης αναγράφεται στην κορυφή η λέξη πλανήτης. Υπάρχει τέλος μία σελίδα η οποία παραθέτει τους εννέα γνωστούς πλανήτες του ηλιακού συστήματος με link ανακατεύθυνσης στην σελίδα του καθενός. Ωστόσο, μία μέρα ο χρήστης διαβάζει ότι ο Πλούτωνας δεν θεωρείται πλέον πλανήτης από τους επιστήμονες αλλά χαρακτηρίζεται ως πλανήτης "νάνος". Έπειτα διαπιστώνει ότι στη σελίδα του Πλούτωνα, αναγράφεται η σωστή ορολογία αλλά στη λίστα με τους γνωστούς πλανήτες δεν έχει γίνει η αλλαγή και αναρωτιέται γιατί οι σελίδες δεν ενημερώθηκαν με μεγαλύτερη συνέπεια. Το πρόβλημα έγκειται στο γεγονός ότι τα δεδομένα της σελίδας περιγράφουν την παρουσίαση των πληροφοριών και όχι τα ίδια τα αντικείμενα της.

Ένας απλός τρόπος για να γίνουν οι εφαρμογές στο Internet πιο ολοκληρωμένες είναι με μία σημασιολογική βάση δεδομένων. Αυτό θα έχει ως αποτέλεσμα οι σελίδες να παράγονται από ερωτήσεις που γίνονται στη βάση δεδομένων. Με άλλα λόγια η ενημέρωση μίας σελίδας θα γίνεται αυτόματα όταν ενημερώνονται και τα περιεχόμενα της βάσης και όχι μόνο. Όλες οι σελίδες που έχουν πληροφορίες σχετικά με ένα θέμα θα αλλάζουν όταν αλλάζει και η καταχώρηση επί του θέματος στη βάση δεδομένων χωρίς να χρειαστεί κάποια παρέμβαση από τον δημιουργό της σελίδας. Το γεγονός αυτό, θα οδηγήσει σε μεγαλύτερη συνέπεια ανάμεσα σε σελίδες που έχουν πληροφορίες για το ίδιο θέμα και λιγότερα μπερδέματα από την πλευρά του χρήστη όταν ψάχνει πληροφορίες για κάποιο θέμα σε περισσότερες από μία σελίδες στο internet. Η κύρια ιδέα λοιπόν του σημασιολογικού ιστού είναι να υποστηρίξει τη διανομή των πληροφοριών στο επίπεδο των δεδομένων και όχι στο επίπεδο της παρουσίασης αυτών. Αυτό σημαίνει ότι δεν θα έχουμε πλέον σελίδες να δείχνουν σε άλλες σελίδες με σύνδεσμους (Uniform Resource Locators, URLs), αλλά δεδομένα θα μπορούν να δείχνουν σε άλλα δεδομένα χρησιμοποιώντας παραπομπές (global references) που ονομάζονται Uniform Resource Identifiers (URIs). Η υποδομή του παγκόσμιου ιστού θα παρέχει ένα μοντέλο δεδομένων όπου πληροφορίες για κάποια οντότητα θα μπορούν να διανεμηθούν σε όλο το Internet. Το μοντέλο δεδομένων του σημασιολογικού ιστού λέγεται Resource Description Framework (RDF) και θα εξηγηθεί σε επόμενη παράγραφο.

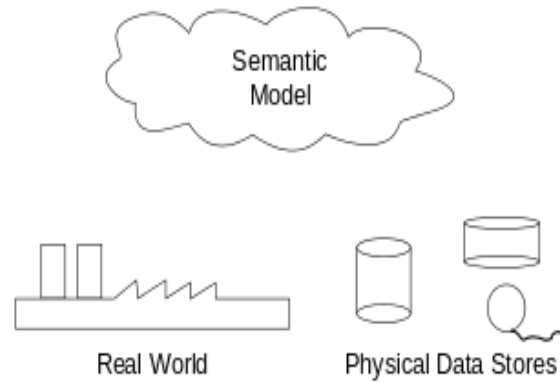
Τι θα αλλάξει όμως με την ενωμάτωση σημασιολογικών βάσεων δεδομένων στον παγκόσμιο ιστό όσον αφορά τον τρόπο με τον οποίο ένας χρήστης μπορεί να δημοσιεύσει πληροφορίες; Στο σημερινό Internet καθένας μπορεί να γράψει οτιδήποτε θέλει για ένα θέμα και εξαρτάται από τον χρήστη στο τι συμπέρασμα θα καταλήξει. Αυτό το χαρακτηριστικό του Internet συχνά αναφέρεται ως AAA (Anyone can say Anything about Any topic) και είναι ζωτικό για τη λειτουργία του παγκόσμιου ιστού για την ελεύθερη ανταλλαγή

πληροφοριών, είτε μιλάμε για τον παγκόσμιο ιστό είτε για τον σημασιολογικό. Στην περίπτωση του σημασιολογικού ιστού λοιπόν, θα πρέπει η υποδομή των δεδομένων να επιτρέπει σε κάθε άτομο να δημιουργεί δεδομένα για κάποια οντότητα με τρόπο που η πληροφορία αυτή θα μπορεί να συνδυάζεται με άλλες πληροφορίες για το ίδιο θέμα από διαφορετικές πηγές. Αυτή η ανάγκη δημιούργησε τη βάση για το σχεδιασμό του RDF. Ωστόσο για να λειτουργήσει ο σημασιολογικός ιστός πρέπει τα δεδομένα που υπάρχουν αυτή τη στιγμή στο Internet να μετατραπούν σε μορφή RDF. Είναι γεγονός ότι στα πρώιμα στάδια του σημασιολογικού ιστού υπήρχε μικρό κίνητρο για μία τέτοια διαδικασία αλλά με το πέρασμα του χρόνου και την αύξηση του ερευνητικού ενδιαφέροντος στα semantics όλο και περισσότερες πληροφορίες είναι διαθέσιμες σε μορφή RDF, άρα γίνονται όλο και πιο χρήσιμες εφαρμογές που χρησιμοποιούν διανεμημένα δεδομένα (distributed data). Αυτή εξάλλου είναι και η βασική ιδέα του δικτυακού φαινομένου (network effect) σύμφωνα με το οποίο η αξία μίας υπηρεσίας εξαρτάται από τον αριθμό των ατόμων που την χρησιμοποιούν.

2.2.2 Σημασιολογικά Μοντέλα

Στη συνέχεια θα παρατεθούν κάποιες πληροφορίες για τα σημασιολογικά μοντέλα πριν προχωρήσουμε στην περιγραφή του μοντέλου RDF. Το σημασιολογικό μοντέλο είναι ένα εννοιολογικό μοντέλο δεδομένων το οποίο επιτρέπει στους χρήστες που ανταλλάσσουν πληροφορίες να βγάζουν νόημα από instances πληροφοριών χωρίς να γνωρίζουν το μετα-μοντέλο. Οι πληροφορίες εκφράζονται σαν δυαδικές σχέσεις ανάμεσα στα δεδομένα. Τυπικά οι δυαδικές σχέσεις έχουν τη μορφή τριπλών οντοτήτων (triples): Subject-RelationType-Object. Για παράδειγμα: «η Ακρόπολη <βρίσκεται στην> Αθήνα». Τα σημασιολογικά μοντέλα συνήθως περιλαμβάνουν αρκετές σχέσεις που μπορούν να χρησιμοποιηθούν για να συνδέσουν δεδομένα όπως για παράδειγμα η σχέση <is located in>. Επιπλέον, τυποποιούν το νόημα τέτοιων σχέσεων, έτσι ώστε να μπορεί να γίνει κατανοητή η σημασία των δεδομένων μέσα από τις σχέσεις που τα συνδέουν με άλλα δεδομένα. Όπως φαίνεται και στο σχήμα 2.2 ο πραγματικός κόσμος, όσον αφορά τους πόρους του, ιδέες, γεγονότα κτλ ορίζεται συμβολικά μέσα σε physical data stores. Το σημασιολογικό μοντέλο δεδομένων είναι στην ουσία ένα αφηρημένο μοντέλο το οποίο ορίζει το συσχετισμό των αποθηκευμένων δεδομένων με τον πραγματικό κόσμο. Για αυτόν τον λόγο το μοντέλο πρέπει να είναι μία αληθινή αναπαράσταση του πραγματικού κόσμου.

Ο σημασιολογικός ιστός χρησιμοποιεί την ιδέα της ιεραρχίας κλάσεων για να εκφράσει τις έννοιες της ομοιότητας και της ποικιλίας. Κλάσεις υψηλού επιπέδου δείχνουν ομοιότητες ανάμεσα σε μία μεγάλη ομάδα οντοτήτων ενώ κλάσεις χαμηλότερου επιπέδου δείχνουν ομοιότητες σε μικρότερες πιο συγκεκριμένες ομάδες αντικειμένων. Στο σχήμα 2.3 φαίνεται



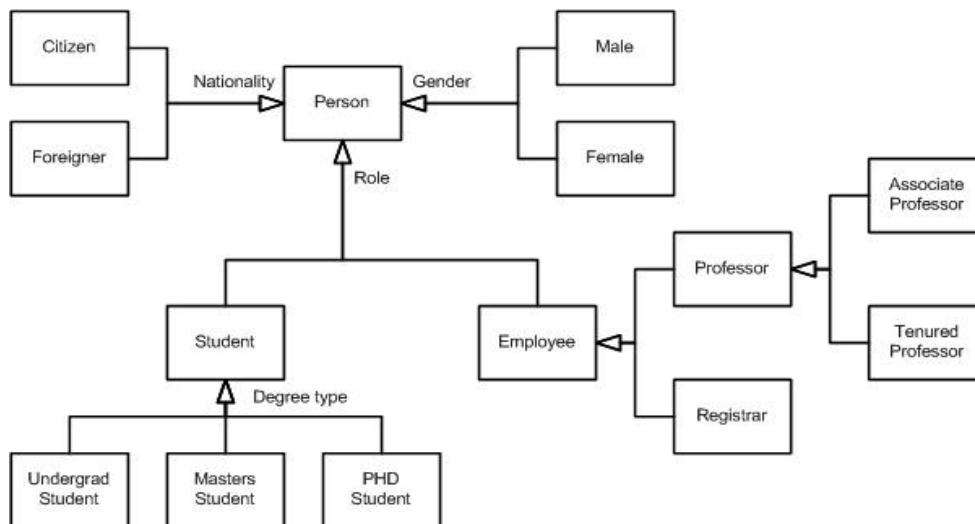
Σχήμα 2.2:Σημασιολογικό Μοντέλο²

πρακτικά η σημασία των κλάσεων και των υποκλάσεων στο σημασιολογικό μοντέλο. Η κλάση υψηλότερου επιπέδου είναι η κλάση Person η οποία συνδέεται με τις υποκλάσεις της μέσω σχέσεων (relation types). Οι σχέσεις που μπορούν να εκφραστούν από το παραπάνω μοντέλο γίνονται εύκολα κατανοητές και δείχνουν λεπτομέρειες της κλάσης Person ανάλογα με το relation type που θα χρησιμοποιηθεί. Ένα τυχαίο άτομο μπορεί να περιγραφεί με το παραπάνω μοντέλο ως εξής:

- Person<hasNationality>Citizen
- Person<hasRole>Student
- Person<hasGender>Male
- Student<hasDegreeType>Undergrad Student

Από τις τέσσερις παραπάνω σχέσεις μπορεί κάποιος εύκολα να συμπεράνει ότι το άτομο στο οποίο αναφερόμαστε κατάγεται από τη χώρα στην οποία ζει, είναι γένους αρσενικού και έχει την ιδιότητα του προπτυχιακού φοιτητή.

² https://en.wikipedia.org/wiki/Semantic_data_model



Σχήμα 2.3: Παράδειγμα σημασιολογικού μοντέλου³

2.2.3 RDF , RDF Schema (RDFs) & OWL

Το RDF (Resource Description Framework) είναι μία οικογένεια προδιαγραφών σχεδιασμένη από το World Wide Web Consortium (W3C) για να χρησιμοποιηθεί ως μοντέλο μεταδεδομένων (metadata data model). Πλέον χρησιμοποιείται σαν μία γενική μέθοδος για την εννοιολογική περιγραφή ή τη μοντελοποίηση πληροφοριών και εφαρμόζεται σε web resources, χρησιμοποιώντας μία ποικιλία από συντακτικές σημειογραφίες. Το RDF χρησιμοποιεί triples της μορφής subject<predicate>object για να δείξει τη σχέση (predicate) ανάμεσα σε ένα υποκείμενο (subject) και ένα αντικείμενο (object). Οι RDFs και OWL είναι γλώσσες οντολογιών που έχουν χτιστεί πάνω από την RDF. Συγκεκριμένα η RDFs είναι επέκταση της RDF που επιτρέπει σε πόρους να χαρακτηριστούν σαν κλάσεις ή ιδιότητες και μία καλύτερη εκφραστικότητα. Έχει μεγαλύτερο λεξιλόγιο με περισσότερους σημασιολογικούς περιορισμούς. Η OWL (Web Ontology Language) επεκτείνει την RDFs με επιπλέον κατασκευαστές όπως πληθικότητα (cardinality), περιορισμό στις τιμές ή χαρακτηριστικά στις ιδιότητες (π.χ. μεταβατικότητα). Βασίζεται στις αρχές των περιγραφικών λογικών (DL) και φέρνει τη δύναμή της λογικής στον σημασιολογικό ιστό.

Πριν προχωρήσω σε περαιτέρω ανάλυση των παραπάνω εννοιών θα ορίσω την έννοια της οντολογίας η οποία είναι κομβικής σημασίας όταν αναφερόμαστε στις παραπάνω σημασιολογικές γλώσσες. Στην πληροφορική οντολογία είναι ένας σαφής και τυπικός ορισμός μίας κοινής και συμφωνημένης εννοιολογικής μορφοποίησης που αφορά σε ένα πεδίο ενδιαφέροντος. Στην ουσία είναι ένα σύνολο εννοιών, σχέσεων και ιδιοτήτων και χρησιμοποιείται συνήθως για συλλογιστική και για την δομημένη περιγραφή γνώσης ενός πεδίου ενδιαφέροντος. Οι οντολογίες έχουν καθιερωθεί ως δομημένα πλαίσια για την οργάνωση πληροφορίας και χρησιμοποιούνται κατά κόρον στον σημασιολογικό ιστό ως μία μορφή αναπαράστασης γνώσης για τον κόσμο.

³ https://en.wikipedia.org/wiki/Semantic_data_model

Επιστρέφοντας στο RDF, ενδιαφέρον παρουσιάζει η περίπτωση στην οποία περισσότερα από ένα triple αναφέρονται στην ίδια οντότητα. Στην περίπτωση αυτή είναι πιο βολικό να πούμε ότι κάθε triple είναι ένα βέλος από το υποκείμενο στο αντικείμενο όπου το βέλος αναπαριστά τη σχέση μεταξύ τους, το οποίο καταλήγει σε έναν κατευθυνόμενο γράφημα. Ένα από τα πλεονεκτήματα των triples είναι η ευκολία με την οποία μπορούμε να συγχωνεύσουμε δεδομένα από διαφορετικές πηγές. Εφόσον οι πληροφορίες αναπαριστώνται από triples, η συγχώνευση πληροφοριών από δύο γραφήματα είναι απλή διαδικασία. Το μόνο που χρειάζεται να κάνουμε είναι να κατασκευάσουμε ένα γράφημα το οποίο περιέχει συγχωνευμένα όλα τα triples από το κάθε γράφημα. Το ερώτημα που πρέπει να απαντηθεί φυσικά κατά τη συγχώνευση γραφημάτων είναι το εξής: πότε ένας κόμβος ενός γράφου είναι ίδιος με έναν κόμβο κάποιου άλλου γράφου.

Ας υποθέσουμε ότι οι κόμβοι και οι ακμές ενός γράφου έχουν ονόματα όπως Σωκράτης και Ελλάδα. Η σχέση μεταξύ τους είναι προφανής: Σωκράτης <κατάγεται από> Ελλάδα. Στο σημασιολογικό ιστό ωστόσο αυτή δεν είναι επαρκής πληροφορία για να πούμε ότι ο κόμβος Σωκράτης ενός γραφήματος είναι ίδιος με τον κόμβο Σωκράτης ενός άλλου γραφήματος. Το RDF λύνει αυτό το ζήτημα με τη χρήση URIs (Uniform Resource Identifiers). Η σύνταξη και η μορφή των URIs είναι γνωστή στους περισσότερους χρήστες του διαδικτύου εξαιτίας της πιο γνωστής έννοιας URL. Ένα URI παρέχει μία οικουμενική ταυτότητα σε έναν πόρο ο οποίος είναι κοινός στο χώρο του διαδικτύου. Άρα αν δύο agents στο διαδίκτυο θέλουν να αναφερθούν στον ίδιο όρο καλό είναι να συμφωνήσουν σε ένα κοινό URI για αυτόν τον όρο. Τα URIs και τα URLs έχουν ακριβώς την ίδια μορφή διότι όπως τα URLs έτσι και τα URIs είναι αναγνωριστικά με οικουμενική σκοπιά. Το γεγονός ότι μπορούμε να έχουμε μοντέλα στον σημασιολογικό ιστό, τα αντικείμενα των οποίων έχουν οικουμενικούς identifiers επιτρέπει στα μοντέλα να συμμετέχουν στην υποδομή του παγκόσμιου ιστού. Επιστρέφοντας στο προηγούμενο πρόβλημα η λύση είναι απλή: Ένας κόμβος ενός γράφου συγχωνεύεται με έναν κόμβο κάποιου άλλου γράφου αν έχουν ακριβώς το ίδιο URI.

Όπως προαναφέρθηκε τα URIs λειτουργούν πολύ αποτελεσματικά για να δηλώσουν την ταυτότητα μίας οντότητας στον παγκόσμιο ιστό, αλλά παρουσιάζουν κάποια δυσκολία στο γράψιμό τους όταν κατασκευάζουμε μοντέλα. Για τον λόγο αυτό, συνήθως χρησιμοποιούμε συντομεύσεις τους οι οποίες ονομάζονται qnames. Στην απλούστερη μορφή του ένα URI το οποίο εκφράζεται σαν qname αποτελείται από δύο μέρη: το namespace και το identifier (αναγνωριστικό). Για παράδειγμα το URI της Αγγλίας το οποίο μπορεί να είναι το <http://www.WorkingOntologist.org/Examples/Countries.owl#England> το οποίο έχει identifier England και namespace geo θα γράφεται geo:England. Σε αυτόν τον σύνδεσμο το namespace geo είναι το αναγνωριστικό του <http://www.WorkingOntologist.org/Examples/Countries.owl#> ενώ το αναγνωριστικό England δηλώνει την οντότητα England που ανήκει στο αρχείο του παραπάνω συνδέσμου. Είναι σημαντικό βέβαια να αναφερθεί ότι τα qnames χρησιμοποιούνται μόνο για την διευκόλυνση αναπαράστασης ενός σημασιολογικού μοντέλου και δεν αποτελούν οικουμενικά αναγνωριστικά στο διαδίκτυο. Τον ρόλο αυτό μπορεί να παίξει μόνο ένα πλήρες URI γι' αυτό και κάθε qname θα πρέπει να συνοδεύεται από μία δήλωση του URI το οποίο αναπαριστά. Το W3C έχει ορίσει κάποια namespaces για τις τεχνολογίες του σημασιολογικού ιστού τα πιο σημαντικά εκ των οποίων είναι τα παρακάτω:

- *rdf*: δηλώνει αναγνωριστικά που χρησιμοποιούνται στο RDF. Ο αριθμός των αναγνωριστικών είναι αρκετά μικρός και χρησιμοποιούνται για να δηλώσουν τύπους και ιδιότητες στα RDF. Το επίσημο URI για το namespace είναι το <http://www.w3.org/1999/02/22-rdfsyntax-ns#>
- *rdfs*: δηλώνει αναγνωριστικά που χρησιμοποιούνται στο RDFS (RDF Schema Language), το οποίο είναι επέκταση του RDF και έχει μεγαλύτερο λεξιλόγιο με περισσότερους σημασιολογικούς περιορισμούς και URI <http://www.w3.org/2000/01/rdf-schema#>
- *owl*: δηλώνει αναγνωριστικά που χρησιμοποιούνται στην OWL (Web Ontology Language), η οποία είναι επέκταση της RDFS και έχει URI: <http://www.w3.org/2002/07/owl#>

Στη συνέχεια θα παραθέσω τις κλάσεις και τις ιδιότητες (properties) των γλωσσών RDF και RDFS δίνοντας βάση σε αυτές τις οποίες χρησιμοποίησα περισσότερο κατά το πρακτικό κομμάτι της διπλωματικής για την κατασκευή της δικής μου οντολογίας καθώς και σε χρήσιμες ιδιότητές τους.

2.2.3.1 RDF

2.2.3.1.1 Κλάσεις (Classes)

- *rdf:XMLLiteral* – the class of XML literal values
- *rdf:Property* – the class of properties
- *rdf:Statement* – the class of RDF statements
- *rdf:Alt*, *rdf:Bag*, *rdf:Seq* – containers of alternatives, unordered containers, and ordered containers (*rdfs:Container* is a super-class of the three)
- *rdf:List* – the class of RDF Lists
- *rdf:nil* – an instance of *rdf:List* representing the empty list

Η πιο σημαντική από τις παραπάνω κλάσεις είναι η *rdf:property* η οποία χρησιμοποιείται σαν τύπος στο RDF για να δείξει πότε κάποιο αναγνωριστικό χρησιμοποιείται σαν ρήμα (predicate) και όχι σαν υποκείμενο ή αντικείμενο. Για παράδειγμα:

:married<*rdf:type*>*rdf:property* που δηλώνει ότι ο τύπος του αναγνωριστικού :married είναι *property* δηλαδή χρησιμοποιείται σαν predicate σε ένα triple.

2.2.3.1.2 Ιδιότητες (Properties)

- *rdf:type* – an instance of *rdf:Property* used to state that a resource is an instance of a class
- *rdf:first* – the first item in the subject RDF list
- *rdf:rest* – the rest of the subject RDF list after *rdf:first*
- *rdf:value* – idiomatic property used for structured values
- *rdf:subject* – the subject of the subject RDF statement
- *rdf:predicate* – the predicate of the subject RDF statement
- *rdf:object* – the object of the subject RDF statement

Με το *rdf:type* δηλώνουμε ότι ένα αντικείμενο είναι μέλος μίας ομάδας. Για παράδειγμα:

: Shakespeare<rdf: type>: Author

: Author<rdf: type>Profession

:Profession<rdf:type>Compensation

Τα παραπάνω τρία triples δηλώνουν διαδοχικά ότι ο Shakespeare ανήκει στην ομάδα συγγραφείς, η έννοια συγγραφέας ανήκει στην ομάδα των επαγγελματιών ενώ τα επαγγέλματα είναι στην ουσία μία μορφή παροχών.

2.2.3.2 RDFS

2.2.3.2.1 Κλάσεις

- *rdfs:Resource* – the class resource, everything
- *rdfs:Literal* – the class of literal values, e.g. strings and integers
- *rdfs:Class* – the class of classes
- *rdfs:Datatype* – the class of RDF datatypes
- *rdfs:Container* – the class of RDF containers
- *rdfs:ContainerMembershipProperty* – the class of container membership properties, *rdf:_1*, *rdf:_2*, ..., all of which are sub-properties of *rdfs:member*

Η διαφορά του *rdfs:Class* με το *rdf:type* είναι ότι το *rdf:type* δηλώνει ότι κάτι είναι μέλος μίας ομάδας ενώ με το *rdfs:Class* δηλώνεται η ίδια η ομάδα. Ο τρόπος με τον οποίο μπορεί να οριστεί μία κλάση φαίνεται στα παρακάτω triples:

: AllStarPlayer<rdf: type>rdfs: Class

: Surgeon<rdf: type>rdfs: Class

: Staff<rdf: type>rdfs: Class

2.2.3.2.2 Ιδιότητες

- *rdfs:subClassOf* – the subject is a subclass of a class
- *rdfs:subPropertyOf* – the subject is a subproperty of a property
- *rdfs:domain* – a domain of the subject property
- *rdfs:range* – a range of the subject property
- *rdfs:label* – a human-readable name for the subject
- *rdfs:comment* – a description of the subject resource
- *rdfs:member* – a member of the subject resource
- *rdfs:seeAlso* – further information about the subject resource
- *rdfs:isDefinedBy* – the definition of the subject resource

Ας δούμε τι σημαίνουν στην πράξη κάποια από τα παραπάνω και κάποιες από τις ιδιότητές τους χρησιμοποιώντας ψευδογλώσσα για να γίνουν πιο κατανοητά.

rdfs:subClassOf

- IF (A *rdfs:subClassOf* B) AND (x *rdf:type* A) THEN (x *rdf:type* B)
- IF [(A *rdfs:subClassOf* C), (B *rdfs:subClassOf* C)] AND [(x *rdf: type* A) OR (x *rdf: type* B)] THEN (x *rdf: type* C)

- IF [(C rdfs:subClassOf A),(C rdfs:subClassOf B)] AND (x rdf:type c) THEN [(x rdf:type A),(x rdf:type B)]

rdfs:subPropertyOf

- IF (P rdfs:subPropertyOf R) AND (A P B) THEN (A R B)
- IF [(P rdfs:subPropertyOf R),(Q rdfs:subPropertyOf R)] AND [(x P y) OR (x Q y)] THEN (x R y)
- IF [(P rdfs:subPropertyOf R),(P rdfs:subPropertyOf S)] AND (x P y) THEN [(x R y),(x S y)]

rdfs:domain και rdfs:range

Το rdfs:domain αναφέρεται στο υποκείμενο ενός triple το οποίο χρησιμοποιεί το P σαν ρήμα (predicate) και το rdfs:range αναφέρεται στο αντικείμενο ενός τέτοιου triple.

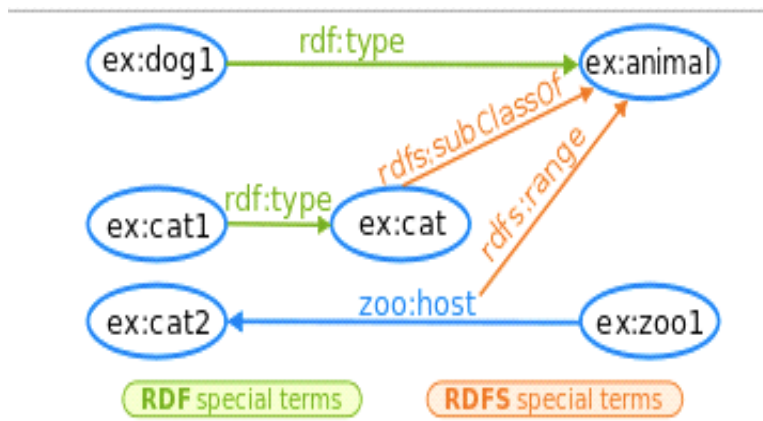
- IF (P rdfs:domain D) AND (x P y) THEN (x rdf:type D)
- IF (P rdfs:range R) AND (x P y) THEN (y rdf:type R)
- IF (P rdfs:domain D) AND (D rdfs:subClassOf C) THEN (P rdfs:domain C)
- IF (P rdfs:range R) AND (R rdfs:subClassOf C) THEN (P rdfs:range C)

Στον παρακάτω απλό γράφο φαίνονται κάποιες από τις παραπάνω classes και properties.

Τα συμπεράσματα που μπορούν να βγουν από τον γράφο του σχήματος 2.4 είναι τα εξής:

- Dog1 is an animal
- Cat1 is a cat
- Cats are animals
- Zoos host animals
- Zoo1 hosts the Cat2

Τα παραπάνω δεδομένα και σχέσεις για να μπορούν να διαβαστούν από μία μηχανή πρέπει να σειριοποιηθούν δηλαδή να μετατραπούν σε μία σειρά απο bytes έτσι ώστε να μπορούν να περνούν εύκολα ως παράμετροι. Η πιο σύννητης σειριοποίηση για δεδομένα τέτοιου είδους λέγεται Turtle και είναι μία συμπαγής σειριοποίηση στην οποία τα δεδομένα εμφανίζονται σε



Σχήμα 2.4: Παράδειγμα RDF γράφου⁴

μορφή φιλική προς το ανθρώπινο μάτι. Η σειριοποίηση των δεδομένων του σχήματος 4 σε Turtle φαίνεται παρακάτω:

- @prefix rdf: <<http://www.w3.org/1999/02/22-rdf-syntax-ns#>> .
- @prefix rdfs: <<http://www.w3.org/2000/01/rdf-schema#>> .
- @prefix ex: <<http://example.org/>> .
- @prefix zoo: <<http://example.org/zoo/>> .
- ex:dog1 rdf:type ex:animal .
- ex:cat1 rdf:type ex:cat .
- ex:cat rdfs:subClassOf ex:animal .
- zoo:host rdfs:range ex:animal .
- ex:zoo1 zoo:host ex:cat2 .

Άλλες σειριοποιήσεις είναι οι ακόλουθες:

- N-Triples. Απλή μορφή σειριοποίησης η οποία είναι εύκολνα αναλυθεί αλλά δεν είναι τόσο συμπαγή όσο η Turtle.
- N-Quads. Υπερσύνολο της N-Triples για τη σειριοποίηση μεγάλου πλήθους RDF γράφων.
- JSON-LD. Σειριοποίηση βασισμένη σε JSON.
- N3 or Notation 3. Μη τυπική σειριοποίηση, παρόμοια με την Turtle, η οποία διαθέτει επιπλέον κανόνες όπως η δυνατότητα ορισμού κανόνων συμπερασμάτων.
- RDF/XML. Σύνταξη βασισμένη σε XML η οποία αποτέλεσε την πρώτη μορφή σειριοποίησης γράφων RDF.

2.2.4 SPARQL Query Language

Η SPARQL είναι η επικρατούσα γλώσσα ερωτημάτων για RDF γράφους, μοιάζει με τη γλώσσα SQL και αποτέλεσε πρόταση του W3C το 2008 (SPARQL 1.0) ενώ η επόμενη έκδοσή της (SPARQL 1.1) εφαρμόστηκε το 2013. Με άλλα λόγια η SPARQL είναι μία σημασιολογική γλώσσα ερωτημάτων για βάσεις δεδομένων η οποία είναι ικανή στο να ανακτά και να διαχειρίζεται δεδομένα τα οποία είναι αποθηκευμένα σε μορφή RDF. Επιπλέον, επιτρέπει σε μία ερώτηση να αποτελείται από μοτίβα triples, συνδιασμούς ερωτήσεων και άλλα

⁴ https://en.wikipedia.org/wiki/RDF_Schema

προαιρετικά μοτίβα. Υπάρχουν αρκετά εργαλεία τα οποία επιτρέπουν σε κάποιον χρήστη να συνδέεται και να κατασκευάζει ερωτήσεις SPARQL σε βάσεις δεδομένων και οντολογίες που έχει ανεβάσει ο ίδιος στην εφαρμογή όπως είναι το ARQ και το Dydra. Τα εργαλεία αυτά διαθέτουν αρκετά default prefixes ώστε ο χρήστης να μπορεί να κάνει ερωτήσεις που περιλαμβάνουν properties αυτών χωρίς να δηλώσει τα URIs τους, κάτι το οποίο εξοικονομεί χώρο στην παρουσίαση της ερώτησης και χρόνο από την πλευρά του χρήστη. Παράλληλα, υπάρχουν και εργαλεία τα οποία μπορούν να μεταφράσουν ερωτήσεις SPARQL σε άλλες γλώσσες ερωτημάτων όπως η SQL και η XQuery.

Υπάρχουν τέσσερις διαφορετικές μορφές ερωτήσεων που επιτρέπει η γλώσσα SPARQL κάθε μία εκ των οποίων εξυπηρετεί διαφορετικό σκοπό.

- **SELECT query:** Χρησιμοποιείται για την εξαγωγή δεδομένων από μία συλλογή και τα αποτελέσματα επιστρέφονται σε μορφή πίνακα όπου κάθε στήλη αναπαριστά κάθε μία από τις μεταβλητές που ζητήθηκαν. Αποτελεί τον τύπο ερώτησης που θα μας απασχολήσει κυρίως σε αυτήν τη διπλωματική.
- **CONSTRUCT query:** Επιστρέφει triples. Επιτρέπει στον χρήστη να τραβήξει triples από μία πηγή δεδομένων χωρίς να τα αλλάξει ή να τραβήξει κάποιες τιμές και να τις χρησιμοποιήσει για να δημιουργήσει καινούρια triples.
- **ASK query:** Η ask ρωτάει αν κάποιο δωθέν γράφημα περιγράφει ένα σύνολο από triples σε ένα συγκεκριμένο δεδομένο γράμμα και επιστρέφει μία μεταβλητή true ή false. Η ερώτηση αυτή είναι πολύ χρήσιμη για τον έλεγχο συνθηκών σε ένα σύνολο δεδομένων.
- **DESCRIBE query:** Ζητάει triples τα οποία περιγράφουν ένα συγκεκριμένο πόρο. Η επιλογή των triples που θα επιστραφούν εξαρτάται από τον επεξεργαστή ερωτήσεων, το οποίο οδηγεί μερικές φορές σε αντιφατικά αποτελέσματα. Για αυτόν τον λόγο η ερώτηση DESCRIBE δεν είναι πολύ δημοφιλής.

Στη συνέχεια θα παρατεθούν κάποια παραδείγματα ερωτημάτων πάνω στην οντολογία foaf(friend-of-a-friend), η οποία περιγράφει ανθρώπους, τις δραστηριότητες τους και τις σχέσεις τους με άλλους ανθρώπους. Τα ερωτήματα που θα μας απασχολήσουν είναι της μορφής SELECT. Η παρακάτω ερώτηση επιστρέφει τα ονόματα και τα e-mails όλων των ατόμων του δεδομένου γράμματος.

PREFIX foaf: <<http://xmlns.com/foaf/0.1/>>

SELECT ?name ?email

WHERE

```
{  
  ?person a foaf:Person .  
  ?person foaf:name ?name .  
  ?person foaf:mbox ?email .  
}
```

Μερικές παρατηρήσεις σχετικά με την παραπάνω ερώτηση. Οι μεταβλητές που ακολουθούν τον όρο SELECT συμβολίζουν στην ουσία τα δεδομένα που θέλουμε να μας επιστρέψει η

ερώτηση,την οποία περιλαμβάνουν οι αγκύλες μετά τον όρο WHERE.Το ρήμα(predicate) της πρώτης πρότασης “a” είναι συντόμευση του RDF property rdf:type.Επίσης,τα υποκείμενα και τα αντικείμενα των triples που συμβολίζονται με ?person,?name,?e-mail θα μπορούσαν να έχουν συμβολιστεί με απλά γράμματα π.χ. ?a,?b, ?c και δεν θα άλλαζε κάτι στο αποτέλεσμα της ερώτησης.Είναι ωστόσο χρήσιμο να συμβολίζουμε τους όρους αυτούς με ονόματα τα οποία διευκολύνουν την ανάγνωση του triple και οδηγούν σε μεγαλύτερη ευκρίνεια.Ένας ακόμα τρόπος για το συμβολισμό των subjects,objects και predicates είναι με τα πλήρη URIs τους κάτι το οποίο δεν θα άλλαζε κάτι στο αποτέλεσμα της ερώτησης,θα επέτρεπε στον χρήστη να μην χρησιμοποιήσει κάποιο prefix αλλά θα έκανε την ερώτηση πιο δυσανάγνωστη και πιο δύσκολη να γραφτεί.Η ίδια ερώτηση χρησιμοποιώντας URIs φαίνεται παρακάτω:

```

SELECT ?name ?email
WHERE
{
  ?person a    <http://xmlns.com/foaf/0.1#Person>.
  ?person <http://xmlns.com/foaf/0.1#name> ?name .
  ?person <http://xmlns.com/foaf/0.1#mbox> ?email .
}

```

Σε κάθε περίπτωση η ερώτηση αυτή θα επιστρέψει έναν πίνακα δύο στηλών ο οποίος αντιστοιχεί τα ονόματα των ατόμων που περιλαμβάνει η βάση δεδομένων με τα e-mail τους.Στην ίδια ερώτηση η γραμμή SELECT θα μπορούσε να είναι της μορφής: SELECT ?name ?email FROM <example.ttl>.Η λέξη FROM επιτρέπει στον χρήστη να επιλέξει ένα συγκεκριμένο αρχείο για να ρωτήσει.Από την άλλη,αν ο χρήστης θέλει να βρει το e-mail ενός συγκεκριμένου ατόμου του οποίου ξέρει το όνομα,για παράδειγμα Nick η ερώτηση θα έχει την παρακάτω μορφή:

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?email
WHERE
{
  ?person a    foaf:Person .
  ?person foaf:name "Nick" .
  ?person foaf:mbox ?email .
}

```

Επιπλέον,αν ήθελε να μάθει τα πάντα για τον Nick,δηλαδή όλα τα triples στα οποία το άτομο Nick είναι subject θα έκανε την εξής ερώτηση:

```

PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?propertyname ?propertyvalue
WHERE
{
  ?person a    foaf:Person .
  ?person foaf:name "Nick" .
  ?person ?propertyname ?propertyvalue .
}

```

Μία χρήσιμη ιδιότητα της SPARQL είναι πως όταν χρησιμοποιούμε το ίδιο subject σε συνεχόμενα triples, το subject μπορεί να παραληφθεί. Για παράδειγμα η προηγούμενη ερώτηση γράφεται:

```
?person a foaf:Person .
      foaf:name "Nick" .
?propertyname ?propertyvalue .
```

Τέλος το σύμβολο * δίπλα από την εντολή Select δηλώνει στην SPARQL ότι θέλει να επιστραφούν στα αποτελέσματα όλα τα subjects και objects των triples που υπάρχουν ανάμεσα στις αγκύλες.

Στη συνέχεια θα παραθέσω κάποιες χρήσιμες εντολές και συναρτήσεις της SPARQL, καθώς και τον τρόπο που αυτές συντάσσονται, οι οποίες επιτρέπουν στον χρήστη να κάνει πιο πολύπλοκες ερωτήσεις πάνω σε μία βάση δεδομένων.

DISTINCT:

Η εντολή Distinct χρησιμοποιείται πριν τη δήλωση μίας μεταβλητής στην ενότητα του Select και απαγορεύει την επιστροφή ενός διπλού αποτελέσματος. Αν δηλαδή σε μία βάση δεδομένων ένα όνομα είναι καταχωρημένο πάνω από μία φορά χωρίς την εντολή Distinct η SPARQL θα επέστρεφε όλα τα Instances του.

Σύνταξη:

```
SELECT DISTINCT ?propertyname
```

UNION:

Η εντολή Union επιτρέπει στον χρήστη να ορίσει περισσότερα από ένα διαφορετικά triples και στη συνέχεια να ζητήσει ένα συνδυασμό όλων των δεδομένων που τα ικανοποιούν. Ο πίνακας αποτελεσμάτων έχει στήλες για όλες τις μεταβλητές που έχουν οριστεί.

Σύνταξη:

```
SELECT *
WHERE {
  { ?Person foaf:name "Nick"
    ?Person foaf:mbox ?email }
  UNION
  { ?Person foaf:name "Andrew"
    ?Person foaf:mbox ?email }
}
```

Η παραπάνω ερώτηση θα επιστρέψει τα e-mail όλων των ατόμων που έχουν όνομα Nick ή Andrew.

FILTER:

Μία πρόταση η οποία εκφράζεται μέσα στην εντολή Filter θέτει έναν περιορισμό στα αποτελέσματα του συνόλου των εντολών μέσα στις οποίες εμφανίζεται το φίλτρο. Η έκφραση του φίλτρου μπορεί να είναι όσο πολύπλοκη θέλει ο χρήστης αρκεί να επιστρέφει μία λογική τιμή. Ακολουθούν δύο παραδείγματα σύνταξης της εντολής με διαφορετικές εκφράσεις μέσα στο φίλτρο.

Σύνταξη:

?Person foaf:name ?name

?Person foaf:mbox ?e-mail

FILTER (?name = "Nick")

Επιστρέφει μόνο τα e-mail των ατόμων που έχουν όνομα Nick.

?item :cost ?price

FILTER (?price > 10)

Επιστρέφει όλα τα αντικείμενα που έχουν τιμή μεγαλύτερη του 10.

LIMIT:

Η εντολή Limit είναι αρκετά απλή. Δηλώνεται στο τέλος της ερώτησης και δέχεται ως όρισμα έναν ακέραιο αριθμό ο οποίος συμβολίζει το πλήθος των αποτελεσμάτων που θέλουμε να επιστραφούν. Για παράδειγμα με την εντολή LIMIT 2 θα επιστραφούν δύο αποτελέσματα.

GROUP BY:

Επιτρέπει τον διαχωρισμό των δεδομένων ανάλογα με την μεταβλητή που δηλώνει ο χρήστης. Για παράδειγμα η σύνταξη GROUP BY ?name θα ομαδοποιήσει τα δεδομένα ανάλογα με την τιμή του ονόματος του χρήστη.

ORDER BY:

Επιστρέφει αριθμητικά δεδομένα με αύξουσα σειρά. Για να εμφανίζονται τα δεδομένα με φθίνουσα σειρά χρησιμοποιείται η εντολή ORDER BY DESC. Για παράδειγμα η εντολή ORDER BY (?price) εμφανίζει στα αποτελέσματα τα αντικείμενα από το πιο φθινό στο πιο ακριβό.

MAX/MIN:

Συνάρτηση η οποία επιστρέφει τη μέγιστη/ελάχιστη αριθμητική τιμή μίας μεταβλητής. Για παράδειγμα οι SELECT {Max(?price) as ?Maxprice}, SELECT {Min(?price) as ?Minprice} θα επιστρέψουν το αντικείμενα με τη μεγαλύτερη/μικρότερη τιμή.

SUM/AVG:

Συνάρτηση που επιστρέφει το άθροισμα/μέσο όρο των τιμών μίας μεταβλητής η οποία συμβολίζει αριθμητικές τιμές. Η σύνταξη αυτών των συναρτήσεων ακολουθεί την ίδια λογική με τις συναρτήσεις MAX/MIN οι οποίες εξηγήθηκαν παραπάνω. Για παράδειγμα SELECT {SUM(?price) as ?Sumprice}, SELECT {AVG(?price) as ?Avgprice}.

COUNT:

Συνάρτηση που επιστρέφει πόσες φορές κάποιο συγκεκριμένο instance μίας μεταβλητής εμφανίζεται στη βάση δεδομένων. Για παράδειγμα η σύνταξη:

```
SELECT ?name (COUNT(?name) as ?nametotal)
```

```
WHERE{
```

```
  ?Person foaf:name ?name
```

```
}
```

```
FILTER (?name = "Nick")
```

Θα επιστρέψει πόσες φορές εμφανίζεται το όνομα "Nick" στη βάση δεδομένων.

IF:

Λογική συνάρτηση η οποία αξιολογεί μία η περισσότερες εκφράσεις και επιστρέφει μία τιμή ανάλογα με την τιμή της έκφρασης. Η If δέχεται τρία ορίσματα. Αν το πρώτο όρισμα αξιολογηθεί ως true επιστρέφει την τιμή του δεύτερου ορίσματος διαφορετικά επιστρέφει την τιμή του τρίτου ορίσματος. Για παράδειγμα η έκφραση:

```
((IF (2 > 3, "Two is bigger", "Three is bigger"))) AS? answer
```

Υπολογίζει την τιμή της λογικής πρότασης $2 > 3$. Δεδομένου ότι η πρόταση αποτιμάται ως `false` το query θα επιστρέψει την πρόταση "Three is bigger".

3 Σχετικές Εργασίες

Τα τελευταία χρόνια έχει εκπονηθεί ένας μεγάλος αριθμός εργασιών οι οποίες χρησιμοποιούν κοινωνικο-θεωρητικές τεχνικές σε συνδυασμό με υπολογιστικά μοντέλα για τη συλλογή και την κατανόηση πληροφοριών που σχετίζονται με τις ανθρώπινες δραστηριότητες. Κάποιες από αυτές τις εργασίες υποστηρίζουν μικρές ομάδες δραστηριοτήτων και χρησιμοποιούν τη θεωρία δραστηριοτήτων σε συνδυασμό με σημασιολογικά μοντέλα για να κατανοήσουν τα δεδομένα τα οποία λαμβάνουν από αυτές τις δραστηριότητες. Άλλες εργασίες δίνουν μεγαλύτερη έμφαση στα δεδομένα κοινωνικής δικτύωσης με παράλληλη ανάπτυξη εφαρμογών για την καταχώρηση, μοντελοποίηση και εμφάνιση αυτών των δεδομένων. Σε αυτό το κεφάλαιο θα παρουσιαστούν κάποιες εργασίες οι οποίες έχουν στόχο την καταγραφή και παρακολούθηση δραστηριοτήτων έχοντας ως θεωρητική βάση την θεωρία δραστηριοτήτων και χρησιμοποιώντας για μοντελοποίηση σημασιολογικά μοντέλα. Οι εργασίες αυτές επηρέασαν την ανάπτυξη της διπλωματικής εργασίας άλλες σε μεγαλύτερο και άλλες σε μικρότερο βαθμό. Επιπλέον θα παρουσιαστούν οι ομοιότητες και οι διαφορές τους με την παρούσα διπλωματική. Η Θεωρία Δραστηριοτήτων θεωρήθηκε σαν η επικρατέστερη θεωρία μοντελοποίησης και κατανόησης της ανθρώπινης δραστηριότητας στον χώρο της τεχνολογίας, και κάτω από την ομπρέλα της έγινε η όποια ανάλυση του χώρου και η ανάπτυξη της μεθοδολογίας της συγκεκριμένης εργασίας.

3.1 Κατανοώντας τα Ψηφιακά Αποτυπώματα: Μία Οντολογική Προσέγγιση της Θεωρίας Δραστηριοτήτων (Making Sense of Digital Traces: An Activity Theory Driven Ontological Approach)

Η εργασία αυτή εκπονήθηκε από τους ερευνητές Stan Karanasios, Dhavalkumar Thakker, Lydia Lau, David Allen, Vania Dimitrova και Alistair Norman του πανεπιστημίου του Leeds. Στην εργασία αυτή οι ερευνητές χρησιμοποιούν τη θεωρία δραστηριοτήτων σε συνδυασμό με οντολογικά μοντέλα για να εξηγήσουν ψηφιακά ίχνη(digital traces).Υποστηρίζουν ότι η θεωρία δραστηριοτήτων μπορεί να χρησιμοποιηθεί για την ανάπτυξη πιο διορατικών μοντέλων δραστηριοτήτων οι οποίες δεν είναι καλά ορισμένες, τα οποία στη συνέχεια μπορούν να χρησιμοποιηθούν για την ανάπτυξη μίας οντολογίας. Χρησιμοποιώντας την οντολογία AMOn η οποία θεωρητικά είναι πολύ κοντά με την Activity theory αναπτύχθηκε μοντέλο τριών επιπέδων και εξηγήθηκε μέσα από ένα παράδειγμα πως μπορεί να χρησιμοποιηθεί για να βοηθήσει ένα άτομο να κατανοήσει τη διαδικτική επικοινωνία σε μία κατάσταση συνέντευξης για εργασία.Για τη μοντελοποίηση αυτής της δραστηριότητας συλλέχθηκαν δεδομένα από έναν μεγάλο αριθμό πηγών.Τέλος αξιολογήθηκε η οντολογία AMOn ως η κυρίαρχη οντολογία η οποία χρησιμοποιήθηκε για τη μοντελοποίηση δραστηριοτήτων.

Κυρίαρχη διαφορά με την παρούσα διπλωματική είναι ότι χρησιμοποιήθηκε μόνο μία οντολογία για την κατασκευή του μοντέλου καθώς και ο ίδιος ο σκοπός της εργασίας ήταν να δείξει ότι η συγκεκριμένη διαδικασία μπορεί να χρησιμοποιηθεί για να εξηγηθούν και να κατανοηθούν τα digital traces.Στην παρούσα διπλωματική ωστόσο έγινε χρήση και συνδυασμός διαφορετικών οντολογιών ενώ χρησιμοποιήθηκαν οντολογίες οι οποίες έχουν σκοπό την μοντελοποίηση και εξήγηση δεδομένων τα οποία λαμβάνονται από κοινωνικά

δίκτυα. Παράλληλα έγιναν ερωτήσεις πάνω στα δεδομένα χρησιμοποιώντας τη γλώσσα SPARQL ενώ τέλος αναπτύχθηκε εφαρμογή παρουσίασης των αποτελεσμάτων. Ωστόσο είναι γεγονός πως και οι δύο εργασίες εκπονήθηκαν πάνω στην ίδια λογική: τη χρήση οντολογιών για μοντελοποίηση δεδομένων τα οποία προέρχονται από ψηφιακές πηγές έχοντας ως θεωρητικό υπόβαθρο τη θεωρία δραστηριοτήτων.

3.2 Ανάλυση και Μοντελοποίηση Ανθρωπίνων Δραστηριοτήτων στα Πληροφοριακά Συστήματα, και Ανάπτυξη Πρότυπης Εφαρμογής για Καταγραφή, Παρουσίαση και Παρακολούθηση Προσωπικών Δραστηριοτήτων

Η εφαρμογή αυτή αναπτύχθηκε από τον υποψήφιο διδάκτορα του εργαστηρίου συστημάτων αποφάσεων της σχολής ΗΜΜΥ στο πλαίσιο της διπλωματικής του εργασίας. Σκοπός της διπλωματικής ήταν η δημιουργία ενός γενικού μοντέλου ικανού για τη γενική περιγραφή όλων των ανθρώπινων δραστηριοτήτων και η χρήση του στην ανάπτυξη εφαρμογής με όνομα Activity Tracker. Η εφαρμογή αυτή στοχεύει στην καταγραφή και παρακολούθηση καθημερινών δραστηριοτήτων του χρήστη και στη διαγραμματική παρουσίαση των αποτελεσμάτων. Με τον τρόπο αυτό ο χρήστης της εφαρμογής μπορεί να παρακολουθεί και να ελέγχει την καθημερινότητά του και την αλληλεπίδραση των δραστηριοτήτων μεταξύ τους. Ως αποτέλεσμα ο χρήστης μπορεί να κάνει καλύτερη κατανομή του χρόνου του και να λαμβάνει ενδιαφέρουσες πληροφορίες σχετικές με το κοινωνικό του πλαίσιο, το βαθμό επίτευξης των στόχων του και την πορεία της ζωής του στην πάροδο του χρόνου. Τέλος η εισαγωγή των δραστηριοτήτων γίνεται μέσω τρίτων υπηρεσιών όπως για παράδειγμα το Runkeeper γεγονός το οποίο ενισχύει τον αυτοματισμό της διαδικασίας.

Η διαφορά της διπλωματικής αυτής με την παρούσα είναι ότι δόθηκε μεγαλύτερη βάση στην ανάπτυξη της εφαρμογής και λιγότερη στη μελέτη οντολογιών και στη σύνδεση μεταξύ τους. Η εφαρμογή που αναπτύχθηκε διαθέτει περισσότερα features καθώς ήταν και το κύριο αντικείμενο της διπλωματικής. Από την άλλη το κύριο αντικείμενο της παρούσας διπλωματικής ήταν η σύνδεση οντολογιών για τη δημιουργία ενός γενικού μοντέλου και η χρήση της γλώσσας SPARQL για την πραγματοποίηση ερωτήσεων πάνω στην οντολογία ενώ η εφαρμογή αναπτύχθηκε για την παρουσίαση των αποτελεσμάτων. Επίσης στην εφαρμογή Activity Tracker τα δεδομένα συλλέγονται από τον ίδιο τον χρήστη ενώ στην εφαρμογή που ανέπτυξα τα δεδομένα λαμβάνονται από τη βάση δεδομένων του Activity Tracker και συνδέονται πάνω στην οντολογία για τη δημιουργία του τελικού μοντέλου.

3.3 Οντολογία Θεωρίας Δραστηριοτήτων για Διαμοιρασμό Γνώσης σε Συστήματα Υγείας (Activity Theory Ontology for Knowledge Sharing in E-Health)

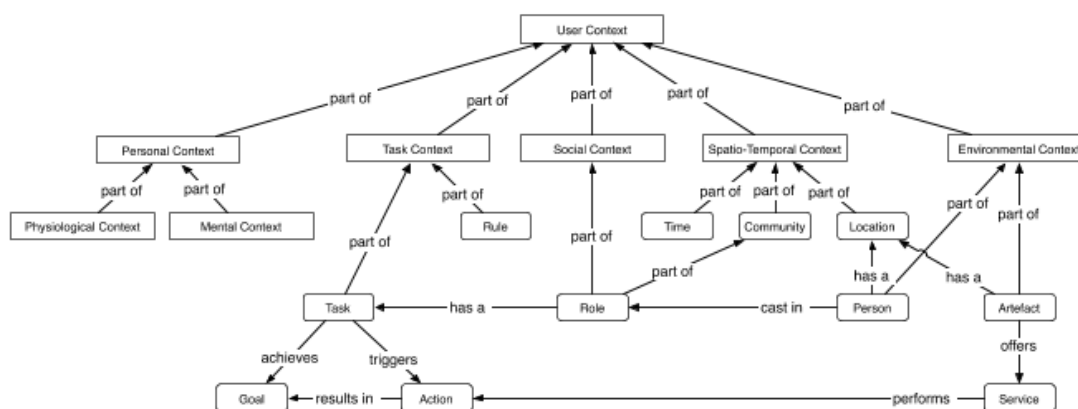
Αυτή η εργασία εκπονήθηκε από τους Gouhua Bai και Yang Guo του Blekinge Institute of Technology, ερευνά οντολογικά μοντέλα για τον διαμοιρασμό γνώσης και βασίζεται στη θεωρία δραστηριοτήτων. Σκοπός της εργασίας ήταν η παροχή μίας πλατφόρμας για συστήματα e-health η οποία να μπορεί να χρησιμοποιηθεί για να ενσωματώσει και να μοιραστεί τη γνώση ανάμεσα σε όλους τους εμπλεκόμενους σε μία ιατροφαρμακευτική

δραστηριότητα. Βασικός άξονας της εργασίας είναι η επίλυση των προβλημάτων όσον αφορά τον διαμοιρασμό γνώσης σε συστήματα E-health τα οποία επιλύονται χρησιμοποιώντας το μοντέλο του Engestrom για τη θεωρία δραστηριοτήτων.

Η ομοιότητα αυτής της εργασίας με την παρούσα διπλωματική έγκειται στο γεγονός ότι χρησιμοποιήθηκε η θεωρία δραστηριοτήτων για την κατασκευή ενός οντολογικού μοντέλου το οποίο μπορεί να χρησιμοποιηθεί για να εξηγήσει μία δραστηριότητα. Ωστόσο, περιορίζεται σε δραστηριότητες E-health και δεν γίνεται εις βάθος μελέτη πάνω σε διαφορετικές οντολογίες. Όπως και στην εργασία του υποκεφαλαίου 2.1 αρκείται στη χρήση του μοντέλου του Engestrom το οποίο θεωρεί ικανό να μοντελοποιήσει μία δραστηριότητα και το χρησιμοποιεί ως βάση για τη σχεδίαση της πλατφόρμας E-health.

3.4 Χρησιμοποιώντας τη Θεωρία Δραστηριοτήτων για Μοντελοποίηση του Γενικού Πλαισίου Γνώσης (Using Activity Theory to Model Context Awareness)

Η εργασία αυτή εκπονήθηκε από τους Anders Kofod-Petersen και Jorg Cassens του Norwegian University of Science and Technology και προτείνει τη χρήση της θεωρίας δραστηριοτήτων για να μοντελοποιήσει context και στη συνέχεια να γεμίσει το μοντέλο ώστε να μπορεί να αξιολογεί καταστάσεις σε pervasive υπολογιστικά περιβάλλοντα. Η μοντελοποίηση πραγματοποιήθηκε με το σχεδιασμό οντολογίας από τους ερευνητές η οποία έχει ως βάση το μοντέλο Activity Theory του Engestrom και συμπληρώνεται με classes σχετικές με context σε pervasive υπολογιστικά συστήματα. Στη συνέχεια πραγματοποιήθηκε mapping από το μοντέλο της θεωρίας δραστηριοτήτων σε επίπεδο context και προτάθηκε ένα τελικό μοντέλο το οποίο φαίνεται παρακάτω:



Εικόνα 3.1: Μοντέλο της δομής πληροφορίας πλαισίου (context) – Anders Kofod-Petersen, Jorg Cassens(2006). Using Activity Theory to Model Context Awareness

Η συγκεκριμένη εργασία έχει αρκετές ομοιότητες σε σχέση με την παρούσα διπλωματική. Κάνει χρήση της θεωρίας δραστηριοτήτων για να δημιουργήσει ένα μοντέλο σχετικό με context σε pervasive υπολογιστικά περιβάλλοντα. Ωστόσο κάνει χρήση του μοντέλου του Engestrom αλλάζοντας τις κλάσεις ώστε η τελική οντολογία να έχει νόημα σε επίπεδο context και δεν γίνεται χρήση ξεχωριστής οντολογίας για pervasive συστήματα. Ως αποτέλεσμα το μοντέλο το οποίο τελικά προτείνεται είναι αρκετά μικρότερο από αυτό στο οποίο κατέληξα στην διπλωματική μου, το οποίο είναι λογικό καθώς σκοπός της εργασίας

είναι να δείξει ότι ο σχεδιασμός context-aware συστημάτων μπορεί να επωφεληθεί από τη χρήση των κοινωνικών μοντέλων του Activity Theory.

4 Πειραματική Ανάλυση Δεδομένων

Στα υποκεφάλαια που ακολουθούν θα γίνει ανάλυση και περιγραφή των οντολογιών που επιλέχθηκαν, η σύνδεση μεταξύ τους καθώς και οι SPARQL ερωτήσεις που έγιναν πάνω στην τελική οντολογία.

4.1 Επιλογή Οντολογιών

Σε αυτό το υποκεφάλαιο γίνεται περιγραφή των οντολογιών που χρησιμοποιήσα και εξηγείται γιατί επέλεξα τις συγκεκριμένες οντολογίες. Η λογική που ακολούθησα στην επιλογή των κατάλληλων οντολογιών που θα συνέθεταν την τελική οντολογία, έγκειται στην ανάγκη να αποτελείται από τρία επίπεδα. Για το πρώτο και υψηλότερο επίπεδο χρειαζόμουν μία οντολογία η οποία να προσεγγίζει αρκετά καλά τη φιλοσοφία της θεωρίας δραστηριοτήτων. Για το δεύτερο και μεσαίο επίπεδο διέκρινα την ανάγκη να υπάρχουν δύο οντολογίες. Μία pervasive οντολογία η οποία να διαθέτει κλάσεις σχετικές με τον online χρήστη και τα υπολογιστικά εργαλεία που χρησιμοποιεί και μία οντολογία που να περιγράφει ικανοποιητικά αντικείμενα που αφορούν τα κοινωνικά δίκτυα. Για το χαμηλότερο επίπεδο μελέτησα οντολογίες οι οποίες μπορούσαν να δώσουν περισσότερες λεπτομέρειες στις κλάσεις των υψηλότερων επιπέδων και να κάνουν εν τέλει την οντολογία μου πιο λεπτομερή.

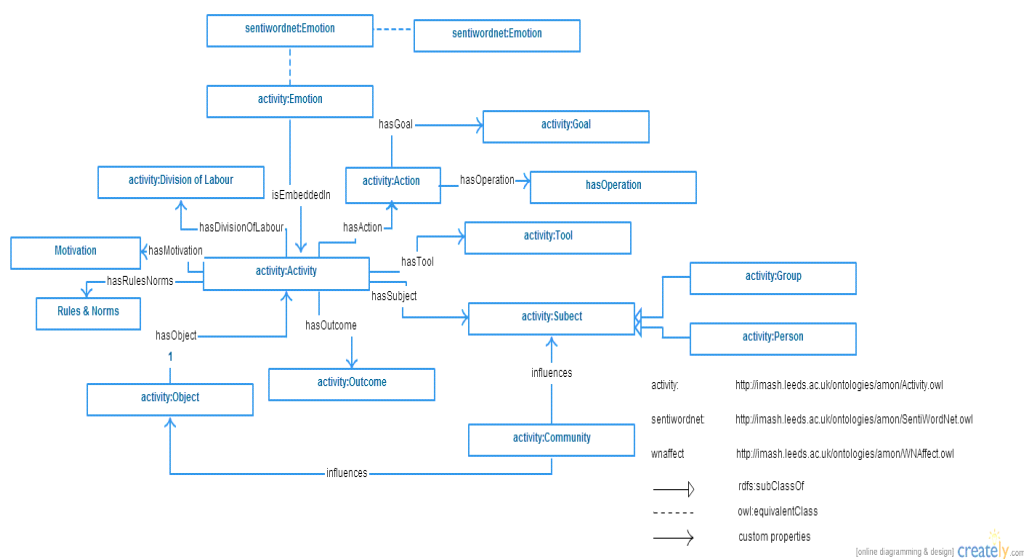
Η τελική οντολογία στην οποία ήθελα να καταλήξω έπρεπε να περιγράφει το μέσο χρήστη του Internet από τη σκοπιά της κοινωνικής θεωρίας και να περιλαμβάνει όλες τις αναγκαίες κλάσεις που θα χρειαζόμουν ώστε να συνδέσω πάνω στην οντολογία τα πραγματικά δεδομένα από την εφαρμογή Activity tracker ώστε να μπορώ της κάνω ερωτήσεις και να βγάλω χρήσιμα συμπεράσματα. Η εφαρμογή Activity Tracker αναπτύχθηκε από τον υποψήφιο διδακτορικό του εργαστηρίου Συστημάτων Αποφάσεων Ευάγγελο Αρβανιτάκη στο πλαίσιο της διπλωματικής του εργασίας και στοχεύει στην καταγραφή και παρακολούθηση καθημερινών δραστηριοτήτων των χρηστών. Περιγράφει ένα μεγάλο σύνολο δραστηριοτήτων που περιλαμβάνουν δραστηριότητες κοινωνικής δικτύωσης, αθλητικές δραστηριότητες όπως τρέξιμο και δραστηριότητες ρουτίνας (ξεκούραση, φαγητό κτλ.).

4.1.1 Υψηλό επίπεδο

Η οντολογία που επέλεξα για το υψηλό επίπεδο είναι η AMOn (Activity Model Ontology) διότι πρόκειται για μία οντολογία η οποία δημιουργήθηκε έχοντας ως βάση την θεωρία δραστηριοτήτων. Υπάρχουν και άλλες οντολογίες οι οποίες βασίζονται στις έννοιες της Activity Theory όπως οι REA, ARIS και Enterprise Ontology, ωστόσο έχουν δημιουργηθεί με σκοπό να περιγράψουν δραστηριότητες σε συγκεκριμένους επιχειρηματικούς τομείς, ενώ η AMOn αποτελεί ένα πολύ καλό γενικό οντολογικό μοντέλο της θεωρίας δραστηριοτήτων. Μάλιστα οι περιγραφές των κλάσεων της οντολογίας περιέχουν ορισμούς από την ίδια τη θεωρία δραστηριοτήτων και οι σχέσεις μεταξύ τους ακολουθούν το μοντέλο του Engeström. Η AMOn αποτελείται από τρεις μονάδες, τη μονάδα δραστηριοτήτων τις Activity module, Interpersonal Communication module και Cultural Aspects in Interpersonal Communication module.

Η πρώτη από αυτές αποτελείται από βασικές έννοιες της θεωρίας δραστηριοτήτων όπως SUBJECT, OBJECT, TOOLS, MOTIVATION, OUTCOME, COMMUNITY, DIVISION OF LABOUR, RULE, NORM, ACTION, OPERATION και τις σχέσεις που τις συνδέουν όπως για παράδειγμα

Action 'is part of' some Activity, Action hasOperation some Operation, Action hasGoal some Goal, Activity hasMotivation some Motivation. Οι σχέσεις αυτές περιγράφουν ακριβώς τη θεωρία δραστηριοτήτων όπως την περιέγραψα στο θεωρητικό κομμάτι της διπλωματικής. Το διάγραμμα της μονάδας δραστηριοτήτων η οποία στην ουσία είναι και η κύρια μονάδα με την οποία ασχολήθηκα όσον αφορά την οντολογία AMOn φαίνεται στο σχήμα 4.1. Η δεύτερη μονάδα περιλαμβάνει έννοιες που επεκτείνουν τις βασικές έννοιες του Activity module. Για παράδειγμα η οντότητα Tool επεκτείνεται ώστε να περιλαμβάνει Mental Tools (Verbal Communication, Non Verbal Communication, Body Language etc.) και Physical Tools (CV, Clothing etc.). Τέλος το Cultural Aspects Module περιλαμβάνει έννοιες από διάφορες πολιτισμικές θεωρίες και τις συνδέει στη δεύτερη μονάδα όπου αυτό κρίνεται σωστό.



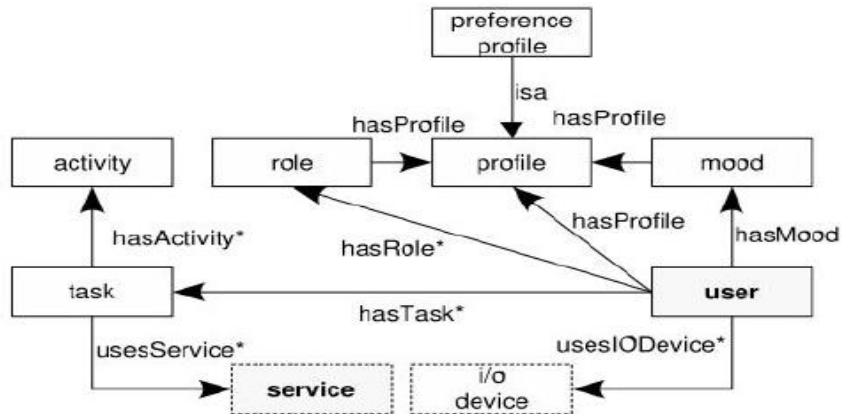
Σχήμα 4.1: Διάγραμμα του activity module της AMOn

4.1.2 Μεσαίο επίπεδο

Όπως προαναφέρθηκε για το μεσαίο επίπεδο χρειάστηκα αρχικά μία pervasive οντολογία η οποία διαθέτει κλάσεις για τον online χρήστη και τις συσκευές που χρησιμοποιεί για να αποκτήσει πρόσβαση σε online υπηρεσίες. Μελέτησα οντολογίες όπως η CC/PP, COBRA-ONT, CoDAMoS και Delivery Context. Από αυτές η CC/PP διαθέτει κλάσεις που περιγράφουν τις προτιμήσεις του χρήστη και τις δυνατότητες των συσκευών που χρησιμοποιεί τις οποίες διακρίνει σε hardware, software και application. Από την άλλη η Delivery Context, παρέχει ένα μοντέλο το οποίο περιγράφει την επικοινωνία μίας συσκευής και μίας υπηρεσίας αλλά δεν διαθέτει κλάσεις οι οποίες αντιπροσωπεύουν τον χρήστη. Είναι προφανές ότι καμία από αυτές τις οντολογίες δεν περιλαμβάνουν αρκετές κλάσεις που να εξυπηρετούν τον σκοπό της διπλωματικής και επιπλέον δεν ταιριάζουν με τη θεωρία δραστηριοτήτων. Επιπλέον, η COBRA-ONT διαθέτει αντικείμενα που περιγράφουν τις υπηρεσίες στις οποίες θέλει να έχει πρόσβαση ο χρήστης. Εν τέλει κατέληξα στην οντολογία CoDAMoS, η οποία μπορεί να μην είναι τόσο αναλυτική στις έννοιες που περιγράφει όσο κάποιες από τις υπόλοιπες, ωστόσο περιλαμβάνει όλα τα αντικείμενα που θεωρήσα ότι πρέπει να υπάρχουν στην pervasive

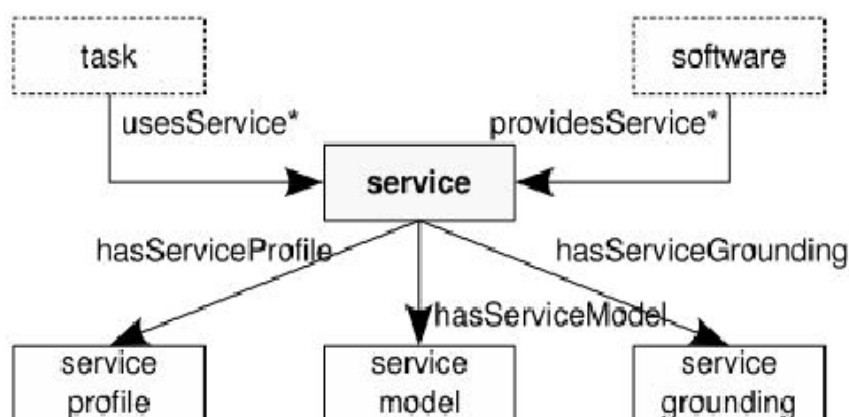
οντολογία που θα χρησιμοποιήσω και ταιριάζει σε ικανοποιητικό βαθμό με την Activity Theory.

Η CoDAMoS ορίζει τέσσερις κύριες οντότητες: χρήστη, περιβάλλον, πλατφόρμα και υπηρεσίες και έχει σχεδιαστεί με σκοπό να περιγράψει το χρήστη μίας σύγχρονης εφαρμογής, καθώς και τις συσκευές και διεπαφές που χρησιμοποιεί. Παρακάτω θα περιγραφεί και με τη βοήθεια σχημάτων η μορφή της οντολογίας. Στο σχήμα 4.2 φαίνονται οι κύριες κλάσεις της οντολογίας. Παρατηρούμε ότι η



Σχήμα 4.2: Διάγραμμα των κύριων κλάσεων της CoDAMoS

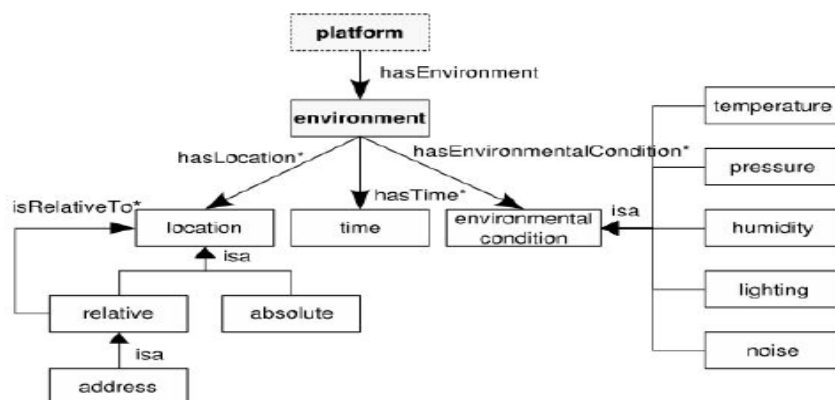
CoDAMoS περιγράφει πολύ καλά την έννοια του χρήστη χρησιμοποιώντας οντότητες για να δείξει το profile του, την διάθεσή του (η οποία είναι και κοντινή έννοια με την Emotion της AMOn), τις συσκευές που χρησιμοποιεί για να πραγματοποιήσει τις εργασίες του. Στο Σχήμα 4.3 φαίνονται οι σχέσεις που έχουν οι υπηρεσίες που χρησιμοποιεί ο χρήστης με τις εργασίες που πραγματοποιεί και το software που χρησιμοποιεί. Οι εργασίες που πραγματοποιεί ο χρήστης για παράδειγμα η επίσκεψη στην ιστοσελίδα του Facebook χρησιμοποιούν κάποια υπηρεσία (στο παράδειγμά μας το ίδιο το Facebook) την οποία παρέχει κάποιο λογισμικό. Κάθε υπηρεσία έχει κάποιες βάσεις, ένα μοντέλο και ένα προφίλ.



Σχήμα 4.3: Σχέσεις της κλάσης service στην CoDAMoS

Τέλος στο Σχήμα 4.4 φαίνεται η σχέση που έχει η πλατφόρμα του χρήστη, η οποία περιλαμβάνει το hardware του χρήστη (μνήμη, CPU, συσκευές εισόδου-εξόδου κτλ) με το περιβάλλον. Είναι προφανές ότι οι περιβαλλοντικές συνθήκες μπορούν να επηρεάσουν την

απόδοση της πλατφόρμας του χρήστη και κατ'επέκταση την εμπειρία του όταν προσπαθεί να χρησιμοποιήσει κάποια υπηρεσία. Για παράδειγμα υψηλές θερμοκρασίες μπορούν να υπερθερμάνουν τον υπολογιστή και να εμποδίσουν τον χρήστη να πραγματοποιήσει τις εργασίες του με ταχύτητα και άνεση.



Σχήμα 4.4: Σχέση πλατφόρμας-περιβάλλοντος στην CoDAMoS

Για τη δεύτερη οντολογία του μεσαίου επιπέδου μελέτησα social network οντολογίες, συγκεκριμένα τις FOAF, UPOS και Activity Streams. Η FOAF ασχολείται με τη σύνδεση ανθρώπων και πληροφοριών, ανεξάρτητα αν οι πληροφορίες είναι ψηφιακές ή πραγματικά δεδομένα. Διακρίνεται σε τρεις διαφορετικούς τύπους συνδέσεων: σύνδεση κοινωνικών δικτύων, φιλίες και συνεργασίες. Πρόκειται για μία γενικότερη οντολογία η οποία δεν επικεντρώνεται στις έννοιες των κοινωνικών δικτύων και για αυτόν τον λόγο δεν την επέλεξα. Η UPOS στηρίζεται πάνω στην FOAF και αντιμετωπίζει θέματα σχετικά με το profile ενός χρήστη και ασχολείται με τη δημιουργία υπο-προφίλ ανάλογα με την εκάστοτε κατάσταση του χρήστη. Η έρευνα που πραγματοποίησα ωστόσο, με οδήγησε στο συμπέρασμα ότι η Activity Streams 2.0 είναι η καλύτερη social network οντολογία με πολλές εφαρμογές σε projects μεγάλων εταιρειών και αποτελεί μονόδρομο για όλους τους ερευνητές που θέλουν να παρουσιάσουν δεδομένα κοινωνικής δικτύωσης σε σημασιολογική μορφή.

Το λεξιλόγιο της Activity Streams 2.0 ορίζει ένα σύνολο από αφηρημένα classes και properties τα οποία περιγράφουν δραστηριότητες και διαχωρίζονται σε δύο ομάδες.

1. Μία βασική ομάδα από properties τα οποία περιγράφουν τη γενική δομή μίας δραστηριότητας και
2. Μία διευρυμένη ομάδα από properties τα οποία καλύπτουν συγκεκριμένους τύπους δραστηριοτήτων και αντικειμένων τα οποία είναι κοινά σε αρκετές εφαρμογές ιστού.

Οι κύριες κλάσεις της Activity Streams παρέχουν τη βάση για το υπόλοιπο λεξιλόγιο της οντολογίας και περιλαμβάνουν τις ακόλουθες κλάσεις:

- 1.Object:** Περιγράφει ένα αντικείμενο οποιοδήποτε είδους. Είναι η κλάση που αποτελεί τη βάση για όλα τα αντικείμενα που ορίζονται από το λεξιλόγιο της οντολογίας.
- 2.Link:** Είναι μία έμμεση αναφορά σε έναν πόρο ο οποίος ορίζεται από ένα URL δημιουργεί δηλαδή τη σχέση που συνδέει ένα αντικείμενο της οντολογίας με τη διεύθυνση που έχει στο Internet.

3.Actor: Υποκλάση του Object η οποία περιγράφει οντότητες ικανές να πραγματοποιήσουν κάποια δραστηριότητα.

4.Activity: Αποτελεί υποκλάση του Object και περιγράφει κάποια μορφή δραστηριότητας η οποία συμβαίνει.Χρησιμοποιεί σαν μία αφηρημένη κλάση για όλες τις δραστηριότητες που ορίζονται στην οντολογία διότι από μόνη της δεν διαθέτει συγκεκριμένες λεπτομέρειες για το είδος της δραστηριότητας που πραγματοποιείται.

5.Intransitive Activity: Είναι υποκλάση της Activity και περιγράφει δραστηριότητες οι οποίες δεν δρουν πάνω σε κάποιο υποκείμενο.Με άλλα λόγια το property “object” το οποίο δηλώνει το υποκείμενο μίας δραστηριότητας δεν χρησιμοποιείται με αυτήν την κλάση.

6.Collection: Αποτελεί υποκλάση του Object και περιγράφει μία συλλογή από instances αντικειμένων ή links.

7.Ordered Collection: Υποκλάση του Collection όπου τα αντικείμενα μίας συλλογής είναι πάντα ταξινομημένα.

8.Collection Page: Διευρύνει την κλάση Collection προσθέτοντας περισσότερα properties με τα οποία μπορεί να γίνει η σειριοποίηση των αντικειμένων της συλλογής.

9.Ordered Collection Page: Παρόμοια κλάση με την Ordered Collection η οποία ταξινομεί τα αντικείμενα ενός Collection Page.

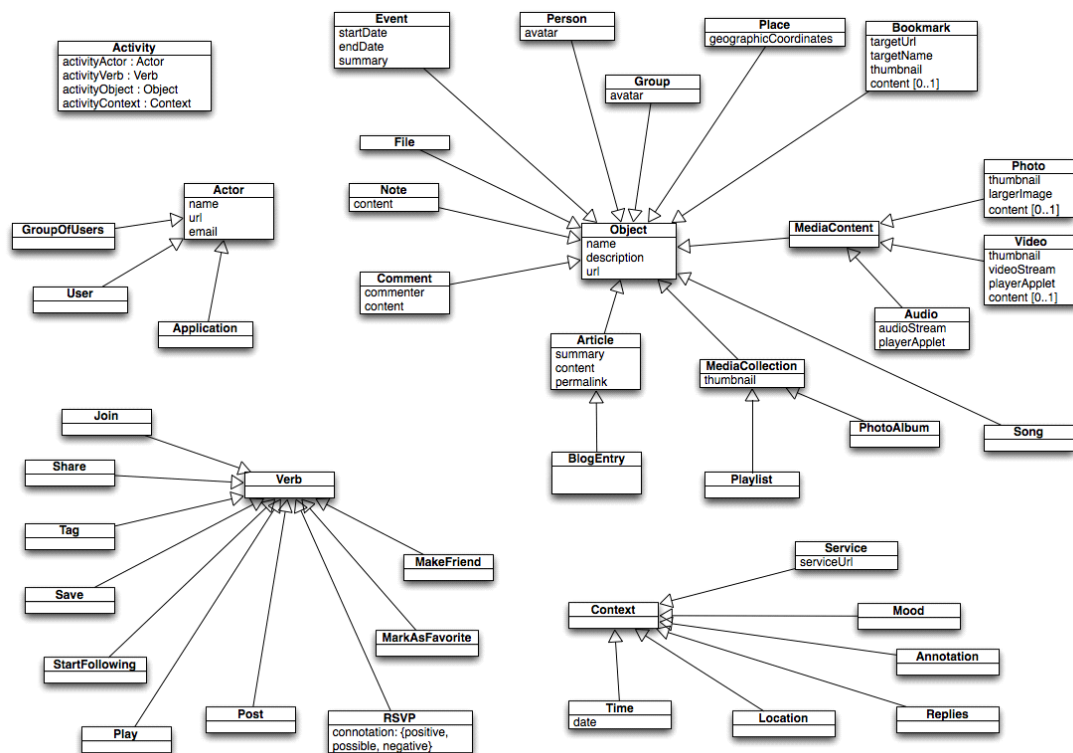
Οι διευρυμένες κλάσεις της Activity Streams περιλαμβάνουν υποκλάσεις κάποιων από τις βασικές κλάσεις και διαχωρίζονται σε τρεις ομάδες.

1.Activity Types: Περιλαμβάνουν όλες τις υποκλάσεις της Activity και περιγράφουν πιο συγκεκριμένους τύπους δραστηριοτήτων όπως Accept,Announce,Delete,Ignore κτλ.

2.Actor Types: Υποκλάσεις της Actor και περιγράφουν τους διαφορετικούς τύπους ατόμων ή ομάδων ατόμων τα οποία πραγματοποιούν κάποια δραστηριότητα και περιλαμβάνουν τους τύπους Application,Group,Organization,Person,Service.

3.Object Types: Υποκλάσεις της Object οι οποίες περιγράφουν τους διαφορετικούς τύπους αντικειμένων που ορίζει η οντολογία όπως Article,Audio,Video,image,Place κτλ.

Στο σχήμα 4.5 φαίνεται ένα βασικό γράφημα της οντολογίας.



Σχήμα 4.5:Γράφημα της οντολογίας activity streams 2.0

4.1.3 Χαμηλό Επίπεδο

Για το χαμηλό επίπεδο μελέτησα οντολογίες οι οποίες ορίζουν λεξιλόγια τα οποία μπορούν να δώσουν περισσότερες λεπτομέρειες στις γενικές κλάσεις των προηγούμενων οντολογιών. Δύο οντολογίες που προσφέρουν εκτενή λεξιλόγια για έννοιες όπως Activity, Person και Object είναι η DBpedia και η Schema.org. Από τις δύο, η Schema.org προσφέρει μεγαλύτερο λεξιλόγιο για την έννοια της δραστηριότητας ενώ θεώρησα ότι η DBpedia έχει αχρειαστές λεπτομέρειες για τον σκοπό της διπλωματικής, για παράδειγμα διαθέτει την κλάση αθλητής και υποκλάσεις που περιγράφουν όλα τα διαφορετικά είδη αθλητών. Συνεπώς κατέληξα στην οντολογία Schema.org.

Η Schema.org παρέχει μία συλλογή από λεξιλόγια τα οποία οι σχεδιαστές ιστοσελίδων μπορούν να χρησιμοποιήσουν για να σημαδέψουν τις σελίδες τους με τρόπο που αυτές θα μπορούν να κατανοηθούν από μηχανές αναζήτησης. Στην ουσία τη χρησιμοποίησα σαν επέκταση της Activity Streams καθώς περιγράφει μία πληθώρα τύπων αντικειμένων, κάθε ένα εκ των οποίων διαθέτει τα δικά του properties τα οποία περιγράφουν το αντικείμενο. Ο πιο γενικός τύπος αντικείμενου είναι η κλάση Thing η οποία περιγράφεται από τέσσερα Properties: name, description, URL και image. Τα υπόλοιπα αντικείμενα είναι υποκλάσεις του Thing και πέρα από τα συγκεκριμένα Properties που τα περιγράφουν, κληρονομούν και τα Properties του Thing. Στη συνέχεια περιγράφονται οι κλάσεις της οντολογίας.

1.Action: Μία πράξη η οποία πραγματοποιείται από ένα υποκείμενο πάνω σε ένα αντικείμενο. Το υποκείμενο μπορεί να είναι ένα άτομο ή ένας οργανισμός και η ολοκλήρωση της πράξης οδηγεί σε κάποιο αποτέλεσμα. Αποτελείται από ένα πλήθος υποκλάσεων οι περισσότερες εκ των οποίων διαθέτουν τις δικές τους υποκλάσεις για να

περιγράφουν πλήρων τον εκάστοτε τύπο πράξης.Οι υποκλάσεις είναι οι πράξεις: Achieve, Assess, Consume, Control, Create, Find, interact, Move, Organize, Play, Search, Trade, Transfer και Update.

2.Creative work: Ο πιο γενικός τύπος κάποιας δημιουργικής εργασίας η οποία μπορεί να είναι βιβλίο, ταινία, πρόγραμμα software κτλ. Συγκεκριμένα κάποιες από τις υποκλάσεις της Creative work είναι οι: Article, Blog, Book, Clip, Code, Comment, Conversation,, Digital Document, Episode, game, Map, Media object,, Movie, Music composition, Painting κτλ. Υπάρχουν αρκετές ακόμα υποκλάσεις γεγονός που υποδεικνύει πόσο λεπτομερές είναι η Schema.org.

3.Event: Γεγονός το οποίο συμβαίνει σε κάποιο συγκεκριμένο χρόνο και χώρο, για παράδειγμα μία συναυλία, μία διάλεξη ή ένα φεστιβάλ. Ενδεικτικά κάποιες από τις υποκλάσεις της Event είναι οι Business event, Children's event, Comedy event, Dance event, Food event, Music event κτλ.

4.Intangible: Γενική κλάση που αντιπροσωπεύει ένα μεγάλο αριθμό από άυλα αντικείμενα όπως μία ποσότητα ή μία βαθμολογία. Κάποιες από τις υποκλάσεις της είναι οι Brand, Digital document permission, Enumeration, Game server, Invoice, Language, Job posting κτλ.

5.Organization: Ένας οποιοσδήποτε οργανισμός όπως για παράδειγμα ένα σχολείο ή μία επιχείρηση. Έχει υποκλάσεις τις Airline, Corporation, Educational organization, Government organization, Local business, Medical organization, NGO(non-government organization), performing group και Sports organization.

6.Person: Ένα οποιοδήποτε πρόσωπο πραγματικό ή φανταστικό. Δεν διαθέτει υποκλάσεις.

7.Place: Περιλαμβάνει οντότητες που βρίσκονται σε ένα σταθερό φυσικό χώρο. Περιλαμβάνει τοποθεσίες όπως Accommodation, Administrative area, Civic structure, Landform, Landmarks or historical building, Local business, Residence και Tourist attraction.

8. Product: Ένα οποιοδήποτε προσφερόμενο προϊόν ή υπηρεσία. Για παράδειγμα ένα ζευγάρι παπούτσια, εισιτήρια συναυλίας, τραπεζικές υπηρεσίες κτλ. Έχει τέσσερις υποκλάσεις τις Individual product, Product model, Some products και Vehicle.

4.1.4 Μοντέλο Activity Tracker

Όπως προαναφέρθηκε στην εισαγωγή του κεφαλαίου η εφαρμογή Activity Tracker καταγράφει τις δραστηριότητες ενός χρήστη. Κάθε δραστηριότητα έγκειται σε μία συγκεκριμένη κλάση (π.χ. Running, Status update κτλ.) και συνδέεται μέσω properties με τον χρήστη, τον σκοπό του, τα εργαλεία που χρησιμοποιεί κτλ. Αναπτύχθηκε μάλιστα έχοντας ως θεωρητικό υπόβαθρο τη θεωρία δραστηριοτήτων γι' αυτό και ταιριάζει με τις κλάσεις της οντολογίας που ανέπτυξα. Ωστόσο τα δεδομένα του Activity Tracker ήταν αποθηκευμένα σε μορφή SQL άρα έπρεπε για τη σύνδεσή τους με την οντολογία μου να τα μετατρέψω σε μορφή RDFs. Χρησιμοποίησα γι' αυτόν το σκοπό το πρόγραμμα DB Transformer το οποίο πραγματοποιεί mapping από μία βάση δεδομένων σε μορφή RDFs και OWL. Οι κλάσεις της οντολογίας που δημιουργήθηκε εν τέλει περιγράφονται παρακάτω:

1.Activitytracker_activity: Αναπαριστά τη δραστηριότητα την οποία καταγράφει ο χρήστης στο σύστημα. Περιλαμβάνει datatype properties τα οποία δίνουν σε κάθε instance το όνομα της δραστηριότητας, την περιγραφή της, την κατηγορία της και ένα identifier.

2. Activitytracker_friend: Χρησιμοποιείται σε περίπτωση συλλογικών δραστηριοτήτων στις οποίες ο χρήστης αλληλεπιδρά με κάποιο άλλο άτομο ή οργανισμό στο πλαίσιο

ολοκλήρωσης της δραστηριότητάς του. Τα datatype properties της κλάσης περιλαμβάνουν το όνομα του friend, το όνομα του χρήστη ο οποίος αλληλεπιδρά με τον friend και ένα identifier για κάθε instance της κλάσης friend.

3. Activitytracker_object: Αναφέρεται στο αντικείμενο το οποίο χρησιμοποιεί ο χρήστης για την ολοκλήρωση ή την καταγραφή της δραστηριότητάς του και μπορεί να είναι είτε μία εφαρμογή (π.χ. Youtube, Twitter) είτε κάποιο φυσικό αντικείμενο (smartphone, smart watch). Περιλαμβάνει datatype properties τα οποία περιγράφουν το όνομα του object, το identifier του και το identifier του χρήστη που χρησιμοποιεί το αντικείμενο για ένα συγκεκριμένο instance.

4. Activitytracker_performs: Κλάση η οποία παρέχει ένα πλήθος από datatype properties για τον καλύτερο ορισμό μίας δραστηριότητας. Δίνει για κάθε instance την τοποθεσία της δραστηριότητας, τους χρόνους έναρξης και λήξης, το σκοπό και αν αυτός επιτεύχθηκε κτλ.

5. Activitytracker_performs_using: Χρησιμοποιείται για να συνδέσει την κλάση performs με την object και στην ουσία δείχνει πιο αντικείμενο χρησιμοποίησε ο χρήστης σε κάποιο instance δραστηριότητας.

6. Activitytracker_performsproviderinfo: Χρησιμοποιείται για να περιγράψει την υπηρεσία που χρησιμοποιεί ο χρήστης στην περίπτωση online δραστηριότητας και δηλώνει το όνομα του παρόχου της υπηρεσίας, ένα URL και κάποια identifiers για την υπηρεσία και τον χρήστη που τη χρησιμοποιεί.

7. Activitytracker_places: Περιγράφει την τοποθεσία όπου έλαβε μέρος η δραστηριότητα και την περιγράφει με datatype properties τα οποία δηλώνουν το γεωγραφικό μήκος και πλάτος, τη διεύθυνση και το όνομα της τοποθεσίας.

8. Activitytracker_routine: Περιγράφει δραστηριότητες οι οποίες επαναλαμβάνονται σε συγκεκριμένες ώρες κάθε μέρα ή κάποιες συγκεκριμένες ημέρες της εβδομάδας. Για παράδειγμα η δραστηριότητα συγκοινωνία για κάποιον χρήστη μπορεί να περιγραφεί μέσω αυτής της κλάσης με datatype properties τα οποία δηλώνουν τις ώρες έναρξης και λήξης της ρουτίνας, το identifier της δραστηριότητας την οποία αναπαριστά και τις ημέρες στις οποίες απαναλαμβάνεται η δραστηριότητα (π.χ. καθημερινές ή Σαββατοκύριακο)

9. Activitytracker_user: Περιγράφει το χρήστη που χρησιμοποιεί την εφαρμογή και περιλαμβάνει αρκετές λεπτομέρειες όπως ονοματεπώνυμο, username, e-mail, αν ο χρήστης είναι ενεργός αυτή τη στιγμή κτλ.

4.2 Σύνδεση οντολογιών

Σε αυτό το υποκεφάλαιο θα γίνει περιγραφή της σύνδεσης των δεδομένων των οντολογιών που περιγράφηκαν στο προηγούμενο κομμάτι του κεφαλαίου. Η σύνδεση έγινε σε τρία επίπεδα: Υψηλό-μεσαίο και μεσαίο-χαμηλό και τέλος σύνδεση των πραγματικών δεδομένων του Activity Tracker στην τελική οντολογία.

4.2.1 Σύνδεση υψηλού-μεσαίου επιπέδου

Αρχικά θα αναλυθεί η σύνδεση της οντολογίας AMOn με τις CoDAMoS και Activity Streams 2.0 ξεχωριστά.

4.2.1.1 Σύνδεση AMOn-CoDAMoS

Οι κλάσεις της CoDAMoS, οι οποίες είναι σχετικές με την AMOn είναι οι ακολουθίες: Activity, Environment, EnvironmentalCondition (. Humidity, Lighting, Noise, Pressure,

Temperature), Hardware (IODevice, Resource), Location (AbsoluteLocation, RelativeLocation), Modality, Mood, Platform, Profile (PreferenceProfile), Role, Service, ServiceGrounding, ServiceModel, ServiceProfile, Software (Middleware, OperatingSystem, RenderingEngine, VirtualMachine), Task, Time, και User. Επιπλέον υπάρχουν κάποιες συνδέσεις ανάμεσα στις δύο οντολογίες οι οποίες απαριθμούνται παρακάτω.

1. Activity (CoDAMoS) is SubClassOf Activity (AMOn)
2. Task (CoDAMoS) is SubClassOf Action (AMOn)
3. Hardware (CoDAMoS) is SubClassOf PhysicalTools (AMOn)
4. Software (CoDAMoS) is SubClassOf PhysicalTools (AMOn)
5. Platform (CoDAMoS) is SubClassOf PhysicalTools (AMOn)
6. Service (CoDAMoS) is SubClassOf Tool (AMOn)
7. Mood (CoDAMoS) is SubClassOf Mood(affective state-mental state) (AMOn)
8. Environment (CoDAMoS) is SubClassOf Rules&Norms(Conventions) (AMOn)
9. User (CoDAMoS) is SubClassOf Subject (AMOn)
10. Role (CoDAMoS) is InfluencedBy DivisionOfLabour (AMOn)
11. Time (CoDAMoS) is SubClassOf Descriptor (AMOn)
12. Location (CoDAMoS) is SubClassOf Location (AMOn)

4.2.1.2 Σύνδεση AMOn-Activity Streams 2.0

Οι κλάσεις της Activity Streams 2.0, οι οποίες είναι σχετικές με την AMOn είναι οι Link και Object(Actor, Activity, Group, Organization) ενώ οι συνδέσεις μεταξύ τους είναι οι εξής:

1. Person(Activity Streams) is SubClassOf Individual(AMOn)
2. Activity (Activity Streams) is SubClassOf Action (AMOn)
3. Link (Activity Streams) is SubClassOf MentalTools (AMOn)
4. Object (Activity Streams) is SubClassOf Object (AMOn)
5. Group (Activity Streams) is SubClassOf Group (AMOn)
6. Organization (Activity Streams) is SubClassOf Organisation (AMOn)

4.2.2 Σύνδεση μεσαίου-χαμηλού επιπέδου

Σε αυτό το υποκεφάλαιο περιγράφεται η σύνδεση της οντολογίας Schema.org με τις οντολογίες του μεσαίου επιπέδου.

4.2.2.1 Σύνδεση Activity Streams 2.0-Schema.org

Οι σχετικές κλάσεις της Schema.org με την Activity Streams 2.0 είναι οι Action, Creative work, Person, Organization και Product και συνδέονται μεταξύ τους ως εξής.

1. Action(Schema.org) SubClassOf Activity(Activity Streams)
2. Creative Work(Schema.org) SubClassOf Object(Activity Streams)
3. Person(Schema.org) SubClassOf Person(Activity Streams)
4. Image(Schema.org) SubClassOf Image(Activity Streams)
5. Organization(Schema.org) SubClassOf Organization(Activity Streams)
6. Product(Schema.org) SubClassOf Object(Activity Streams)

4.2.2.2 Σύνδεση CoDAMoS-Schema.org

Οι σχετικές κλάσεις ανάμεσα στην CoDAMoS και την Schema.org είναι οι Time, Event και Place. Παρακάτω φαίνονται όλες οι πιθανές συνδέσεις ανάμεσα στις δύο οντολογίες.

1. Date(Schema.org) SubClassOf Time(CoDAMoS)
2. Date Time(Schema.org) SubClassOf Time(CoDAMoS)
3. Time(Schema.org) SubClassOf Time(CoDAMoS)
4. Event(Schema.org) SubClassOf Activity(CoDAMoS)
5. Place(Schema.org) SubClassOf Location(CoDAMoS)

4.2.3 Σύνδεση Δεδομένων του Activity Tracker

Τέλος οι σχετικές κλάσεις ανάμεσα στα δεδομένα του Activity Tracker και της τελικής οντολογίας καθώς και οι συνδέσεις μεταξύ τους φαίνονται παρακάτω:

1. Activitytracker_activity SubClassOf Activity(Activity Streams)
2. Activitytracker_object SubClassOf Object(Activity Streams)
3. Activitytracker_places SubClassOf Location(CoDAMoS)
4. Activitytracker_user SubClassOf Person(Activity Streams)

4.3 SPARQL Queries

Στο κεφάλαιο αυτό θα γίνει καταγραφή όλων των SPARQL Queries που έκανα πάνω στην οντολογία μου. Χρησιμοποίησα την εφαρμογή Dydra.com η οποία επιτρέπει σε έναν χρήστη να κάνει SPARQL Queries πάνω σε σημασιολογικές βάσεις δεδομένων που μπορεί να ανεβάσει ο ίδιος στην εφαρμογή. Επίσης σε αυτό το σημείο πρέπει να αναφέρω ότι για τα Queries που θα ακολουθήσουν δεν χρησιμοποίησα την τελική μου οντολογία για τους παρακάτω λόγους. Η οντολογία στην οποία κατέληξα ήταν πολύ μεγάλη σε μέγεθος. Περιελάμβανε περισσότερα από 170000 statements με αποτέλεσμα το πρόγραμμα Protege να μην μπορεί να κάνει reasoning σε αυτήν και γενικότερα να μην μπορεί να την αντέξει υπολογιστικά. Εξάλλου για τα SPARQL Queries που ήθελα να κάνω δεν χρειαζόμουν το σύνολο της οντολογίας. Αρκέστηκα λοιπόν στο να συνδέσω την AMOn και την Activity Streams οι οποίες περιέχουν τα απολύτως απαραίτητα όσον αφορά τα Classes και Properties που χρειαζόμουν. Επιπλέον αντί να συνδέσω σε αυτές τα δεδομένα της βάσης από τον Activity Tracker δημιούργησα μόνος μου δύο άτομα, τον εαυτό μου και τον υπεύθυνο της διπλωματικής μου (τα δεδομένα του οποίου μετέφερα επακριβώς από τη βάση δεδομένων), και δημιούργησα κάποια Instances δραστηριοτήτων και λεπτομερειών γύρω από αυτές. Αυτό έγινε και πάλι για να κάνω την τελική οντολογία λιγότερο βαριά για το Protégé και να μπορώ να εργάζομαι πάνω σε αυτήν χωρίς να δυσχεραίνεται η λειτουργία του προγράμματος. Εξάλλου ο σκοπός των queries ήταν να δείξω ότι η γλώσσα ερωτημάτων SPARQL μπορεί να χρησιμοποιηθεί επαρκώς για να κάνει ερωτήσεις πάνω σε σημασιολογικά δεδομένα και να μου δώσει τα αναμενόμενα αποτελέσματα. Ακολουθούν οι ερωτήσεις χωρισμένες σε υποκεφάλαια και με κάποια επεξηγηματικά λόγια για την καθεμία από αυτές.

4.3.1 Λίστα δραστηριοτήτων/Χρήστη

Η πρώτη ερώτηση που έκανα είναι αρκετά απλή και την συμπεριέλαβα κυρίως για να δείξω τη φιλοσοφία που ακολούθησα στις υπόλοιπες ερωτήσεις και τη μορφή των αντικειμένων και των properties. Η εν λόγω ερώτηση ζητάει να επιστραφούν οι δραστηριότητες που είναι καταγεγραμμένες για κάθε χρήστη και τις επιστρέφει σε μορφή πίνακα όπου η στήλη ?c αντιπροσωπεύει το όνομα του χρήστη και η ?b το όνομα της δραστηριότητας. Εδώ να σημειώσω ότι για τα predicates των triples χρησιμοποίησα τα πλήρη URIs. Τέλος, να αναφέρω ότι το πρώτο triple που χρησιμοποιεί το predicate <http://www.w3.org/ns/activitystreams#name>, το οποίο είναι ενσωματωμένο datatype property της activity streams επιτρέπει την αντικατάσταση ενός URI με το string που έχω δηλώσει ως όνομα του instance. Από την άλλη οι δραστηριότητες εμφανίζονται με το

πλήρες URI τους για να φανεί η διαφορά ανάμεσα στις δύο τεχνικές και πόσο χρήσιμη είναι η πρώτη για την εξοικονόμηση χώρου στον πίνακα αποτελεσμάτων.

```
SELECT ?c ?b
```

```
WHERE {
```

```
  ?a <http://www.w3.org/ns/activitystreams#name> ?c.
```

```
  ?a <http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasActivity> ?b
```

```
}
```

?c	?b
Christos	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#statusupdate
Christos	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#socialmedia
Christos	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#sleeping
Christos	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#eating
Christos	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#commuting
Christos	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#videowatching
Christos	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#running
Iosif	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#working
Iosif	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#videoupload
Iosif	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#shareretweet
Iosif	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#statusupdate
Iosif	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#socialmedia
Iosif	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#sleeping
Iosif	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#eating
Iosif	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#commuting
Iosif	http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#running

4.3.2 Κοινές δραστηριότητες των δύο χρηστών

Η δεύτερη ερώτηση εμφανίζει όλες τις κοινές δραστηριότητες των δύο χρηστών, καθώς είναι μία πληροφορία που έκρινα ότι θα ήθελε να μάθει ένας χρήστης όταν συνδέεται με κάποιον άλλο μέσω της εφαρμογής. Με αυτόν τον τρόπο μπορεί να μάθει αν έχει κοινά ενδιαφέροντα με μία επαφή του και αν ενδιαφέρεται εν τέλει να συνδεθεί μαζί του. Χρησιμοποίησα την εντολή `Distinct ?b` για να μην εμφανιστούν διπλά αποτελέσματα, κάτι που θα γινόταν δεδομένου ότι χρησιμοποίησα δύο ίδια triples και θα γινόταν διπλή προσπέλαση των δεδομένων. Τα triples είναι της μορφής `Person <hasActivity> Activity`. Τέλος χρησιμοποίησα φίλτρο αποτελεσμάτων στο οποίο στην ουσία δηλώνω ότι στα δύο triples θέλω ο χρήστης του πρώτου να είναι διαφορετικός από τον χρήστη του δεύτερου ενώ οι δραστηριότητες να είναι οι ίδιες. Έτσι πέτυχα να μου επιστραφούν όλες οι κοινές δραστηριότητες.

```
SELECT DISTINCT ?b
```

```
WHERE {
```

```
  ?a <http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasActivity> ?b.
```

?c <<http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasActivity>> ?d.

FILTER (?b = ?d && ?a != ?c)

}

?b

<http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#running>

<http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#commuting>

<http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#eating>

<http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#sleeping>

<http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#socialmedia>

<http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#statusupdate>

4.3.3 Συνολικές καταγραφές συγκεκριμένης δραστηριότητας/χρήστη

Η τρίτη ερώτηση που έκανα ακολουθεί την εξής λογική. Από τη στιγμή που ένας χρήστης έχει συνδεθεί με κάποιον άλλο και έχει διαπιστώσει πως έχει κοινές δραστηριότητες με αυτόν μπορεί να θέλει να συγκρίνει πληροφορίες για κάποια συγκεκριμένη δραστηριότητα. Για όλες τις ερωτήσεις που ακολουθούν αυτήν τη φιλοσοφία χρησιμοποίησα τη δραστηριότητα του τρεξίματος στο φίλτρο καθώς το Dydra δεν παρέχει κάποιο τρόπο ώστε να δηλώσω απλά τη μεταβλητή της δραστηριότητας και να του δείξω με κάποιον άλλο τρόπο ποια δραστηριότητα επιθυμώ να ελέγξω. Στην ενότητα του Select ζητάω αρχικά δύο μεταβλητές που συμβολίζουν το όνομα του χρήστη και τις δραστηριότητες και στη συνέχεια χρησιμοποίησα τη συνάρτηση Count η οποία μετράει πόσες καταγραφές ενός συγκεκριμένου instance υπάρχουν στη σημασιολογική βάση. Τέλος διαχώρησα τα ποτελέσματα ανά χρήστη με την εντολή GROUP BY ?a, όπου ?a η μεταβλητή που αντιπροσωπεύει το χρήστη και ζήτησα να εμφανίζονται πρώτα τα δεδομένα του χρήστη ο οποίος έχει καταγεγραμμένες περισσότερες δραστηριότητες αυτού του τύπου με την εντολή ORDER BY DESC(?amount).

```
SELECT ?d ?c (COUNT(?b) as ?amount)
```

```
WHERE {
```

```
?b <http://www.w3.org/ns/activitystreams#name> ?c.
```

```
?a <http://www.w3.org/ns/activitystreams#name> ?d.
```

```
?a <http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasPerformed> ?b.
```

```
FILTER(?c = "running" )
```

```
}
```

```
GROUP BY ?a
```

```
ORDER BY DESC(?amount)
```

?d	?c	?amount
Iosif	running	5
Christos	running	3

4.3.4 Συνολικές καταγραφές όλων των δραστηριοτήτων/χρήστη

Η τέταρτη ερώτηση επεκτείνει τα αποτελέσματα της προηγούμενης καθώς ζητάει το σύνολο καταγραφών από όλες τις δραστηριότητες των χρηστών. Η ερώτηση αυτή είναι πανομοιότυπη με την προηγούμενη με τη διαφορά ότι δεν χρησιμοποιήσα φίλτρο που ζητάει κάποια συγκεκριμένη δραστηριότητα.

```
SELECT ?c ?d (COUNT(?b) as ?amount)
```

```
WHERE {
```

```
  ?b <http://www.w3.org/ns/activitystreams#name> ?c.
```

```
  ?a <http://www.w3.org/ns/activitystreams#name> ?d.
```

```
  ?a <http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasPerformed> ?b.
```

```
}
```

```
GROUP BY ?c ?a
```

?c	?d	?amount
statusupdate	Christos	2
socialmedia	Christos	3
videowatching	Christos	7
running	Christos	3
statusupdate	Iosif	8
running	Iosif	5
videoupload	Iosif	3
shareretweet	Iosif	3
socialmedia	Iosif	1

4.3.5 Ποσοστό μίας δραστηριότητας επί των συνολικών καταγραφών

Η πέμπτη ερώτηση ζητάει το ποσοστό μίας συγκεκριμένης δραστηριότητας επί των συνολικών καταγραφών για όλες τις δραστηριότητες ανά χρήστη. Η λογική πίσω από αυτό το query είναι ότι για τον χρήστη μπορεί να μην αρκεί το γεγονός ότι έχει μία κοινή δραστηριότητα με κάποιον άλλον ενώ ο αριθμός των καταγραφών δεν είναι πάντα αξιόπιστη πληροφορία. Αυτό που έχει μεγαλύτερη σημασία είναι η συχνότητα της δραστηριότητας. Για την εύρεσή της υπολόγισα όπως στην τρίτη ερώτηση τον αριθμό των καταγραφών της δραστηριότητας και επιπλέον όλες τις καταγραφές ανά χρήστη χρησιμοποιώντας και πάλι την εντολή Count. Κατάφερα να διαχωρήσω τα δύο Counts χρησιμοποιώντας φίλτρο για τη δραστηριότητα στα triples που μετράνε τις καταγραφές αυτής και αφήνοντας τα άλλα δύο triples να προβούν σε υπολογισμούς χωρίς φίλτρο ώστε να μετρήσουν το σύνολο των καταχωρημένων δραστηριοτήτων του κάθε χρήστη. Επιπλέον, στο φίλτρο έδωσα συγκεκριμένα instances για τα subjects των triples δηλώνοντας ότι το ?a αναφέρεται στον χρήστη Iosif και το ?d στον χρήστη Χρήστο καθώς υπήρχαν αρκετά δεδομένα και συναρτήσεις για να μπορεί να γίνει ο διαχωρισμός με την εντολή GROUP BY. Τέλος, υπολόγισα τα ποσοστά της δραστηριότητας για τους δύο χρήστες χρησιμοποιώντας τις εντολές ((?Iosif/?Iosiftotal) AS ?IosifPerc) και ((?Christos/?Christostotal) AS ?ChristosPerc) οι οποίες διαιρούν δύο τιμές και αναθέτουν το αποτέλεσμα σε μία τρίτη.

```
SELECT ?activity (COUNT(distinct(?b)) as ?Iosif) (COUNT(distinct(?e)) as ?Christos)
(COUNT(distinct(?f)) as ?Iosiftotal) (COUNT(distinct(?g)) as ?Christostotal) ((?Iosif/?Iosiftotal)
AS ?IosifPerc) ((?Christos/?Christostotal) AS ?ChristosPerc)
```

```
WHERE {
```

```
?b <http://www.w3.org/ns/activitystreams#name> ?activity.
```

```
?e <http://www.w3.org/ns/activitystreams#name> ?activity.
```

```
?a <http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasPerformed> ?b.
```

```
?d <http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasPerformed> ?e.
```

```
?a <http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasPerformed> ?f.
```

```
?d <http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasPerformed> ?g.
```

```
FILTER(?activity = "running" &&
```

```
?a=<http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#Iosif> &&
```

```
?d=<http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#Christos>)
```

```
}
```

?activity	?Iosif	?Christos	?Iosiftotal	?Christostotal	?IosifPerc	?ChristosPerc
running	5	3	20	15	0.25	0.2

4.3.6 Μέσος όρος διάρκειας μίας δραστηριότητας

Αυτή η ερώτηση ζητάει πληροφορίες σχετικά με τον μέσο χρόνο διάρκειας μίας δραστηριότητας για τους δύο χρήστες. Τα πρώτα τρία triples δεν κάνουν κάτι διαφορετικό από τα προηγούμενα queries, καθώς αναθέτουν απλά τις δύο μεταβλητές που χρειάζομαι στα ονόματα των instances που χρησιμοποίησα και στη συνέχεια ζητάει όπως πριν τις συνολικές καταχωρήσεις μίας δραστηριότητας. Με το τέταρτο triple ωστόσο δηλώνεται ότι η κάθε καταχώρηση της δραστηριότητας έχει διάρκεια ?f. τέλος χρησιμοποιώντας στην ενότητα του SELECT τη συνάρτηση AVG, η οποία υπολογίζει το μέσο όρο ενός συνόλου το οποίο στη συνέχεια αναθέτει σε μία μεταβλητή έτυχα να μου επιστραφεί το επιθυμητό αποτέλεσμα το οποίο εμφάνισα όπως και προηγουμένως σε φθίνουσα σειρά.

```
SELECT ?Name ?Activity (AVG(?f) AS ?Meantime)
```

```
WHERE {
```

```
?b <http://www.w3.org/ns/activitystreams#name> ?Activity.
```

```
?a <http://www.w3.org/ns/activitystreams#name> ?Name.
```

```
?a <http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasPerformed> ?b.
```

```
?b <http://www.w3.org/ns/activitystreams#duration> ?f.
```

```
FILTER(?Activity = "running")
```

}

GROUP BY ?a

ORDER BY DESC(?Meantime)

?Name	?Activity	?Meantime
Iosif	running	41.2
Christos	running	22.333334

4.3.7 Compare activities

Η τελευταία ερώτηση που πραγματοποίησα πάνω στην οντολογία μοιάζει αρκετά με την τρίτη. Στην ουσία μετράει τις συνολικές καταχωρήσεις μίας δραστηριότητας για κάθε χρήστη αλλά παράλληλα συγκρίνει τις δύο τιμές με μία λογική συνάρτηση και εμφανίζει ένα μήνυμα ανάλογα με το αποτέλεσμα. Ο σκελετός της ερώτησης, το κομμάτι δηλαδή που περιέχεται ανάμεσα στις δύο αγκύλες μετά το WHERE δεν παρουσιάζει κάτι διαφορετικό σε σχέση με τις παραπάνω ερωτήσεις ενώ το φίλτρο είναι ίδιο με αυτό της πέμπτης ερώτησης. Στην ενότητα του SELECT ωστόσο, χρησιμοποίησα τη λογική συνάρτηση if, η οποία εξετάζει μία συνθήκη, στην προκειμένη περίπτωση αν οι καταχωρημένες δραστηριότητες του Iosif είναι περισσότερες από αυτές του Χρήστου και ακολουθείται από δύο μηνύματα. Αν η συνθήκη είναι αληθής εμφανίζει το πρώτο κατά σειρά μήνυμα ενώ αν είναι ψευδής το δεύτερο. Σε αυτό το query ο Iosif έχει περισσότερες δραστηριότητες τρεξίματος καταχωρημένες οπότε εμφανίζεται το μήνυμα Iosif has more activities of this type than Christos. Σημειωτέον, ότι η έξοδος της ερώτησης δεν κάνει αναφορά ότι ο χρήστης ζήτησε πληροφορίες για τη δραστηριότητα “τρέξιμο”. Έκανα αυτήν την επιλογή έτσι ώστε να μπορώ με μία απλή αλλαγή της δραστηριότητας που φιλτράρω να μπορώ απευθείας να έχω κάποιο αποτέλεσμα για μία άλλη δραστηριότητα χωρίς η έξοδος να χάνει το νήμά της.

```
SELECT (if (COUNT(distinct(?b))>COUNT(distinct(?e)),"Iosif has more activities of this type than Christos", "Christos has more activities of this type than Iosif") as ?answer)
(COUNT(distinct(?b)) as ?Iosiftotal) (COUNT(distinct(?e)) as ?Christostotal)
```

```
WHERE {
```

```
?b <http://www.w3.org/ns/activitystreams#name> ?c.
```

```
?e <http://www.w3.org/ns/activitystreams#name> ?c.
```

```
?a <http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasPerformed> ?b.
```

```
?d <http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#hasPerformed> ?e.
```

```
FILTER(?c = "running" &&
```

```
?a=<http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#Iosif> &&
```

```
?d=<http://www.semanticweb.org/hp/ontologies/2016/8/untitled-ontology-424#Christos>)
```

```
}
```

?answer	?Iosiftotal	?Christostotal
---------	-------------	----------------

losif has more activities of this type than Christos	5	3
--	---	---

5 Ανάλυση και Σχεδίαση Συστήματος

Στα υποκεφάλαια που ακολουθούν θα γίνει ανάλυση των λειτουργιών της εφαρμογής που ανέπτυξα χρησιμοποιώντας το Django framework. Επιπλέον θα πραγματοποιηθεί περαιτέρω εστίαση στα δομικά τμήματα της εφαρμογής και περιγραφή της σχεδίασης του συστήματος σε επίπεδο αρχιτεκτονικής ανάλυσης βασιζόμενο στο επιλεγμένο πλαίσιο.

5.1 Απαιτήσεις συστήματος

Στο υποκεφάλαιο αυτό γίνεται αναφορά στις απαιτήσεις που αναμένεται να ικανοποιεί η εφαρμογή. Πρόκειται για γενικές απαιτήσεις που ένας χρήστης θα θέλει από το σύστημα ενώ η αναφορά στις πιο συγκεκριμένες λειτουργίες που αυτό επιτελεί θα γίνει σε επόμενο κεφάλαιο.

Απαιτήσεις συστήματος:

- Ανάπτυξη εφαρμογής διαδικτύου η οποία να είναι συνδεδεμένη με τη βάση δεδομένων του Activity tracker.
- Παρουσίαση γενικής πληροφορίας σχετικά με τους εγγεγραμμένους χρήστες στο Activity tracker και των δραστηριοτήτων που είναι καταχωρημένες.
- Σύνδεση της εφαρμογής με τη σημασιολογική βάση δεδομένων που αναπτύχθηκε.
- Παρουσίαση των πληροφοριών που αποκομίστηκαν μέσω των ερωτήσεων SPARQL.
- Ύπαρξη σύνδεσμων ανακατεύθυνσης στα endpoint URLs των SPARQL ερωτήσεων στην εφαρμογή Dydra.
- Εγγραφή και σύνδεση χρήστη στην εφαρμογή.

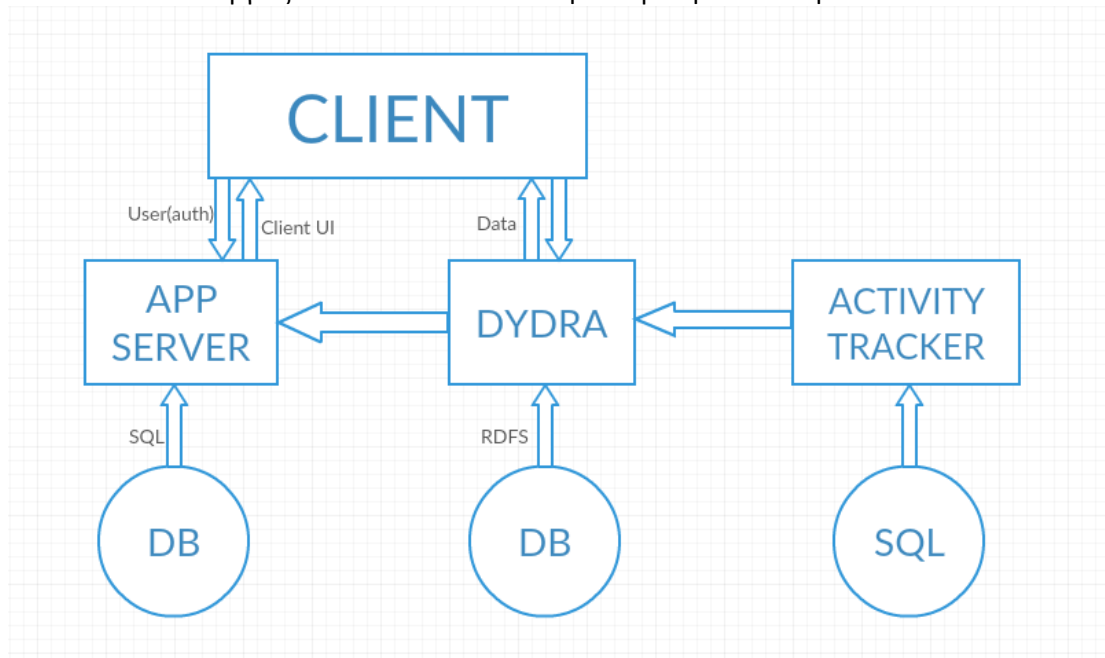
5.2 Αρχιτεκτονική συστήματος

Η εφαρμογή που αναπτύχθηκε ακολουθεί το μοντέλο αρχιτεκτονικής πελάτη – εξυπηρετητή (client-server). Σύμφωνα με το συγκεκριμένο μοντέλο υπάρχουν δύο πλευρές στο σύστημα οι οποίες επικοινωνούν μεταξύ τους. Η client πλευρά ζητάει δεδομένα από τη server πλευρά η οποία της επιστρέφει τα δεδομένα που ζήτησε. Στην παρούσα εφαρμογή η server πλευρά είναι υπεύθυνη για τη διαχείριση της βάσης δεδομένων, τη σύνδεση με τα δεδομένα της διεπαφής του Dydra και την παρουσίαση των πληροφοριών που ζητάει ο χρήστης σε κατανοητή και αναγνώσιμη μορφή. Σε αυτό το κεφάλαιο θα γίνει περιγραφή των υποσυστημάτων της εφαρμογής και θα παρουσιαστεί ένα διάγραμμα που δείχνει την αλληλεπίδραση των υποσυστημάτων.

- **Βάση δεδομένων(Database):** Η εφαρμογή είναι συνδεδεμένη με δύο βάσεις δεδομένων. Η μία είναι σε μορφή SQL και δημιουργήθηκε με την μετατροπή των δεδομένων της τελικής οντολογίας σε SQL μορφή με την εφαρμογή MySQL. Διαθέτει όλες τις κλάσεις από τις οντολογίες που περιγράφηκαν στο υποκεφάλαιο 4.2 καθώς και τα δεδομένα που συλλέχθηκαν από το Activity tracker. Η δεύτερη είναι σε μορφή RDFS και είναι στην ουσία η μικρότερη οντολογία την οποία ανέβασα στην εφαρμογή Dydra για να παραγματοποιήσω τις SPARQL ερωτήσεις. Διαθέτει τις οντολογίες AMOn και Activity Streams 2.0 καθώς και τα instances χρηστών και δραστηριοτήτων που δημιούργησα βασιζόμενος πάντα στα πραγματικά δεδομένα του Activity tracker. Οι βάσεις δεδομένων αλληλεπιδρούν με

την πλευρά του Server για την αναζήτηση και παρουσίαση των δεδομένων που ζητούνται.

- **Πλευρά Πελάτη(Client Side):** Η υπομονάδα αυτή είναι υπεύθυνη για τη δημιουργία κατάλληλων διεπαφών για την αποστολή αιτημάτων που επιτρέπουν στον χρήστη να αλληλεπιδρά με το server και κατ'επέκταση τη βάση δεδομένων. Για παράδειγμα είναι υπεύθυνη για τη δημιουργία των διεπαφών σύνδεσης και εγγραφής του χρήστη και των πινάκων στους οποίους θα εμφανίζεται η τελική πληροφορία.
- **Πλευρά Εξυπηρετητή(Server Side):** Η υπομονάδα αυτή έχει την αποκλειστική ευθύνη της αλληλεπίδρασης με τη βάση δεδομένων, ως αποτέλεσμα ενός αιτήματος του client. Με το πέρας της αλληλεπίδρασης αυτής, στέλνει κατάλληλη απάντηση πίσω στην client πλευρά είτε με την μορφή δεδομένων, είτε με τη μορφή σήματος, περιγράφοντας την έκβαση της ενέργειας που αιτήθηκε να πραγματοποιήσει. Με άλλα λόγια είναι υπεύθυνη για την εισαγωγή των στοιχείων του χρήστη, την παροχή και κωδικοποίηση των ζητούμενων πληροφοριών και άλλες υπολειτουργίες που θα αναλυθούν στην επόμενη υποενότητα.



Σχήμα 5.1: Μοντέλο αρχιτεκτονικής εφαρμογής

Στο σχήμα 5.1 φαίνεται το μοντέλο της αρχιτεκτονικής του συστήματος. Όπως φαίνεται η πλευρά του εξυπηρετητή συνδέεται με τις δύο βάσεις δεδομένων που περιγράφηκαν παραπάνω: την SQL βάση δεδομένων και τη σημασιολογική βάση δεδομένων μέσω της εφαρμογής Dydra. Η πλευρά του πελάτη επικοινωνεί με την πλευρά του εξυπηρετητή μέσω αιτημάτων και ο εξυπηρετητής πρέπει να επιστρέψει τις κατάλληλες διεπαφές για την διεκπεραίωση του αιτήματος. Η πλευρά του πελάτη μπορεί επίσης να επικοινωνεί με την εφαρμογή Dydra μέσα από links ανακατεύθυνσης τα οποία μεταφέρουν τον χρήστη στα endpoint URLs των ερωτήσεων που έγιναν πάνω στην RDFS βάση δεδομένων. Η σύνδεση με τον Activity tracker η οποία φαίνεται στο σχήμα δεν είναι μόνιμη και υπάρχει για την εμφάνιση πληροφοριών που συλλέχθηκαν από το εν λόγω σύστημα εφόσον ο χρήστης πραγματοποιήσει σχετικό αίτημα.

5.3 Περιγραφή Λειτουργιών Υποσυστήματος

Στο κεφάλαιο αυτό θα αναφερθούν οι λειτουργίες που απαιτείται να εκτελεί το κάθε υποσύστημα και θα περιγραφούν αναλυτικότερα καθεμία από αυτές. Συγκεκριμένα οι λειτουργίες που πρέπει να επιτελεί το σύστημα είναι οι ακόλουθες:

1. Εγγραφή ενός χρήστη στο σύστημα με συμπλήρωση πεδίων.
2. Σύνδεση και αποσύνδεση του χρήστη από το σύστημα.
3. Εμφάνιση των χρηστών και των δραστηριοτήτων του Activity tracker.
4. Σύνδεση με την βάση δεδομένων που χρησιμοποιήθηκε στην εφαρμογή Dydra.
5. Εμφάνιση των δεδομένων που προέκυψαν από όλες τις SPARQL ερωτήσεις που πραγματοποιήθηκαν στο Dydra.
6. Ύπαρξη links ανακατεύθυνσης τα οποία μεταφέρουν το χρήστη στα endpoint URLs της εφαρμογής Dydra.

Επειδή τα υποσυστήματα που ανήκουν στην εφαρμογή είναι πρακτικά λίγα σε αριθμό, επόμενο είναι, ότι καθένα από αυτά θα εκτελεί μια πληθώρα λειτουργιών. Επιπροσθέτως, επειδή υπάρχει client – server αρχιτεκτονική, η πλειονότητα των λειτουργιών απαιτεί και τα δύο υποσυστήματα, με τον client να παρέχει συχνά τη διεπιφάνεια και το server να πραγματοποιεί την υλοποίηση. Για το λόγο αυτό δεν θα αναλυθεί κάθε υποσύστημα χωριστά, αλλά θα γίνει ξεχωριστή εξέταση των λειτουργιών, τονίζοντας κάθε φορά τις αρμοδιότητες και ευθύνες της κάθε υπομονάδας.

5.3.1 Εγγραφή Χρήστη στην Εφαρμογή με Συμπλήρωση Πεδίων

Η λειτουργία αυτή είναι βασική για να επιτραπεί στον χρήστη πρόσβαση στην εφαρμογή. Ο Client είναι υπεύθυνος για τη δημιουργία κατάλληλων πεδίων για την εισαγωγή δεδομένων που αφορούν το όνομα χρήστη και τον κωδικό πρόσβασης. Ακολουθεί HTTP αίτημα και μεταφορά των δεδομένων αυτών στον Server, όπου πραγματοποιείται έλεγχος ώστε ο νέος χρήστης να μην έχει κοινά στοιχεία εγγραφής με κάποιον παλαιότερο. Αν υπάρξει σφάλμα τότε επιστρέφεται σχετικό μήνυμα στην πλευρά του πελάτη, διαφορετικά δημιουργείται νέος χρήστης με τα εισαγμένα στοιχεία και επιστρέφεται μήνυμα επιτυχίας στον πελάτη ο οποίος στη συνέχεια μεταφέρεται στην οθόνη σύνδεσης στο σύστημα.

5.3.2 Σύνδεση και Αποσύνδεση του Χρήστη από το Σύστημα

Εφόσον έχει πραγματοποιηθεί επιτυχημένα η εγγραφή ενός νέου χρήστη στο σύστημα, παρουσιάζεται η ανάγκη δημιουργίας κατάλληλης διεπαφής για τη σύνδεση του χρήστη στην εφαρμογή. Η πλευρά του πελάτη είναι υπεύθυνη για την δημιουργία των κατάλληλων πεδίων Username και Password τα οποία χρειάζεται ο χρήστης για να συνδεθεί. Ο εξυπηρετητής ελέγχει τα δεδομένα και είτε επιστρέφει μήνυμα λάθους στην περίπτωση εισαγωγής λανθασμένων δεδομένων είτε μεταφέρει τον χρήστη στην βασική οθόνη στην περίπτωση που τα στοιχεία σύνδεσης ήταν σωστά.

5.3.3 Εμφάνιση των Χρηστών και Δραστηριοτήτων του Activity Tracker

Ο χρήστης πρέπει να έχει τη δυνατότητα επισκόπησης των χρηστών και δραστηριοτήτων που είναι καταχωρημένες στον Activity tracker. Η επισκόπηση αυτή αφορά στο σύνολο των χρηστών και δραστηριοτήτων της εφαρμογής και όχι των συγκεκριμένων χρηστών στους οποίους πραγματοποιήθηκαν οι ερωτήσεις SPARQL και περιλαμβάνει λεπτομέρειες σχετικά με τους χρήστες (Όνομα, Ημερομηνία γέννησης κτλ) και τις δραστηριότητες (όνομα,

περιγραφή κτλ). Συνεπώς ο Client είναι υπεύθυνος για τη δημιουργία αυτών των διεπαφών και για την επικοινωνία με το Server με σκοπό την ανάκτηση των πληροφοριών που ζητήθηκαν.

5.3.4 Εμφάνιση των Χρηστών από την RDFS Βάση Δεδομένων

Πέρα από την εμφάνιση του συνόλου των χρηστών, ο επισκέπτης στην εφαρμογή πρέπει να έχει στη διάθεσή του τη λίστα των χρηστών πάνω στους οποίους έγιναν οι ερωτήσεις τις οποίες θα μπορεί να μελετήσει στις επόμενες σελίδες της εφαρμογής. Η λίστα αυτή είναι μικρή (περιέχει μόνο δύο ονόματα) αλλά είναι σημαντική διότι θα περιλαμβάνει πρόσβαση στις σελίδες όπου ο επισκέπτης μπορεί να δει τις κοινές δραστηριότητες των χρηστών καθώς και άλλες χρήσιμες πληροφορίες σχετικά με αυτές τις δραστηριότητες. Ο πελάτης είναι υπεύθυνος για τη δημιουργία αυτής της επιφάνειας και για την επικοινωνία με το server για την ανάκτηση των πληροφοριών.

5.3.5 Εμφάνιση των Δεδομένων από τις SPARQL Ερωτήσεις

Όταν ο χρήστης βρίσκεται στη σελίδα όπου εμφανίζονται οι χρήστες της οντολογίας πάνω στην οποία πραγματοποιήθηκαν οι ερωτήσεις SPARQL πρέπει να έχει στη διάθεσή του links τα οποία θα τον μεταφέρουν στις σελίδες όπου εμφανίζονται τα αποτελέσματα αυτών των ερωτήσεων. Στην πρώτη σελίδα θα μπορεί να δει links τα οποία θα τον μεταφέρουν στις οθόνες όπου παρουσιάζονται οι κοινές δραστηριότητες των χρηστών καθώς και οι δραστηριότητες που είναι καταγεγραμμένες για κάθε χρήστη ενώ από την οθόνη κοινών δραστηριοτήτων θα μπορεί να μεταφερθεί σε σελίδες με πιο συγκεκριμένα αποτελέσματα για την κάθε δραστηριότητα και σύγκριση αυτών ανάμεσα στους δύο χρήστες. Όπως και στις προηγούμενες λειτουργίες ο client πρέπει να δημιουργήσει αυτές τις επιφάνειες και τα links ανακατεύθυνσης ενώ είναι υπεύθυνος και για την επικοινωνία με το server ο οποίος επιστρέφει τις πληροφορίες που ζητάει ο χρήστης.

5.3.6 Δημιουργία Redirect Links στα Endpoint URLs του Dydra

Όταν ο χρήστης βρίσκεται στην εκάστοτε σελίδα όπου φαίνονται τα αποτελέσματα από μία από τις ερωτήσεις που έγιναν μέσω της γλώσσας SPARQL θα πρέπει να υπάρχουν links τα οποία θα μεταφέρουν τον χρήστη στα endpoint URLs της εφαρμογής Dydra. Η επιφάνεια στην οποία θα μεταφέρεται ο χρήστης δεν παρουσιάζει κάτι διαφορετικό όσον αφορά τις εμφανιζόμενες πληροφορίες και περιλαμβάνεται στην εφαρμογή για λόγους πληρότητας και ενδεχόμενης γνωριμίας του χρήστη με την εφαρμογή Dydra. Ο client πρέπει να δημιουργήσει τα links ανακατεύθυνσης και είναι υπεύθυνος για την επικοινωνία με το server ο οποίος μεταφέρει τον χρήστη στις σελίδες της εφαρμογής Dydra.

5.4 Αρχιτεκτονική Συστήματος Πλαισίου DJANGO

Στο κεφάλαιο αυτό θα πραγματοποιηθεί περαιτέρω εστίαση στα δομικά τμήματα της εφαρμογής, εξειδικεύοντας όλα όσα αναφέρθηκαν στο προηγούμενο υποκεφάλαιο. Συγκεκριμένα θα γίνει παρουσίαση των οθονών που δημιουργήθηκαν σε μορφή λίστας και στη συνέχεια θα πραγματοποιηθεί αρχιτεκτονική ανάλυση με περισσότερες λεπτομέρειες για κάθε οθόνη που σχεδιάστηκε. Οι οθόνες που θα αναλυθούν είναι οι ακόλουθες:

1. **Start:** Οθόνη που προτρέπει νέους χρήστες να εγγραφούν στο σύστημα και παλαιότερους χρήστες να συνδεθούν στην εφαρμογή.
2. **Register:** Οθόνη εγγραφής στο σύστημα.

3. **Login:** Οθόνη εισόδου χρήστη στο σύστημα.
4. **Logged in:** Οθόνη που εμφανίζεται όταν ο χρήστης συνδεθεί. Είναι η βασική οθόνη από την οποία ο χρήστης μπορεί να μεταφερθεί στις υπόλοιπες.
5. **Index:** Οθόνη όπου εμφανίζονται ως λίστα οι δραστηριότητες που είναι καταχωρημένες στον Activity tracker.
6. **Userindex:** Οθόνη όπου εμφανίζονται οι χρήστες της εφαρμογής.
7. **Semanticusers:** Οθόνη όπου εμφανίζονται οι χρήστες που χρησιμοποιήθηκαν στις ερωτήσεις SPARQL.
8. **Commonactivities:** Οθόνη όπου εμφανίζονται οι κοινές δραστηριότητες των χρηστών.
9. **Activityentries:** Οθόνη όπου εμφανίζεται ο αριθμός καταχωρήσεων συγκεκριμένης δραστηριότητας για κάθε χρήστη.
10. **Activityfrequency:** Οθόνη όπου εμφανίζεται η συχνότητα με την οποία κάθε χρήστης πραγματοποιεί συγκεκριμένη δραστηριότητα.
11. **Activitymeantime:** Οθόνη όπου εμφανίζεται ο μέσος χρόνος που ξοδεύει ο κάθε χρήστης σε συγκεκριμένη δραστηριότητα.
12. **Compareactivity:** Οθόνη όπου πραγματοποιείται σύγκριση των χρηστών πάνω σε κάποια από τις δραστηριότητες που έχουν κοινές.

Σημείωση: Οι σελίδες Common activities, Activity entries, Activity frequency, Activity mean time, Compare activity και γενικά όλες οι όψεις που παρουσιάζουν δεδομένα τα οποία έχουν ληφθεί από τα SPARQL Queries που πραγματοποιήθηκαν στην εφαρμογή Dydra διαθέτουν και σύνδεσμους ανακατεύθυνσης οι οποίοι μεταφέρουν τον χρήστη στα Endpoint URLs του Dydra για τα συγκεκριμένα αποτελέσματα.

5.4.1 Οθόνη Start

1. Πρόκειται για την αρχική σελίδα του συστήματος.
2. Δίνει στο χρήστη τη δυνατότητα εγγραφής ή σύνδεσης στο σύστημα με την παροχή κατάλληλων links.
3. Οι συναρτήσεις χειρισμού που συνδέονται με την οθόνη αυτή παρουσιάζονται παρακάτω. Αυτές έχουν υλοποιηθεί με χρήση Python και συνθέτουν τα συσχετιζόμενα με την οθόνη views.

Πίνακας 5-1: Όψεις σχετικές με την οθόνη Start

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
start (diplomatiki/start)	Υπεύθυνη για τη φόρτωση της αρχικής σελίδας	Όταν ο χρήστης επισκεφτεί αυτή τη σελίδα η όψη φορτώνει την αρχική οθόνη (επιστρέφει δηλαδή ένα template ως απάντηση). Ο χρήστης έχει δύο επιλογές. Είτε την εγγραφή του στο σύστημα αν είναι νέος χρήστης ή τη σύνδεσή του στο σύστημα στην περίπτωση που διαθέτει ήδη λογαριασμό.

5.4.2 Οθόνη Register

1. Η οθόνη αυτή είναι υπεύθυνη για την εγγραφή του χρήστη στο σύστημα μέσα από τη συμπλήρωση πεδίων.
2. Περιλαμβάνει δύο υποχρεωτικά πεδία(όνομα χρήστη και κωδικό) με τη συμπλήρωση των οποίων ο χρήστης μπορεί να εγγραφεί στο σύστημα.Δεν περιλαμβάνει κάποιο είδος επαλήθευσης(π.χ. E-mail verification).
3. Αν ο χρήστης εισάγει όνομα ίδιο με αυτό κάποιου άλλου χρήστη εμφανίζεται σχετικό μήνυμα.
4. Η οθόνη συνδέεται με τις ακόλουθες όψεις:

Πίνακας 5-2:Όψεις σχετικές με την οθόνη Register

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
register(diplomatiki/register)	Υπεύθυνη για τη φόρτωση της σελίδας εγγραφής του χρήστη στο σύστημα και την αποστολή σχετικού μηνύματος αν ο χρήστης εισάγει ίδιο όνομα με κάποιον προηγούμενο	Η όψη αυτή ελέγχει την αποστολή των δεδομένων εγγραφής του χρήστη.Αν διαγνώσει κάποιο πρόβλημα εμφανίζει μήνυμα λάθους και επιστρέφει το χρήστη στη σελίδα εγγραφής. Διαφορετικά καταχωρεί τα στοιχεία του χρήστη στη βάση δεδομένων και τον μεταφέρει στην επόμενη σελίδα του συστήματος.
register_success (diplomatiiki/register_success)	Υπεύθυνη για τη φόρτωση της σελίδας επιτυχούς εγγραφής του χρήστη στο σύστημα.	Όταν ένας νέος χρήστης πραγματοποιήσει επιτυχή εγγραφή στην εφαρμογή τότε φορτώνεται αυτή η σελίδα η οποία προτρέπει τον χρήστη να συνδεθεί στο σύστημα με κατάλληλο link ανακατεύθυνσης.

5.4.3 Οθόνη Login

1. Είναι υπεύθυνη για τη σύνδεση του χρήστη στο σύστημα.
2. Δίνει τη δυνατότητα σύνδεσης μέσα από τη συμπλήρωση δύο πεδίων(όνομα χρήστη και κωδικός).
3. Οι συναρτήσεις χειρισμού που συνδέονται με την οθόνη αυτή παρουσιάζονται παρακάτω:

Πίνακας 5-3:Όψεις σχετικές με την οθόνη Login

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
login (diplomatiiki/login)	Υπεύθυνη για τη φόρτωση της οθόνης και τη σύνδεση του χρήστη στην εφαρμογή.	Η όψη αυτή φορτώνει το κατάλληλο template για τη σύνδεση του χρήστη στο σύστημα με την συμπλήρωση κατάλληλων πεδίων.Αν τα πεδία

		συμπληρωθούν σωστά ο χρήστης μεταφέρεται στην κεντρική οθόνη του συστήματος διαφορετικά μεταφέρεται στην οθόνη invalid_login.
Invalid_login (diplomatiki/invalid_login)	Υπεύθυνη για τη φόρτωση της οθόνης η οποία πληροφορεί τον χρήστη ότι τα στοιχεία σύνδεσής του δεν ήταν σωστά.	Σε περίπτωση που ο χρήστης αποστείλει δεδομένα σύνδεσης τα οποία δεν είναι σωστά, φορτώνεται αυτή η σελίδα η οποία δίνει στον χρήστη τη δυνατότητα να μεταφερθεί εκ νέου στην οθόνη σύνδεσης για να πληκτρολογήσει σωστά τα στοιχεία του.

5.4.4 Οθόνη Logged in

1. Πρόκειται για την κύρια σελίδα του συστήματος.
2. Προσφέρει στο χρήστη την επιλογή να αποσυνδεθεί από το σύστημα ή να προχωρήσει σε μία από τις επόμενες οθόνες για επισκόπηση χρηστών και δραστηριοτήτων.
3. Οι συναρτήσεις χειρισμού που συνδέονται με την οθόνη αυτή παρουσιάζονται παρακάτω:

Πίνακας 5-4: Όψεις σχετικές με την οθόνη Logged in

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
loggedin (diplomatiki/loggedin)	Υπεύθυνη για τη φόρτωση της κεντρικής σελίδας της εφαρμογής	Ενεργοποιείται όταν ο χρήστης επισκέπτεται τον σύνδεσμο της σελίδας αυτής
logout (diplomatiki/logout)	Υπεύθυνη για την αποσύνδεση του χρήστη από το σύστημα	Όταν ο χρήστης πατήσει το σύνδεσμο αυτό αποσυνδέεται από το σύστημα και εμφανίζεται η όψη logout. Τότε ο χρήστης έχει την επιλογή να ξανασυνδεθεί εφόσον το επιθυμεί.
Index(diplomatiki/index)	Υπεύθυνη για τη φόρτωση των δραστηριοτήτων σε μορφή λίστας.	Ενεργοποιείται όταν ο χρήστης πατήσει το σύνδεσμο που αντιστοιχεί στη σελίδα αυτή. Εμφανίζει το σύνολο των δραστηριοτήτων του Activity tracker σε μορφή λίστας.
userindex (diplomatiki/userindex)	Υπεύθυνη για τη φόρτωση των χρηστών σε μορφή λίστας.	Ενεργοποιείται όταν ο χρήστης πατήσει το σύνδεσμο που αντιστοιχεί στη σελίδα αυτή. Εμφανίζει

		το σύνολο των χρηστών του Activity tracker σε μορφή λίστας.
semanticusers (diplomatiki/semanticusers)	Υπεύθυνη για τη φόρτωση των χρηστών οι οποίοι έχουν στιγμιότυπα στην οντολογία που χρησιμοποιήθηκε στο Dydra.	Ενεργοποιείται όταν ο χρήστης πατήσει το σύνδεσμο που αντιστοιχεί στη σελίδα αυτή. Εμφανίζει τους δύο χρήστες που χρησιμοποιήθηκαν στις ερωτήσεις SPARQL και διαθέτει επιπρόσθετους συνδέσμους για πιο πολύπλοκα αποτελέσματα.

5.4.5 Οθόνη Index

1. Εμφανίζει το σύνολο των δραστηριοτήτων της εφαρμογής Activity tracker.
2. Κάθε δραστηριότητα αποτελεί και σύνδεσμο που μεταφέρει το χρήστη σε όψη με περισσότερες λεπτομέρειες για την εκάστοτε δραστηριότητα.
3. Η όψη της ίδιας της οθόνης userindex παρουσιάστηκε στο υποκεφάλαιο της οθόνης loggedin. Στον επόμενο πίνακα λοιπόν θα περιγραφεί η όψη που σχετίζεται με τις λεπτομέρειες κάθε δραστηριότητας.

Πίνακας 5-5: Όψεις σχετικές με την οθόνη Index

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
detail (diplomatiki/index/<activity id>)	Υπεύθυνη για τη φόρτωση λεπτομερειών για κάθε δραστηριότητα.	Κάθε δραστηριότητα που περιλαμβάνει ο Activity tracker διαθέτει έναν αριθμό ταυτότητας. Ο χρήστης όντας στην οθόνη index μπορεί πατώντας πάνω σε οποιαδήποτε δραστηριότητα να μεταφερθεί στη σελίδα η οποία περιγράφει τη δραστηριότητα αυτή.

5.4.6 Οθόνη User Index

1. Εμφανίζει τη λίστα των χρηστών που είναι εγγεγραμμένοι στην εφαρμογή Activity tracker.
2. Κάθε όνομα χρήστη αποτελεί και σύνδεσμο που μεταφέρει τον επισκέπτη σε όψη με περισσότερες λεπτομέρειες για τον εν λόγω χρήστη.
3. Η όψη της ίδιας της οθόνης index παρουσιάστηκε στο υποκεφάλαιο της οθόνης loggedin. Στον επόμενο πίνακα λοιπόν θα περιγραφεί η όψη που σχετίζεται με τις λεπτομέρειες κάθε χρήστη.

Πίνακας 5-6: Όψεις σχετικές με την οθόνη User Index

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
userdetail	Υπεύθυνη για τη φόρτωση	Ο χρήστης της εφαρμογής

(diplomatiki/userindex/ <user id>)	λεπτομερειών για κάθε χρήστη.	όντας στην όψη που εμφανίζει τη λίστα των χρηστών του Activity tracker μπορεί πατώντας πάνω σε κάποιο όνομα να μεταφερθεί στη σελίδα λεπτομερειών του αντίστοιχου χρήστη. Οι λεπτομέρειες που εμφανίζονται είναι πολύ βασικές και περιλαμβάνουν όνομα, επώνυμο και αν ο χρήστης έχει κάνει ποτέ login στην εφαρμογή του Activity tracker.
---------------------------------------	----------------------------------	---

5.4.7 Οθόνη Semantic Users

1. Εμφανίζει τα ονόματα των δύο χρηστών που χρησιμοποιήθηκαν στο Dydra για τα SPARQL queries.
2. Περιλαμβάνει συνδέσμους οι οποίοι μεταφέρουν τον χρήστη σε όψεις όπου μπορεί να δει περισσότερες λεπτομέρειες σχετικά με τις δραστηριότητες των χρηστών.
3. Οι όψεις που συνδέονται με τη σελίδα Semantic Users περιγράφονται παρακάτω.

Πίνακας 5-7: Όψεις σχετικές με την οθόνη Semantic Users

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
<username>Activities (diplomatiki/<username> activities)	Υπεύθυνη για την εμφάνιση των δραστηριοτήτων κάθε χρήστη.	Όταν ο χρήστης βρίσκεται στη σελίδα semantic users μπορεί να δει ένα πίνακα με επικεφαλίδες τα ονόματα των δύο χρηστών κάτω από τα οποία υπάρχουν σύνδεσμοι οι οποίοι τον μεταφέρουν στις όψεις όπου μπορεί να δει τις δραστηριότητες στις οποίες συμμετέχει ο κάθε χρήστης.
Common Activities (diplomatiki/commonactivities)	Υπεύθυνη για την εμφάνιση των δραστηριοτήτων που οι δύο χρήστες έχουν κοινές.	Αν ο χρήστης πατήσει τον σύνδεσμο Common Activities θα μεταφερθεί στη σελίδα όπου εμφανίζονται οι κοινές δραστηριότητες των δύο χρηστών και επιπλέον σύνδεσμοι με πιο πολύπλοκες ερωτήσεις σχετικές με αυτές τις δραστηριότητες.
Activity Entries (diplomatiki/allactivityentries)	Υπεύθυνη για την εμφάνιση των καταχωρήσεων όλων	Εμφανίζει σε μορφή πίνακα όλες τις

	των δραστηριοτήτων και για τους δύο χρήστες.	δραστηριότητες που πραγματοποιούν οι δύο χρήστες καθώς και τον αριθμό των καταχωρήσεων του κάθε χρήστη για κάθε δραστηριότητα.
--	--	--

5.4.8 Οθόνη Common Activities

1. Εμφανίζει τις κοινές δραστηριότητες των χρηστών Ιωσήφ και Χρήστος.
2. Αποτελείται από έναν πίνακα ο οποίος περιέχει links για τη φόρτωση όψεων σχετικών με άλλες ερωτήσεις που έγιναν πάνω σε αυτές τις δραστηριότητες.
3. Η όψη Common Activities περιγράφηκε στο υποκεφάλαιο Semantic Users ενώ οι όψεις που σχετίζονται με την Common Activities είναι οι Activity Entries, Activity Frequency, Activity Mean Time και Compare Activity και θα περιγραφούν με λεπτομερείες στα επόμενα υποκεφάλαια.

5.4.9 Οθόνη Activity Entries

1. Εμφανίζει τις καταχωρήσεις των δύο χρηστών για την εκάστοτε κοινή δραστηριότητά τους.
2. Αποτελείται από τρία links ένα για κάθε κοινή δραστηριότητα τα οποία είναι τα diplomatiki/runningentries, diplomatiki/socialmediaentries και diplomatiki/statusupdateentries.
3. Οι όψεις που συνδέονται με αυτήν τη σελίδα παρουσιάζονται παρακάτω:

Πίνακας 5-8: Όψεις σχετικές με την οθόνη Activity Entries

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
<Activity Name> Entries (diplomatiki/<activity name>entries).Για παράδειγμα: diplomatiki/runningentries	Εμφανίζει το πλήθος των καταχωρήσεων συγκεκριμένης δραστηριότητας την οποία οι δύο χρήστες έχουν κοινή.	Όταν ο χρήστης επισκεφθεί έναν από τους τρεις συνδέσμους που σχετίζονται με τις καταχωρήσεις των κοινών δραστηριοτήτων των χρηστών θα μεταφερθεί σε μία όψη όπου υπάρχει πίνακας ο οποίος αντιστοιχεί τους χρήστες με τον αριθμό των καταχωρήσεων που έχουν για την συγκεκριμένη δραστηριότητα.

5.4.10 Οθόνη Activity Frequency

1. Εμφανίζει τη συχνότητα με την οποία ένας χρήστης πραγματοποιεί μία συγκεκριμένη δραστηριότητα. Σε επίπεδο SPARQL η συχνότητα βρέθηκε διαιρώντας τον αριθμό καταχωρήσεων μίας συγκεκριμένης δραστηριότητας με τις συνολικές καταχωρήσεις του κάθε χρήστη.
2. Αποτελείται από τρία links ένα για κάθε κοινή δραστηριότητα τα οποία είναι τα diplomatiki/runningfrequency, diplomatiki/socialmediafrequency και diplomatiki/statusupdatefrequency.

3. Οι όψεις που συνδέονται με αυτήν τη σελίδα παρουσιάζονται παρακάτω:

Πίνακας 5-9: Όψεις σχετικές με την οθόνη Activity Frequency

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
<Activity Name> Frequency (diplomatiki/<activity name>frequency).Για παράδειγμα: diplomatiki/runningfrequency	Εμφανίζει τη συχνότητα εκτέλεσης κάποιας δραστηριότητας και για τους δύο χρήστες.	Ανάλογα με την προηγούμενη όψη στον πίνακα της οθόνης Common Activities υπάρχουν τρεις σύνδεσμοι ένας για κάθε κοινή δραστηριότητα οι οποίοι μεταφέρουν το χρήστη στη σελίδα Activity frequency. Αποτελείται από έναν πίνακα ο οποίος διαθέτει πληροφορίες σχετικά με τις καταχωρήσεις της δραστηριότητας και τις συνολικές καταχωρήσεις για κάθε χρήστη καθώς και τις συχνότητες εκτέλεσης της εκάστοτε δραστηριότητας.

5.4.11 Οθόνη Activity Mean Time

1. Εμφανίζει το μέσο χρόνο που σπαταλά κάθε χρήστης για μία από τις κοινές δραστηριότητές τους.
2. Αποτελείται από τρία links ένα για κάθε κοινή δραστηριότητα τα οποία είναι τα diplomatiki/runningmeantime, diplomatiki/socialmediameantime και diplomatiki/statusupdatemeantime.
3. Οι όψεις που συνδέονται με αυτήν τη σελίδα παρουσιάζονται παρακάτω:

Πίνακας 5-10: Όψεις σχετικές με την οθόνη Activity Mean Time

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
<Activity Name> Mean Time (diplomatiki/<activity name>meantime).Για παράδειγμα: diplomatiki/runningmeantime	Εμφανίζει το μέσο χρόνο εκτέλεσης κάποιας δραστηριότητας και για τους δύο χρήστες.	Ανάλογα με την προηγούμενη όψη στον πίνακα της οθόνης Common Activities υπάρχουν τρεις σύνδεσμοι ένας για κάθε κοινή δραστηριότητα οι οποίοι μεταφέρουν το χρήστη στη σελίδα Activity mean time. Αποτελείται από έναν πίνακα ο οποίος αντιστοιχεί τους χρήστες με τον μέσο χρόνο που ξοδεύουν για μία συγκεκριμένη δραστηριότητα.

5.4.12 Οθόνη Compare Activity

1. Εμφανίζει τον αριθμό των καταχωρήσεων που έχει κάθε χρήστης για μία συγκεκριμένη δραστηριότητα,πραγματοποιεί σύγκριση και εμφανίζει σχετικό μήνυμα ανάλογα με το είδος της δραστηριότητας.
2. Αποτελείται από τρία links ένα για κάθε κοινή δραστηριότητα τα οποία είναι τα `diplomatiki/comparerunning`, `diplomatiki/comparesocialmedia` και `diplomatiki/comparestatusupdate`.
3. Οι όψεις που συνδέονται με αυτήν τη σελίδα παρουσιάζονται παρακάτω:

Πίνακας 5-11: Όψεις σχετικές με την οθόνη Compare Activity

Όνομα και Σύνδεσμος Όψης	Χρήση	Περιγραφή
Compare <Activity Name> (<code>diplomatiki/compare<activity name></code>).Για παράδειγμα: <code>diplomatiki/comparerunning</code>	Συγκρίνει τους δύο χρήστες όσον αφορά μία από τις κοινές τους δραστηριότητες.	Ανάλογα με την προηγούμενη όψη στον πίνακα της οθόνης Common Activities υπάρχουν τρεις σύνδεσμοι ένας για κάθε κοινή δραστηριότητα οι οποίοι μεταφέρουν το χρήστη στη σελίδα Compare Activity . Αποτελείται από έναν πίνακα ο οποίος διαθέτει πληροφορίες σχετικά με τις καταχωρήσεις της δραστηριότητας και τις συνολικές καταχωρήσεις για κάθε χρήστη καθώς και ένα σχετικό μήνυμα ανάλογα με τη δραστηριότητα.

6 Υλοποίηση Συστήματος

Έχοντας ολοκληρώσει το σχεδιασμό και την ανάλυση αρχιτεκτονικής του συστήματος, το παρόν κεφάλαιο θα εστιάσει στην ανάλυση ζητημάτων που αφορούν την υλοποίηση της εφαρμογής. Συγκεκριμένα, θα γίνει αναφορά στις πλατφόρμες, εργαλεία και βιβλιοθήκες που χρησιμοποιήθηκαν για την υλοποίησή της και θα παρουσιαστούν στιγμιότυπα της εφαρμογής.

6.1 Πλατφόρμες και Προγραμματιστικά Εργαλεία

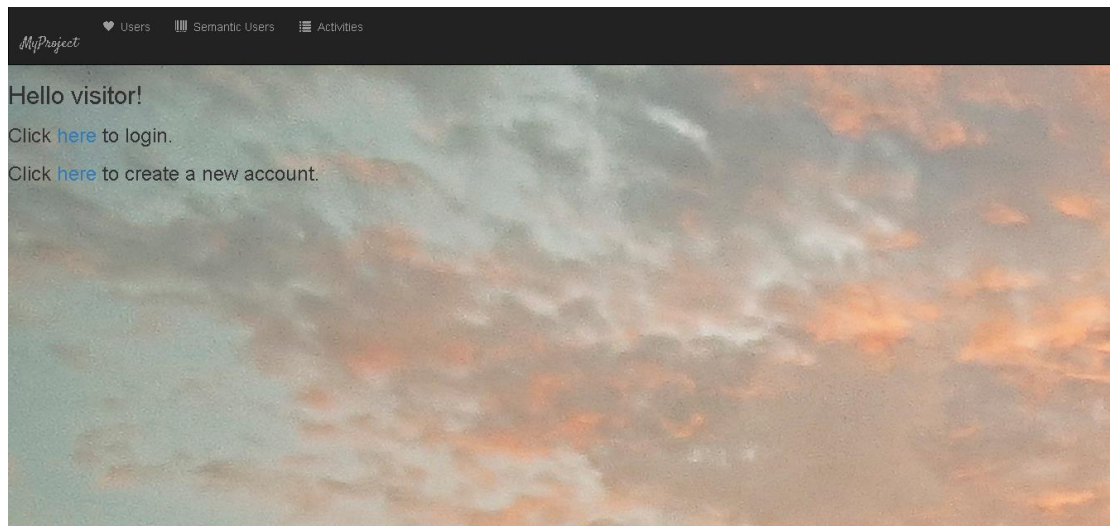
Στο υποκεφάλαιο αυτό θα γίνει αναφορά στα προγράμματα που χρησιμοποιήθηκαν σε κάθε στάδιο της υλοποίησης της διπλωματικής.

1. Για την ανάπτυξη της οντολογίας έγινε χρήση του προγράμματος **Protégé**. Πρόκειται για τον πιο ευρέως χρησιμοποιούμενο editor οντολογιών και αναπτύσσεται στο πανεπιστήμιο του Stanford. Παρέχει στον χρήστη κατάλληλες διεπαφές για να ορίζει οντολογίες, να συνδέει διαφορετικές οντολογίες. Διαθέτει plugins για τη γραφική αναπαράσταση των δεδομένων και reasoners οι οποίοι επικυρώνουν ότι το μοντέλο που έχει κατασκευαστεί είναι συνεπές.
2. Για τη μετατροπή της βάσεως δεδομένων του Activity Tracker από μορφή SQL σε RDFS έγινε χρήση του προγράμματος DB Transformer το οποίο πραγματοποιεί mapping από μία σχεσιακή βάση δεδομένων σε RDFS χρησιμοποιώντας τα εργαλεία D2RQ, Jena και OWL API. Η διαδικασία μπορεί να περιγραφεί ως εξής: το D2RQ συνδέεται στη σχεσιακή βάση δεδομένων, προσδιορίζει τη δομή της και φορτώνει αυτή τη δομή σε έναν RDF γράφο ο οποίος πηγάζει από το Jena graph. Στη συνέχεια το εργαλείο Jena μετατρέπει αυτόν το γράφο σε μορφή RDF/XML ενώ με το OWL API μπορούν τα σχεσιακά δεδομένα να μετατραπούν σε μία οντολογία της μορφής OWL/XML. Η τελική οντολογία είναι το ίδιο λεπτομερές με τη σχεσιακή βάση δεδομένων.
3. Για την πραγματοποίηση των SPARQL Queries πάνω στην τελική οντολογία χρησιμοποιήθηκε η εφαρμογή Dydra. Πρόκειται για μία cloud based βάση δεδομένων γραφημάτων στην οποία ο χρήστης μπορεί να φορτώσει ένα αρχείο το οποίο περιέχει μία σημασιολογική βάση δεδομένων σε σειριοποίηση Turtle, N-Triples, N-Quads, RDF/JSON, RDF/XML και RDFa. Στη συνέχεια ο χρήστης μπορεί εύκολα να γράψει κώδικα SPARQL ώστε να κάνει ερωτήσεις πάνω στην οντολογία του.
4. Για την ανάπτυξη της εφαρμογής έγινε χρήση του ολοκληρωμένου περιβάλλοντος ανάπτυξης PyCharm το οποίο αναπτύχθηκε από την εταιρεία Jet Brains και κάνει χρήση του πλαισίου Django το οποίο είναι υλοποιημένο σε γλώσσα προγραμματισμού Python. Το Django έχει ενσωματωμένη μια βάση δεδομένων SQLite και ένα server Apache για χρήση εντός του διαστήματος ανάπτυξης του συστήματος. Προσφέρει μεθόδους και τεχνικές εύκολης σχεδίασης και επικοινωνίας τόσο σε επίπεδο πελάτη-εξυπηρετητή όσο και σε επίπεδο εξυπηρετητή-βάσης δεδομένων. Η υλοποίηση των στατικών τμημάτων έγινε με απλή HTML η οποία είναι integrated στο PyCharm ενώ η γραφική διαμόρφωση πραγματοποιήθηκε με χρήση CSS3. Η CSS είναι μία γλώσσα υπολογιστή που χρησιμοποιείται για τον έλεγχο της εμφάνισης ενός εγγράφου (μίας ιστοσελίδας) που έχει γραφτεί σε γλώσσα HTML.

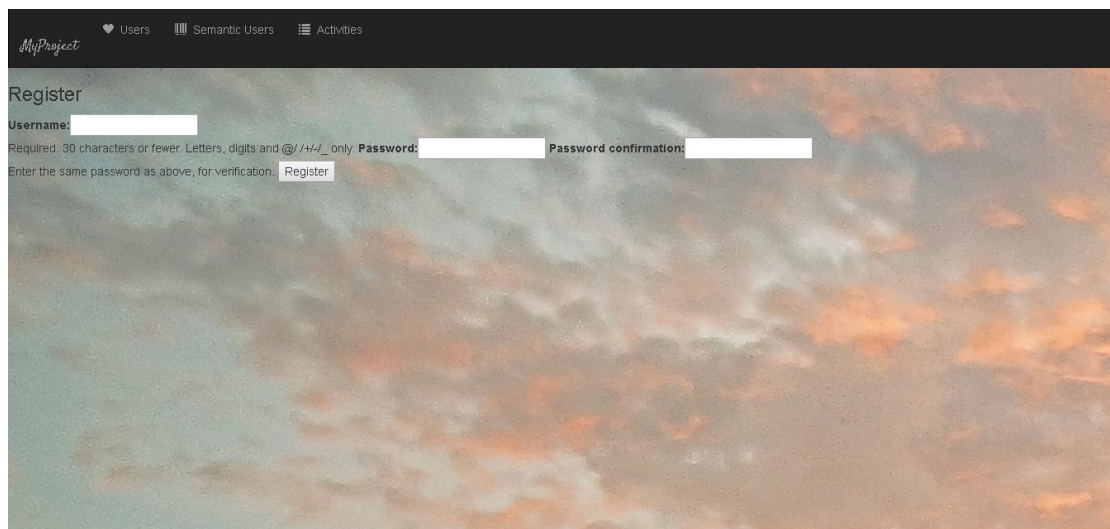
6.2 Στιγμιότυπα και Καθοδήγηση

Έχοντας ολοκληρώσει στο προηγούμενο κεφάλαιο την περιγραφή, τις λειτουργίες και την αρχιτεκτονική της εφαρμογής, στο παρόν υποκεφάλαιο θα παρουσιαστούν στιγμιότυπα της εφαρμογής σε ζεύγη στιγμιότυπο κείμενο και ανάλογα με τη σειρά με την οποία αναλύθηκαν οι οθόνες του συστήματος στο κεφάλαιο 5. Τα στιγμιότυπα έγιναν για έναν τυχαίο χρήστη με Username Chris.

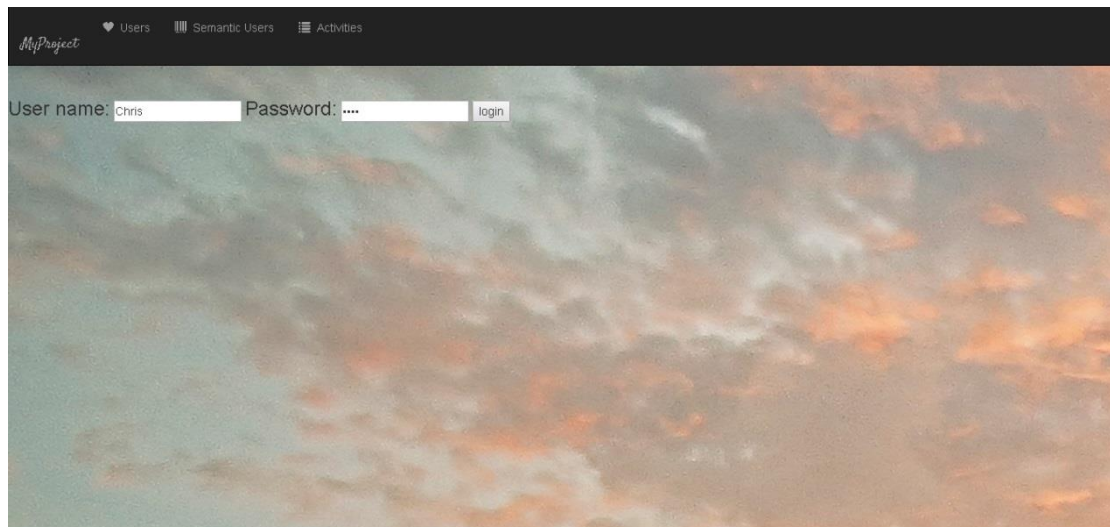
- Στιγμιότυπα οθονών σύνδεσης, εγγραφής και λανθασμένης εισαγωγής στοιχείων.



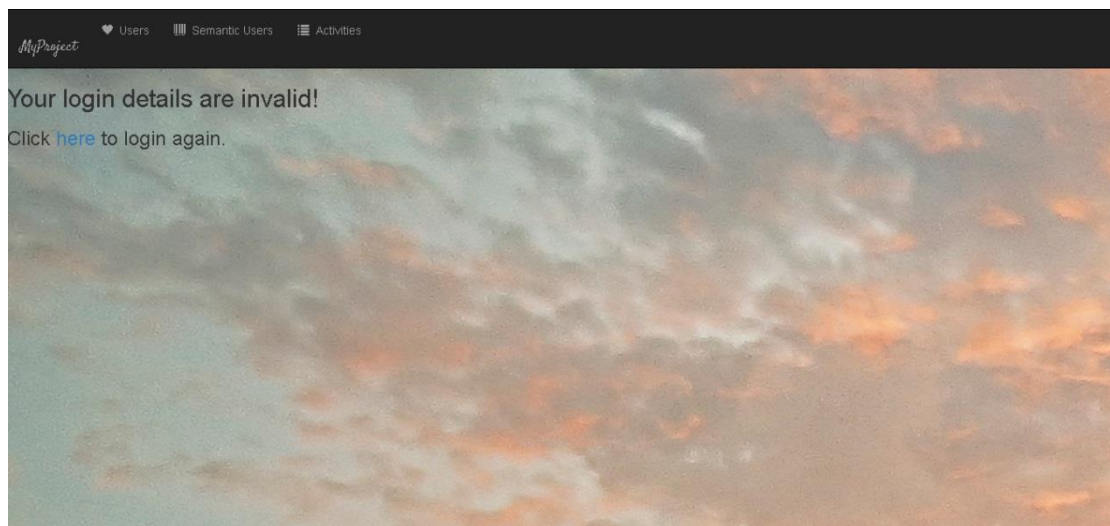
Εικόνα 6.1: Στιγμιότυπο οθόνης start



Εικόνα 6.2: Στιγμιότυπο οθόνης register

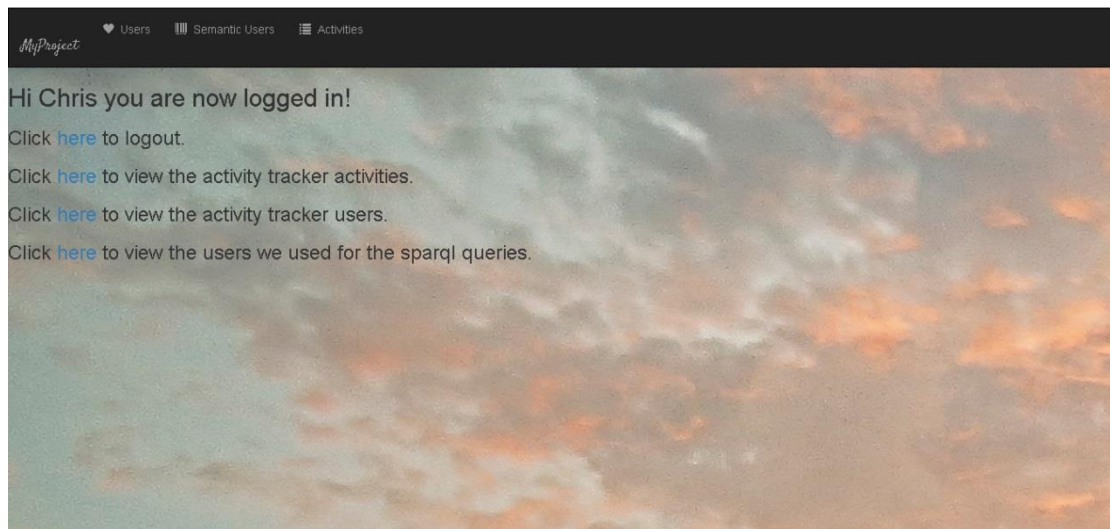


Εικόνα 6.3:Στιγμιότυπο οθόνης login

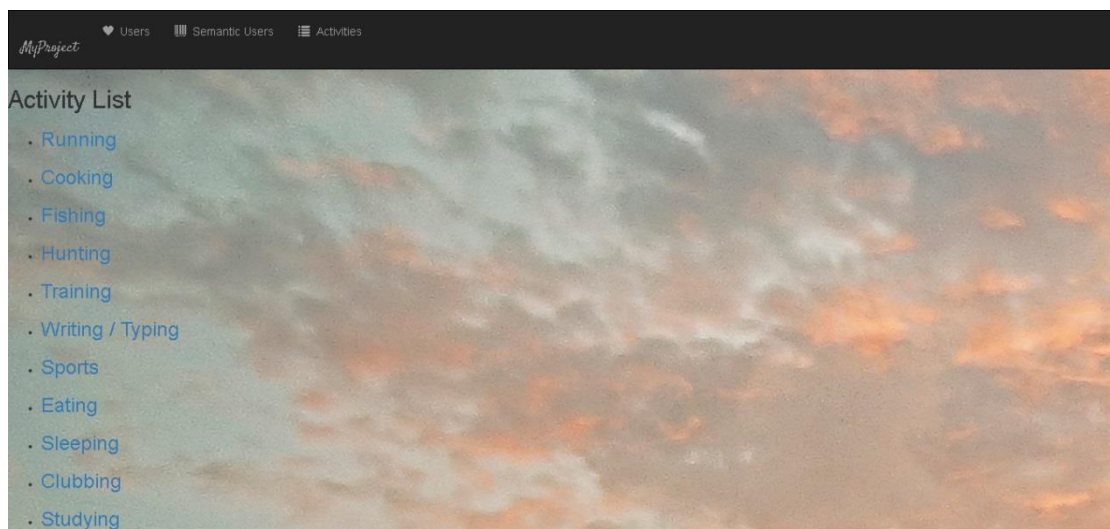


Εικόνα 6.4:Στιγμιότυπο οθόνης invalid_login

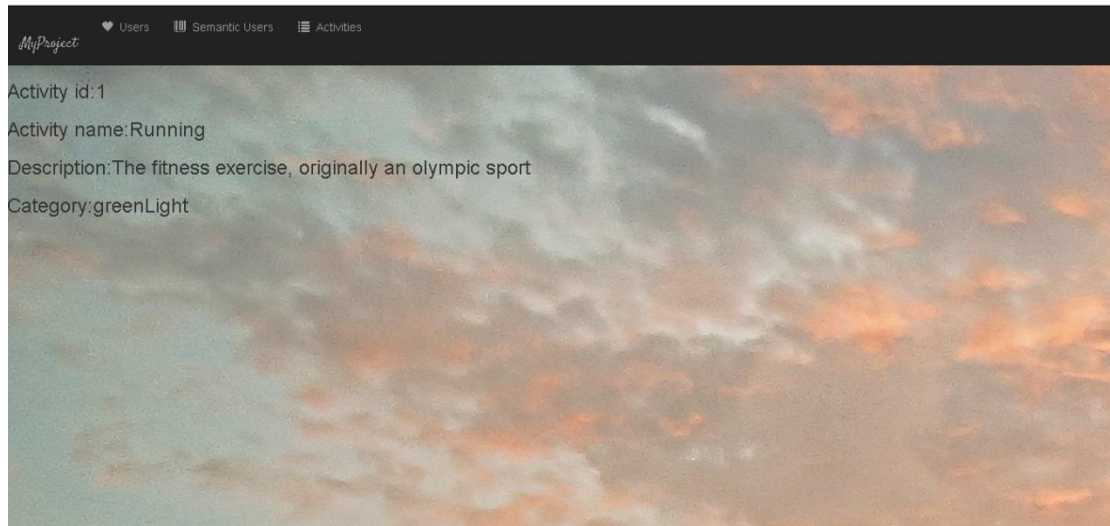
Οι λειτουργίες που επιτελούν οι παραπάνω οθόνες είναι είναι προφανείς και έχει γίνει ήδη περιγραφή στο προηγούμενο κεφάλαιο.Ωστόσο πρέπει να αναφερθεί ότι η χρήση ήδη χρησιμοποιούμενο username στην οθόνη register θα εμφανίσει σφάλμα κάτω από το πεδίο συμπλήρωσης και θα ζητηθεί από τον χρήστη να εισάγει διαφορετικό όνομα χρήστη.Αν τα πεδία στην οθόνη login συμπληρωθούν λανθασμένα ο χρήστης θα μεταφερθεί στην οθόνη invalid login η οποία του ζητάει να δοκιμάσει ξανά σύνδεση.Δεν υπάρχει δυνατότητα ανεύρεσης στοιχείων σε περίπτωση που ο χρήστης για παράδειγμα ξεχάσει τον κωδικό σύνδεσής του καθώς η εφαρμογή αναπτύχθηκε στο πλαίσιο διπλωματικής εργασίας και είναι εύκολο για τον χρήστη να δημιουργήσει νέο λογαριασμό σύνδεσης.



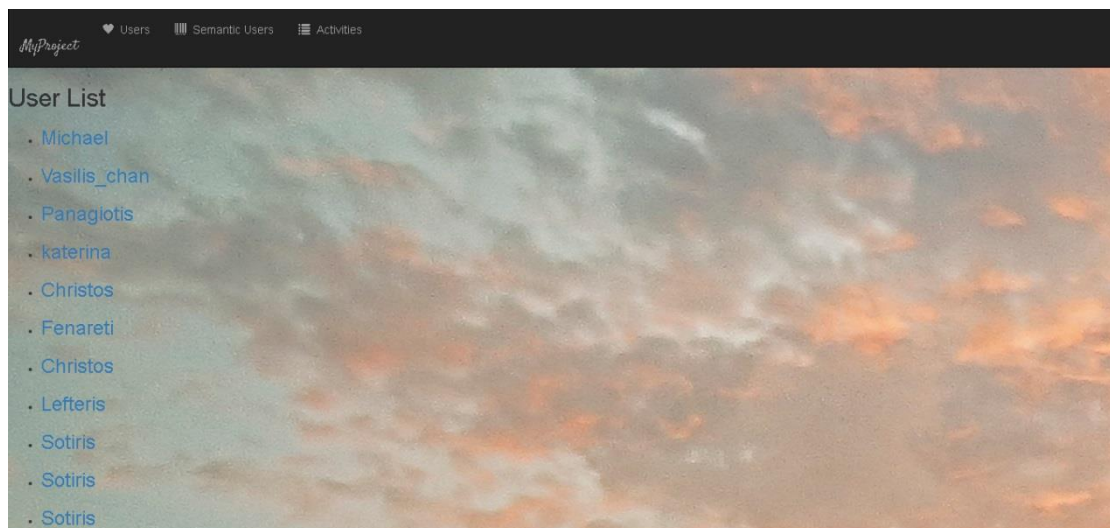
Εικόνα 6.5:Στιγμιότυπο οθόνης loggedin



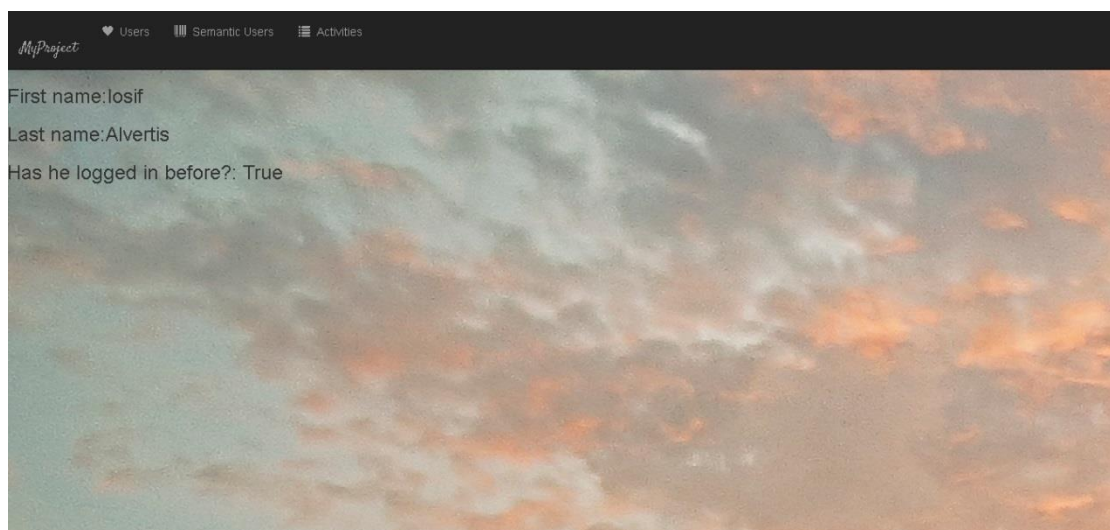
Εικόνα 6.6:Στιγμιότυπο οθόνης index



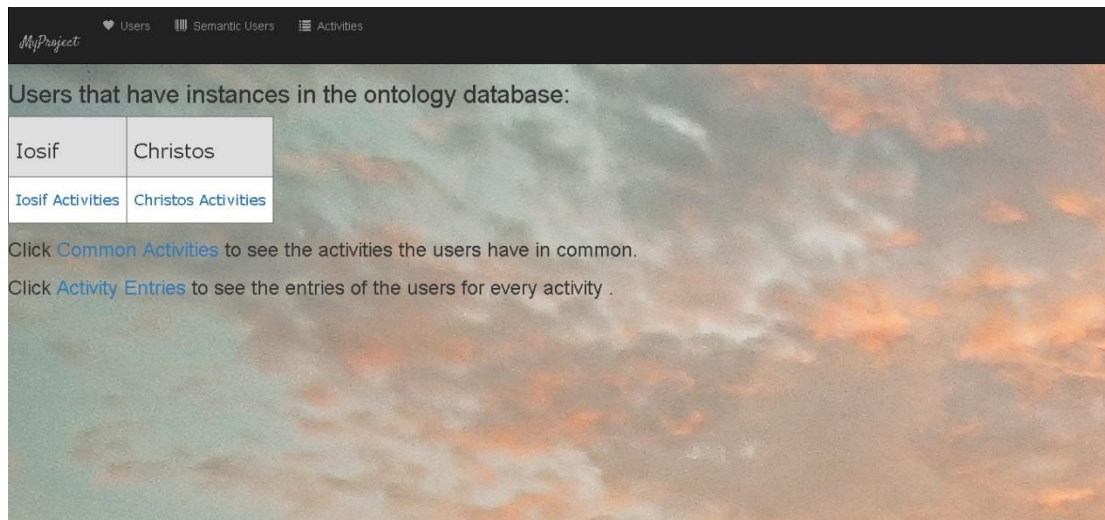
Εικόνα 6.7: Στιγμιότυπο οθόνης Activity details



Εικόνα 6.8: Στιγμιότυπο οθόνης Userindex

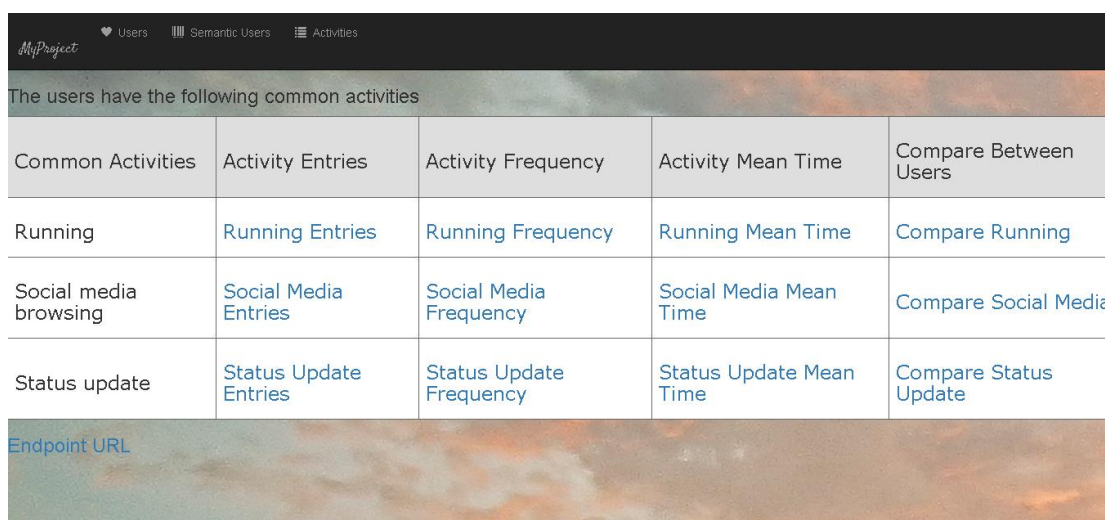


Εικόνα 6.9: Στιγμιότυπο οθόνης Userdetails



Εικόνα 6.10:Στιγμιότυπο οθόνης semantic users.

Στα παραπάνω πέντε στιγμιότυπα φαίνεται καταρχάς η βασική οθόνη της εφαρμογής με την οποία ο χρήστης μπορεί να επιλέξει αποσύνδεση ή μετάβαση σε κάποια άλλη οθόνη. Ανάμεσα στις οθόνες τις οποίες μπορεί να δει ο χρήστης είναι οι index και user index οι οποίες εμφανίζουν σε μορφή λίστας τις δραστηριότητες και τους χρήστες που υπάρχουν στη βάση δεδομένων του Activity Tracker ενώ στις οθόνες activity index και user index εμφανίζονται κάποιες λεπτομέρειες για τις δραστηριότητες και τους χρήστες αντίστοιχα. Στην οθόνη semantic users φαίνονται οι δύο χρήστες πάνω στις δραστηριότητες των οποίων πραγματοποιήθηκαν τα SPARQL Queries. Σε αυτήν την οθόνη παρέχονται σύνδεσμοι στους οποίους εμφανίζονται οι δραστηριότητες του κάθε χρήστη ξεχωριστά, ο αριθμός των καταχωρήσεων που έχει ο κάθε χρήστης για αυτές τις δραστηριότητες καθώς και οι κοινές δραστηριότητες που υπάρχουν μεταξύ τους.



Εικόνα 6.11:Στιγμιότυπο οθόνης Common Activities

CommonActivities
Running
Social media browsing
Status update

Εικόνα 6.12:Στιγμιότυπο από οθόνη του endpoint url του commonactivities

Στα παραπάνω στιγμιότυπα φαίνεται αρχικά η οθόνη common activities στην οποία εμφανίζονται οι κοινές δραστηριότητες των χρηστών σε μορφή πίνακα ενώ υπάρχουν και σύνδεσμοι για κάθε δραστηριότητα με περισσότερες πληροφορίες όπως ο αριθμός των καταχωρήσεων, η συχνότητα και ο μέσος χρόνος διάρκειας μίας δραστηριότητας και για τους δύο χρήστες. Στην εικόνα 22 φαίνονται τα ίδια αποτελέσματα όπως αυτά εμφανίζονται στο endpoint url του d4dra και περιλαμβάνει μόνο τις κοινές δραστηριότητες καθώς το SPARQL query έγινε συγκεκριμένα για αυτές. Οι επιπλέον πληροφορίες που εμφανίζονται στην εφαρμογή προστέθηκαν για καλύτερη εμφάνιση των αποτελεσμάτων ώστε ο χρήστης να έχει όλες τις δυνατές οθόνες διαθέσιμες σε μία μόνο όψη και να μπορεί να επιλέξει την πληροφορία που θέλει να δει.

Username	Running Entries
Iosif	5
Christos	3

Endpoint URL

Εικόνα 6.13:Στιγμιότυπο οθόνης Running Entries

MyProject Users Semantic Users Activities

Frequency of running

Iosif Entries	Christos Entries	Iosif Total Entries	Christos Total Entries	Iosif Frequency	Christos Frequency
5	3	20	15	0.25	0.2

Endpoint URL

Εικόνα 6.14:Στιγμιότυπο οθόνης running frequency

MyProject Users Semantic Users Activities

Mean Time of Running/User

Username	Running Mean Time
Iosif	41.2
Christos	22.333334

Endpoint URL

Εικόνα 6.15:Στιγμιότυπο οθόνης running mean time

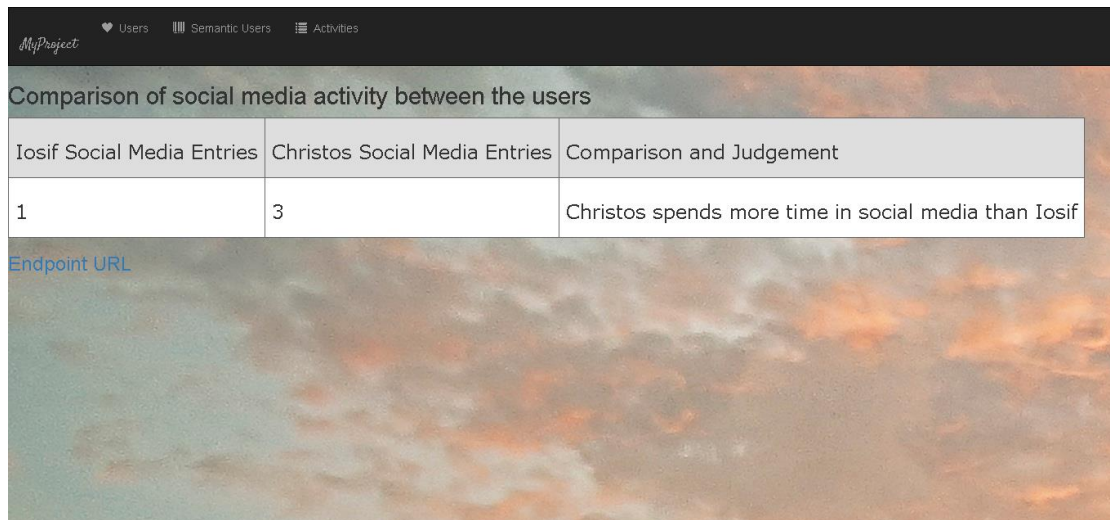
MyProject Users Semantic Users Activities

Comparison of running entries between the users

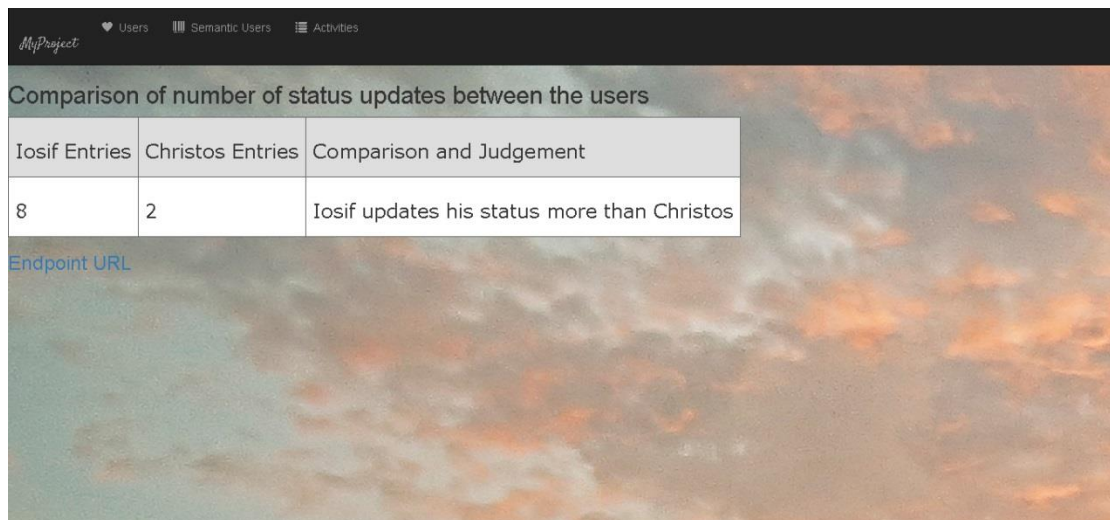
Iosif Running Entries	Christos Running Entries	Comparison and Judgement
5	3	Iosif is more athletic than Christos

Endpoint URL

Εικόνα 6.16:Στιγμιότυπο οθόνης compare running



Εικόνα 6.17:Στιγμιότυπο οθόνης compare social media



Εικόνα 6.18:Στιγμιότυπο οθόνης status update



Εικόνα 6.19:Στιγμιότυπο οθόνης endpoint url του comparestatusupdate

Στα τελευταία στιγμιότυπα εμφανίζονται οι οθόνες στις οποίες μπορεί να μεταβεί ο χρήστης από την οθόνη `common activities`. Οι πληροφορίες που περιέχουν γίνονται εύκολα κατανοητές και έχουν αναπτυχθεί με λεπτομέρειες στο προηγούμενο κεφάλαιο. Να σημειωθεί ότι κάθε μία από αυτές τις οθόνες διαθέτει σύνδεσμο που μεταφέρει τον χρήστη στο endpoint url των ίδιων ερωτήσεων στο Dydra όπως φαίνεται και στην εικόνα 29 για το endpoint url της οθόνης `compare status update`.

7 Επίλογος

Το κεφάλαιο αυτό θα αποτελέσει μια σύνοψη της συνολικής εργασίας, όπου θα παρουσιαστούν τα συμπεράσματα στα οποία αυτή κατέληξε και οι πιθανές προεκτάσεις που προέκυψαν κατά το στάδιο ανάπτυξης της εφαρμογής.

7.1 Σύνοψη και Συμπεράσματα

Μετά από μελέτη που έγινε πάνω σε ήδη υπάρχουσες οντολογίες του χώρου των δραστηριοτήτων παρατηρήθηκε έλλειψη γενικότερων μοντέλων τα οποία περιγράφουν τις καθημερινές δραστηριότητες οι οποίες δεν ανήκουν αποκλειστικά σε μία κατηγορία. Η έλλειψη θεσπισμένων οντολογιών και μοντέλων για την περιγραφή των ενεργειών και δραστηριοτήτων οδήγησε σε μια έρευνα πάνω στις θεωρητικές πτυχές του χώρου και μετά από ανάγνωση διαφόρων εργασιών του σχετικού τομέα, καθορίστηκε ένα μοντέλο ικανό να περιγράψει σε ικανοποιητικό βαθμό το σύνολο των ανθρώπινων δραστηριοτήτων. Η τελική οντολογία στην οποία κατέληξα είναι τριών επιπέδων με το υψηλότερο επίπεδο να παρέχει ένα γενικό μοντέλο περιγραφής των δραστηριοτήτων και των σχέσεων που τις διέπουν βασιζόμενος στη θεωρία δραστηριοτήτων και το μοντέλο που αναπτύχθηκε από τον Engstrom, το οποίο κρίνεται αρκετά ικανοποιητικό για την γενική περιγραφή των ανθρώπινων δραστηριοτήτων. Τα μεσαία επίπεδα μοντελοποιούν δεδομένα και δραστηριότητες από συστήματα κοινωνικής δικτύωσης καθώς και pervasive συστήματα. Το χαμηλότερο μοντέλο από την άλλη αποτελείται από μία οντολογία η οποία δίνει μεγαλύτερο βάθος και λεπτομέρειες στο τελικό μοντέλο ώστε αυτό να είναι ικανό να περιγράψει δραστηριότητες ποικίλων διαφορετικών κατηγοριών.

Έχοντας κατασκευάσει την τελική οντολογία έπρεπε αυτή να συνδεθεί με τα δεδομένα της βάσης του Activity Tracker τα οποία περιλαμβάνουν πραγματικούς χρήστες οι οποίοι επέλεξαν αυτήν την εφαρμογή για να καταγράφουν τις καθημερινές δραστηριότητές τους. Η σύνδεση αυτή ήταν απαραίτητη ώστε να προχωρήσω στη διαδικασία της συλλογιστικής πάνω στο μοντέλο για να διαπιστώσω την εγκυρότητά του και τη δυνατότητα εξαγωγής χρήσιμης πληροφορίας σχετικής με τα πραγματικά δεδομένα της βάσης. Σε αυτό το σημείο εμφανίστηκε ένα βασικό μειονέκτημα της διαδικασίας. Το μοντέλο στο οποίο τελικά κατέληξα ήταν αρκετά μεγάλο (περιείχε περισσότερα από 170000 ορίσματα) και το πρόγραμμα Protégé το οποίο χρησιμοποιήθηκε για τη σύνδεση των οντολογιών δεν μπορούσε να το αντέξει με αποτέλεσμα να παρουσιάζει προβλήματα στην απόδοσή του και να μην μπορεί να κάνει reasoning στα δεδομένα. Η έλλειψη κατάλληλων εργαλείων τα οποία μπορούν να αντέξουν μεγάλα και λεπτομερή μοντέλα ίσως είναι και ένας από τους βασικούς λόγους που τα περισσότερα οντολογικά μοντέλα είναι σχετικά μικρά και έχουν σχέση με συγκεκριμένες κατηγορίες δραστηριοτήτων.

Εξαιτίας αυτού του προβλήματος αναγκάστηκα να ακολουθήσω διαφορετική στρατηγική για να μπορέσω να εξάγω αποτελέσματα από τα δεδομένα χρησιμοποιώντας γλώσσα ερωτημάτων. Συνέθεσα μία μικρότερη οντολογία η οποία περιελάμβανε τα απολύτως απαραίτητα όσον αφορά τις κλάσεις και τα properties αυτών των κλάσεων. Όπου διέκρινα ότι υπάρχει έλλειψη κάποιας κλάσης ή ιδιότητας τη δημιούργησα μόνος μου. Τέλος αντί να συνδέσω πάνω στην οντολογία το σύνολο των δεδομένων από τον Activity Tracker χρησιμοποίησα δύο χρήστες και τις καταχωρημένες δραστηριότητες που είχαν στη βάση δεδομένων. Η οντολογία η οποία προέκυψε ήταν πολύ πιο εύκολη στην διαχείρησή της και στην επεξεργασία της. Ως αποτέλεσμα μπόρεσα να την ανεβάσω στην εφαρμογή Dydra και

να πραγματοποιήσω ένα σύνολο από Queries τα οποία επιστρέφουν χρήσιμη πληροφορία σχετικά με τους δύο χρήστες.

Στο τελικό στάδιο της εργασίας αναπτύχθηκε εφαρμογή χρησιμοποιώντας το πλαίσιο Django το οποίο κρίθηκε κατάλληλο για το σχεδιασμό ενός συστήματος το οποίο θα συνδέεται με τη βάση δεδομένων(οντολογία),θα παίρνει τα αποτελέσματα των ερωτήσεων από το Dydra και θα τα παρουσιάζει σε μορφή τέτοια ώστε ο χρήστης να μπορεί να δει το σύνολο των δεδομένων σε μία οθόνη και να εξάγει χρήσιμα συμπεράσματα.Στο σημείο αυτό πρέπει να αναφερθεί ότι τα δεδομένα που διαχειρίζεται η εφαρμογή είναι στατικά καθώς προέρχονται από την οντολογία.Αυτό στην πράξη σημαίνει ότι η εισαγωγή επιλέον χρηστών και δραστηριοτήτων δεν μπορεί να γίνει δυναμικά μέσα από την εφαρμογή και να πραγματοποιηθούν οι ίδιες ερωτήσεις πάνω στα νέα δεδομένα.Για να γίνει κάτι τέτοιο πρέπει αρχικά να προστεθούν καινούρια individuals στην οντολογία,στη συνέχεια να ανέβει αυτή εκ νέου στην εφαρμογή Dydra και να προσαρμοστούν οι ερωτήσεις ώστε να δίνουν αποτελέσματα για τους νέους χρήστες.Αυτό είναι και ένα βασικό μειονέκτημα της εφαρμογής καθώς η εκπόνηση μίας τέτοιας διαδικασίας απαιτεί γνώση των συστημάτων που χρησιμοποιήθηκαν και μπορεί να αποθαρρύνει τον χρήστη καθώς είναι χρονοβόρα και κουραστική.Παρ'όλα αυτά η ανάπτυξη της εφαρμογής κρίνεται επιτυχημένη όσον αφορά το κομμάτι της σύνδεσης με τα δεδομένα και της παρουσίασης των αποτελεσμάτων.Χωρίς αυτήν η παρουσίαση των αποτελεσμάτων θα γινόταν αποκλειστικά μέσα από τα endpoint urls του Dydra δυσχεραίνοντας έτσι την κατανόησή τους από τον χρήστη.

7.2 Μελλοντικές Επεκτάσεις

Κλείνοντας τη διπλωματική εργασία θα γίνει αναφορά σε ενδεχόμενες μελλοντικές επεκτάσεις της εφαρμογής.

Στην κατάσταση στην οποία βρίσκεται η εφαρμογή με το κλείσιμο της διπλωματικής, αποτελεί στην ουσία μία αρχιτεκτονική η οποία δέχεται στατικά δεδομένα από μία βάση και παρουσιάζει τα αποτελέσματα συγκεκριμένων ερωτήσεων που έγιναν πάνω σε αυτή.

Ως αποτέλεσμα η εφαρμογή χρήζει περαιτέρω ανάπτυξης ώστε να μπορεί να υλοποιεί κάποιες βασικές επεκτάσεις.Αρχικά η εφαρμογή θα μπορούσε να επεκταθεί με τη μεταφορά των ερωτήσεων από το επίπεδο του Dydra μέσα στην ίδια την εφαρμογή καθώς στο παρόν σημείο τα αποτελέσματα κατεβάζονται από το Dydra σε μορφή αρχείου JSON και συμπεριλαμβάνονται μέσα στην εφαρμογή.Κάτι τέτοιο θα επέτρεπε τη σύνθεση νέων ερωτήσεων στην πλατφόρμα που σχεδιάστηκε και δεν θα γινόταν απλή μεταφορά των αποτελεσμάτων από επίπεδο SPARQL στην οθόνη εμφάνισης των αποτελεσμάτων.Το σύστημα θα γινόταν ακόμα πιο χρήσιμο με την κατασκευή μοντέλων με τα οποία ο χρήστης θα μπορεί να συνδεθεί(για παράδειγμα σύνδεση μέσω Facebook) και να καταγράψει τις δικές του δραστηριότητες,να βρει άτομα από τον κοινωνικό του κύκλο και να συγκρίνει τις δραστηριότητες και την καθημερινότητά του με αυτούς.Με βάση τα παραπάνω θα ήταν εξαιρετικά ενδιαφέρουσες οι ακόλουθες επεκτάσεις:

1. Δυνατότητα σύνδεσης του χρήστη μέσα από μέσα κοινωνικής δικτύωσης και ανάπτυξη γραφικού περιβάλλοντος ώστε να μπορεί να καταγράψει τα δικά του δεδομένα και να εφαρμόσει τις ερωτήσεις της εφαρμογής στις δικές του δραστηριότητες.
2. Δυνατότητα υποστήριξης υπηρεσιών συγχρονισμού δεδομένων(π.χ.Facebook status update).

3. Σύνθεση νέων SPARQL ερωτήσεων οι οποίες θα παρουσιάζουν πιο πολύπλοκα αποτελέσματα όπως συναίσθημα του χρήστη κατά την εκτέλεση μίας δραστηριότητας, στόχος της δραστηριότητας κτλ. Αυτό θα μπορούσε να γίνει είτε με την εισαγωγή νέων δεδομένων στην οντολογία ή με τη μεταφορά των ερωτήσεων στην εφαρμογή για να μην χρειάζεται να γίνει η διαδικασία μεταφοράς των JSON αρχείων από το Dydra στην εφαρμογή κάθε φορά που επεκτείνεται αυτή με μία νέα ερώτηση.
4. Παροχή παραπάνω μορφών επισκόπησης και οπτικοποίησης δεδομένων με διαγράμματα, παροχή στατιστικών στοιχείων κτλ.
5. Δημιουργία μοντέλων τα οποία θα επέτρεπαν τη μεταφορά της οντολογίας στην εφαρμογή με δυναμική μορφή. Αυτό σημαίνει ότι ένας χρήστης θα μπορεί μέσα από την εφαρμογή να εισάγει στοιχεία στο σύστημα και να τα αποθηκεύσει στη βάση δεδομένων.
6. Ύπαρξη πεδίου αναζήτησης ώστε ο χρήστης να μπορεί να ψάξει για στοιχεία και αποτελέσματα πάνω σε μία συγκεκριμένη διαδικασία που τον ενδιαφέρει. Αυτή η επέκταση θα είχε νόημα σε κάποιο μελλοντικό στάδιο της εφαρμογής καθώς επί του παρόντος διαχειρίζεται συγκεκριμένα δεδομένα και αποτελέσματα για τα οποία δεν χρειάζεται πεδίο αναζήτησης.
7. Υλοποίηση ιεραρχίας και υποδιαίρεσης των δραστηριοτήτων και των αποτελεσμάτων στο επίπεδο της θεωρίας των δραστηριοτήτων. Με άλλα λόγια πως μοντελοποιείται και εξηγείται κάποιο αποτέλεσμα σύμφωνα με την Activity Theory.

8 Βιβλιογραφία

Engestrom, Yrjo, Miettinen, Reijo, Punamaki, Raija-Leena. (1999). Perspectives on Activity Theory.

Engestrom, Yrjo. (1987). Learning by Expanding: An Activity-theoretical Approach to Developmental Research.

Georgios Meditskos, Stamatia Dasiopoulou, Vasiliki Efstathiou, and Ioannis Kompatsiaris. (2013). Ontology Patterns for Complex Activity Modelling.

Dean Allemang, Jim Hendler (2008). Semantic Web for the Working Ontologist

Bob DuCharme(2013). Learning SPARQL :Quering and Updating with SPARQL 1.1

Stan Karanasios, Dhavalkumar Thakker, Lydia Lau, David Allen, Vania Dimitrova, Alistair Norman(2013). Making Sense of Digital Traces: An Activity Theory Driven Ontological Approach

Evaggelos Arbanitakis(2015). Ανάλυση και Μοντελοποίηση Ανθρωπίνων Δραστηριοτήτων στα Πληροφοριακά Συστήματα, και Ανάπτυξη Πρότυπης Εφαρμογής για Καταγραφή, Παρουσίαση και Παρακολούθηση Προσωπικών Δραστηριοτήτων

Gouhua Bai, Yang Guo(2015). Activity Theory Ontology for Knowledge Sharing in E-health

Anders Kofod-Petersen, Jorg Cassens(2006). Using Activity Theory to Model Context Awareness

9 Παραρτήματα

Ακολουθούν τα παραρτήματα της διπλωματικής αναλύοντας ζητήματα τα οποία δεν έχουν άμεση σχέση με το αντικείμενο της διπλωματικής αλλά συνδέονται με αυτή.

9.1 Παράρτημα I:Οδηγίες εγκατάστασης

Στο υποκεφάλαιο αυτό θα παρουσιαστούν οι οδηγίες για την εγκατάσταση της εφαρμογής σε τοπικό υπολογιστή. Η διαδικασία χωρίζεται σε δύο (2) μέρη. Το 1ο περιλαμβάνει τη λήψη των απαραίτητων προγραμματιστικών εργαλείων και πηγαίου κώδικα που απαιτούνται για την ομαλή λειτουργία της εφαρμογής, ενώ το 2ο περιλαμβάνει τη σύνδεση της εφαρμογής με τα εργαλεία, την πραγματοποίηση αλλαγών και την εκτέλεση εντολών αρχικοποίησης.

- **Λήψη Προγραμματιστικών Πλατφορμών, Εργαλείων & Πηγαίου Κώδικα**

Η διαδικασία λήψης και εγκατάστασης των απαραίτητων μονάδων θα περιγραφεί μέσω των ακόλουθων βημάτων. Αυτή απευθύνεται σε χρήστες λειτουργικού συστήματος Windows, αλλά η διαδικασία είναι παρεμφερής και για λειτουργικά συστήματα Mac OS, Linux, Ubuntu κ.λπ.:

1. Λήψη και εγκατάσταση της **Python 2.7.9** (λήψη) ή νεότερης έκδοσης της Python
2. Μετά το τέλος της εγκατάστασης, απαιτείται τροποποίηση της μεταβλητής συστήματος **Path**, έτσι ώστε να περιλαμβάνει το φάκελο εγκατάστασης της Python, όπως επίσης και τον υποφάκελο `scripts`. Σε περιβάλλον windows, αν αυτή είχε εγκατασταθεί στην τοποθεσία **C:/Python27**, τότε θα έπρεπε να προστεθεί το κείμενο **;%C:\Python27%;C:\Python27\Scripts** στη μεταβλητή Path.
2. Λήψη και εγκατάσταση του framework **Django 1.8** ή νεότερης έκδοσης αυτού. Έχοντας εγκαταστήσει την γλώσσα Python και έχοντας ρυθμίσει τη μεταβλητή Path, αυτό γίνεται εύκολα ανοίγοντας τη γραμμή εντολών (command line) και πληκτρολογώντας την εντολή **pip install Django==1.8.2** από οπουδήποτε στο σύστημα. Αν υπάρχει νεότερη έκδοση διαθέσιμη, τότε ο αριθμός 1.8.2 πρέπει να τροποποιηθεί ώστε να αντιστοιχεί στην έκδοση αυτή.
3. Λήψη και εγκατάσταση του **Pycharm Professional Edition 4.5.x** (λήψη) ή ενός άλλου IDE κατάλληλου για την ανάπτυξη εφαρμογών Django. Το βήμα αυτό είναι προαιρετικό αν ο προγραμματιστής εργάζεται σε περιβάλλον γραμμής εντολών.
4. Λήψη του πηγαίου κώδικα της εφαρμογής από το σύνδεσμο <https://github.com/Christos6/diplomat/>.
5. Εάν ο κώδικας λήφθηκε υπό μορφή .zip, αποσυμπίεση σε μια γνωστή τοποθεσία, εκκίνηση γραμμής εντολών εντός της τοποθεσίας αυτής και εκτέλεση της εντολής **pip install -r requirements.txt**, όπου requirements είναι το όνομα του αρχείου κειμένου που περιλαμβάνει τις απαραίτητες βιβλιοθήκες και εξωτερικές επιπρόσθετες οντότητες που απαιτεί το σύστημα. Το αρχείο αυτό είναι μέρος του κώδικα και ενδέχεται να περιέχει παλαιότερες εκδόσεις των βιβλιοθηκών.

- **Σύνδεση, Πραγματοποίηση Αλλαγών και Αρχικοποίηση Συστήματος**

Στο σημείο αυτό έχουν εγκατασταθεί όλες οι απαραίτητες μονάδες και εργαλεία. Ακολουθώντας τα παρακάτω βήματα αρχικοποιείται το σύστημα και είναι σε θέση να εκτελείται σε περιβάλλον τοπικού εξυπηρετητή.

1. Μέσα από το περιβάλλον του Pycharm IDE, επιλογή ανοίγματος (open) ενός υπάρχοντος έργου (project), μετάβαση στην τοποθεσία όπου αποθηκεύτηκε ο πηγαίος κώδικας και επιλογή του φακέλου με όνομα **diplomat**.

2. Επιλογή **File** → **Settings** → **Project: diplomat** → **Project Interpreter** → Επιλογή της εγκατεστημένης έκδοσης της Python μέσω του φακέλου **Python27** → **Ok** και αναμονή για τη δεικτοδότηση.
3. Πληκτρολόγηση **Ctrl + Alt + R** για εκκίνηση των εντολών `manage.py`. Εκτέλεση της εντολής **migrate** για τη δημιουργία πινάκων στη βάση δεδομένων (αν εμφανιστεί επιλογή διαγραφής μοντέλου επιλέξτε **no**). Εναλλακτικά, εκκίνηση γραμμής εντολών, μετάβαση στο φάκελο **diplomat** και εκτέλεση της εντολής **python manage.py migrate**.
4. Πληκτρολόγηση **Ctrl + Alt + R** και εκτέλεση της εντολής **runserver** ή εκκίνηση γραμμής εντολών, μετάβαση στο φάκελο **diplomat** και εκτέλεση της εντολής **python manage.py runserver**. Αν όλα τα βήματα έχουν πραγματοποιηθεί σωστά, τότε στην διεύθυνση <http://127.0.0.1:8000/diplomatiki> πρέπει να εμφανίζεται η εφαρμογή. Ακόμα στη διεύθυνση <http://127.0.0.1:8000/admin> μπορεί κανείς να έχει πλήρη εποπτεία των μοντέλων – πινάκων της βάσης δεδομένων.

Πλέον η εφαρμογή έχει εγκατασταθεί και είναι πλήρως επεξεργάσιμη. Ο αναγνώστης καλείται να διαβάσει αναλυτικά τα κεφάλαια (5) και (6) της παρούσας εργασίας, για την κατανόηση της αρχιτεκτονικής, της οργάνωσης και του τρόπου λειτουργίας της εφαρμογής.