



**ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ**

**ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ**

**ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ**

**Ανάπτυξη συνεργατικών υπηρεσιών σημασιολογικού  
ιστού με χρήση βατών ασαφών περιγραφικών λογικών**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

του

**Λουκά Αναστασίου**

**Επιβλέπων :** Γιώργος Στάμου  
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2009





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

## Ανάπτυξη συνεργατικών υπηρεσιών σημασιολογικού ιστού με χρήση βατών ασαφών περιγραφικών λογικών

### ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**Αναστασίου Λουκά**

**Επιβλέπων :** Γιώργος Στάμου  
Επίκουρος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30<sup>η</sup> Ιουνίου 2009.

(Υπογραφή)

.....  
Σ. Κόλλιας  
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....  
Α. Γ. Σταφυλοπάτης  
Καθηγητής Ε.Μ.Π.

(Υπογραφή)

.....  
Γ. Στάμου  
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούνιος 2009

---

*(Υπογραφή)*

.....

**ΛΟΥΚΑΣ ΑΝΑΣΤΑΣΙΟΥ**

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

© 2009 – All rights reserved

---

## Περίληψη

Σκοπός της παρούσας διπλωματικής εργασίας είναι η υλοποίηση συνεργατικής υπηρεσίας οργάνωσης βιβλιογραφικών αναφορών με χρήση τεχνολογιών σημασιολογικών ιστού. Η υπηρεσία επιτρέπει τον ασαφή χαρακτηρισμό δημοσιεύσεων και την οργάνωση και ανάκτηση δημοσιεύσεων κάνοντας χρήση μηχανής συμπερασματολογίας η οποία βασίζεται σε περιγραφικές λογικές. Ως γλώσσα περιγραφικής λογικής επιλέχθηκε η ασαφής έκδοση της DL-Lite η οποία παρέχει πλούσια εκφραστικότητα συγκρίσιμη με τη εκφραστικότητα που παρουσιάζουν UML διαγράμματα κλάσεων ενώ ταυτόχρονα έχει βατή πολυπλοκότητα.

### **Λέξεις Κλειδιά:**

Σημασιολογικός Ιστός, Web 2.0, Περιγραφικές Λογικές, DL-Lite, ασαφής-DL-Lite, Οντολογίες,



---

## **Abstract**

The main objective of this diploma thesis was the implementation of a collaborative service for organizing bibliographic references using semantic web technologies. The service allows the fuzzy annotation of a reference using an Ontology while organizes and retrieves publications utilizing an inference engine which is based in Description Logics. As description logic language we selected the fuzzy version of DL-Lite: fuzzy DL-Lite which combines high expressivity (comparable to UML class diagrams) and tractability.

### **Keywords:**

Semantic Web, Web 2.0, Description Logics, DL-Lite, fuzzy-DL-Lite, Ontologies





## **Ευχαριστίες**

Η παρούσα διπλωματική εργασία εκπονήθηκε στο εργαστήριο Ψηφιακής Επεξεργασίας Σημάτων και Εικόνας του Εθνικού Μετσόβιου Πολυτεχνείου. Θα ήθελα να ευχαριστήσω θερμά τον επιβλέποντα υπεύθυνο της διπλωματικής μου εργασίας Γιώργο Στάμου για την καθοδήγησή του καθώς και τον ερευνητή του εργαστηρίου Βασίλη Τζουβάρα για την πολύτιμη συμβολή του. Τέλος θέλω να ευχαριστήσω τους γονείς μου Πιερή και Αγγελική η υποστήριξη των οποίων με βοήθησε τόσο στην εκπόνηση της εργασίας όσο και στην ολοκλήρωση των προπτυχιακών μου σπουδών.



# Πίνακας περιεχομένων

ΠΕΡΙΛΗΨΗ .....	5
ABSTRACT .....	7
ΕΥΧΑΡΙΣΤΙΕΣ.....	1
ΚΑΤΑΛΟΓΟΣ ΣΧΗΜΑΤΩΝ-ΠΙΝΑΚΩΝ.....	5
<b>1 ΕΙΣΑΓΩΓΗ .....</b>	<b>7</b>
1.1 ΟΡΓΑΝΩΣΗ ΚΑΙ ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ ΣΕ ΑΚΑΔΗΜΑΪΚΕΣ ΚΟΙΝΟΤΗΤΕΣ .....	7
1.2 ΑΝΤΙΚΕΙΜΕΝΟ ΔΙΠΛΩΜΑΤΙΚΗΣ .....	8
1.3 ΟΡΓΑΝΩΣΗ ΚΕΙΜΕΝΟΥ .....	9
<b>2 ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ.....</b>	<b>11</b>
2.1 BIBSONOMY .....	11
2.2 CITEULIKE .....	14
2.3 ΣΥΜΠΕΡΑΣΜΑΤΑ ΚΑΙ ΑΞΙΟΛΟΓΗΣΗ .....	15
<b>3 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ .....</b>	<b>17</b>
3.1 ΣΗΜΑΣΙΟΛΟΓΙΚΟ ΔΙΑΔΙΚΤΥΟ.....	17
3.1.1 Ο σημερινός παγκόσμιος ιστός.....	17
3.1.2 Το όραμα του σημασιολογικού διαδικτύου.....	18
3.1.3 Η αρχιτεκτονική του σημασιολογικού διαδικτύου .....	19
3.2 WEB 2.0.....	21
3.3 ΟΝΤΟΛΟΓΙΕΣ.....	31
3.4 ΠΕΡΙΓΡΑΦΙΚΕΣ ΛΟΓΙΚΕΣ .....	33
3.4.1 Γενικά.....	33
3.4.2 DL-Lite.....	38
3.4.3 Ασαφής DL-Lite (fuzzy-DL-Lite).....	42
<b>4 ΑΝΑΛΥΣΗ ΑΠΑΙΤΗΣΕΩΝ ΣΥΣΤΗΜΑΤΟΣ.....</b>	<b>47</b>
4.1 ΠΕΡΙΓΡΑΦΗ ΛΕΙΤΟΥΡΓΙΩΝ.....	47
4.1.1 Χαρακτηρισμός δημοσίευσης (annotation) .....	47
4.1.2 Αναζήτηση με βάση την ερευνητική περιοχή .....	48
4.1.3 Αναζήτηση με βάση τα περιγραφικά χαρακτηριστικά της δημοσίευσης .....	48
4.1.4 Ανάκτηση δημοσιεύσεων με βάση την ετικέτα (προσωπική ή της κοινότητας) .....	48
4.1.4 Περιήγηση,εποπτεία και ανάκτηση των δημοσιεύσεων άλλων χρηστών.....	48
4.2 ΑΡΧΙΤΕΚΤΟΝΙΚΗ .....	49

4.2.1	Υποσύστημα του πελάτη.....	50
4.2.2	Υποσύστημα του εξυπηρετητή.....	50
4.2.3	Υποσύστημα της μηχανής συμπερασματολογίας.....	51
4.2.4	Υποσύστημα της βάσης δεδομένων.....	51
4.3	ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ ΣΥΣΧΕΤΙΣΕΩΝ.....	51
4.4	ΣΧΕΣΙΑΚΟ ΔΙΑΓΡΑΜΜΑ ΒΑΣΗΣ ΔΕΔΟΜΕΝΩΝ.....	52
4.5	ΟΝΤΟΛΟΓΙΑ ΑΝΑΠΑΡΑΣΤΑΣΗΣ ΤΟΥ ΠΕΔΙΟΥ ΤΗΣ ΑΚΑΔΗΜΑΪΚΗΣ ΚΟΙΝΟΤΗΤΑΣ.....	62
4.5.1	Ανάλυση της οντολογίας.....	62
4.5.2	Παράδειγμα εισαγωγής δημοσίευσης.....	72
<b>5</b>	<b>ΥΛΟΠΟΙΗΣΗ.....</b>	<b>75</b>
5.1	ΛΕΠΤΟΜΕΡΕΙΕΣ ΥΛΟΠΟΙΗΣΗΣ.....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
5.1.1	Κλάση <i>bibTex</i> .....	76
5.1.2	Κλάση <i>UserBean</i> .....	80
5.1.3	Κλάση <i>commentInfo</i> .....	82
5.1.4	Κλάση <i>publicationBean</i> .....	83
5.1.5	Κλάση <i>MD5Util</i> .....	85
5.1.6	Κλάση <i>ResultComparator</i> .....	85
5.1.7	Κλάση <i>dbConnection</i> .....	87
5.2	ΠΑΡΟΥΣΙΑΣΗ ΧΡΗΣΗΣ ΤΟΥ ΣΥΣΤΗΜΑΤΟΣ ΜΕ ΤΗ ΒΟΗΘΕΙΑ ΣΕΝΑΡΙΩΝ.....	99
5.2.1	Εγγραφή, είσοδος χρήστη και εκκίνηση καινούργιας συνεδρίας.....	101
5.2.2	Εισαγωγή και χαρακτηρισμός νέας δημοσίευσης.....	106
5.2.3	Αναζήτηση με βάση την ερευνητική περιοχή.....	110
5.2.4	Αναζήτηση με βάση τα περιγραφικά χαρακτηριστικά.....	111
<b>6</b>	<b>ΕΠΙΛΟΓΟΣ.....</b>	<b>121</b>
6.1	ΣΥΝΟΨΗ ΚΑΙ ΣΥΜΠΕΡΑΣΜΑΤΑ.....	121
6.2	ΜΕΛΛΟΝΤΙΚΕΣ ΕΠΕΚΤΑΣΕΙΣ.....	122
<b>7</b>	<b>ΒΙΒΛΙΟΓΡΑΦΙΑ.....</b>	<b>123</b>

## Κατάλογος Σχημάτων-Πινάκων

ΕΙΚΟΝΑ 2.1.1 Η ΚΕΝΤΡΙΚΗ ΣΕΛΙΔΑ ΤΟΥ BIBSONOMY .....	12
ΕΙΚΟΝΑ 2.1.2 Η ΠΑΡΟΥΣΙΑΣΗ ΜΙΑΣ ΔΗΜΟΣΙΕΥΣΗΣ ΣΤΟ BIBSONOMY .....	13
ΕΙΚΟΝΑ 2.1.3 ΔΗΜΟΦΙΛΕΙΣ ΣΥΣΧΕΤΙΣΕΙΣ ΕΥΚΕΤΩΝ ΣΤΟ BIBSONOMY .....	14
ΕΙΚΟΝΑ 3.1.1 Η ΣΤΟΙΒΑ ΤΕΧΝΟΛΟΓΙΩΝ ΤΟΥ ΣΗΜΑΣΙΟΛΟΓΙΚΟΥ ΙΣΤΟΥ.....	21
ΕΙΚΟΝΑ 3.2.1 ΤΟ ΠΑΡΑΔΟΣΙΑΚΟ ΜΟΝΤΕΛΟ ΕΦΑΡΜΟΓΗΣ ΙΣΤΟΥ .....	24
ΕΙΚΟΝΑ 3.2.2 ΤΟ AJAX ΜΟΝΤΕΛΟ ΕΦΑΡΜΟΓΗΣ ΙΣΤΟΥ .....	25
ΕΙΚΟΝΑ 3.2.3 ΣΥΓΧΡΟΝΕΣ ΑΙΤΗΣΕΙΣ ΚΑΙ ΑΠΟΚΡΙΣΕΙΣ.....	26
ΕΙΚΟΝΑ 3.2.4 ΑΣΥΓΧΡΟΝΗ ΕΠΙΚΟΙΝΩΝΙΑ ΚΑΙ ΑΝΑΝΕΩΣΗ ΣΕΛΙΔΑΣ.....	27
ΕΙΚΟΝΑ 3.2.5 ΟΙ ΠΙΟ ΔΗΜΟΦΙΛΕΙΣ ΕΤΙΚΕΤΕΣ ΣΤΟ FLICKR.....	28
ΕΙΚΟΝΑ 3.2.6 ΟΙ ΤΡΕΙΣ ΔΙΑΣΤΑΣΕΙΣ ΤΟΥ WEB 2.0.....	31
ΠΙΝΑΚΑΣ 3.4.1 ΠΑΡΑΔΕΙΓΜΑ ΕΡΜΗΝΕΙΑΣ .....	35
ΠΙΝΑΚΑΣ 3.4.2 ΣΥΝΑΡΤΗΣΗ ΕΡΜΗΝΕΙΑΣ ΤΗΣ <i>AL</i> .....	35
ΠΙΝΑΚΑΣ 3.4.3 ΣΗΜΑΣΙΟΛΟΓΙΑ ΤΗΣ DL-LITE .....	39
ΠΙΝΑΚΑΣ 3.4.3 ΣΗΜΑΣΙΟΛΟΓΙΑ ΤΗΣ FUZZY-DL-LITE.....	43
ΕΙΚΟΝΑ 4.2.1 ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΣΥΣΤΗΜΑΤΟΣ.....	49
ΕΙΚΟΝΑ 4.2.2 ΛΕΠΤΟΜΕΡΗΣ ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΚΑΙ ΥΠΟΣΥΣΤΗΜΑΤΑ ΤΗΣ ΕΦΑΡΜΟΓΗΣ .....	50
ΕΙΚΟΝΑ 4.3.1 ΜΟΝΤΕΛΟ ΟΝΤΟΤΗΤΩΝ-ΣΥΣΧΕΤΙΣΕΩΝ.....	52
ΕΙΚΟΝΑ 4.4.1 ΣΧΕΣΙΑΚΟ ΔΙΑΓΡΑΜΜΑ .....	54
ΕΙΚΟΝΑ 5.1.1 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΗΣ BIBTEX.....	77
ΕΙΚΟΝΑ 5.1.2 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΗΣ USERBEAN.....	80
ΕΙΚΟΝΑ 5.1.3 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΗΣ COMMENTINFO.....	82
ΕΙΚΟΝΑ 5.1.4 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΗΣ PUBLICATIONBEAN.....	83
ΕΙΚΟΝΑ 5.1.5 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΗΣ MD5UTIL .....	85
ΕΙΚΟΝΑ 5.1.6 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΗΣ RESULTCOMPARATOR .....	85
ΕΙΚΟΝΑ 5.1.7 ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΗΣ DBCONNECTION.....	87
ΕΙΚΟΝΑ 5.3.1 Η ΣΕΛΙΔΑ WELCOME.JSP .....	101
ΕΙΚΟΝΑ 5.3.2 Η ΣΕΛΙΔΑ REGISTRATION.JSP.....	102
ΕΙΚΟΝΑ 5.3.3 Η ΣΕΛΙΔΑ HOME.JSP.....	103
ΕΙΚΟΝΑ 5.3.4 ΤΟ ΜΕΝΟΥ ΕΠΙΛΟΓΩΝ.....	104
ΕΙΚΟΝΑ 5.3.5 Η ΑΡΙΣΤΕΡΗ ΣΤΗΛΗ ΤΗΣ ΚΕΝΤΡΙΚΗΣ ΣΕΛΙΔΑΣ.....	105
ΕΙΚΟΝΑ 5.3.6 Η ΔΕΞΙΑ ΣΤΗΛΗ ΤΗΣ ΚΕΝΤΡΙΚΗΣ ΣΕΛΙΔΑΣ.....	106
ΕΙΚΟΝΑ 5.3.7 ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΚΑΙΝΟΥΡΓΙΑΣ ΔΗΜΟΣΙΕΥΣΗΣ – ΒΗΜΑ 1 .....	107
ΕΙΚΟΝΑ 5.3.8 ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΚΑΙΝΟΥΡΓΙΑΣ ΔΗΜΟΣΙΕΥΣΗΣ – ΒΗΜΑ 1Β .....	107
ΕΙΚΟΝΑ 5.3.9 ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΚΑΙΝΟΥΡΓΙΑΣ ΔΗΜΟΣΙΕΥΣΗΣ – ΒΗΜΑ 2 .....	108
ΕΙΚΟΝΑ 5.3.10 ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΚΑΙΝΟΥΡΓΙΑΣ ΔΗΜΟΣΙΕΥΣΗΣ – ΒΗΜΑ 2Β .....	108
ΕΙΚΟΝΑ 5.3.11 ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΚΑΙΝΟΥΡΓΙΑΣ ΔΗΜΟΣΙΕΥΣΗΣ – ΒΗΜΑ 2Γ.....	109
ΕΙΚΟΝΑ 5.3.12 ΧΑΡΑΚΤΗΡΙΣΜΟΣ ΚΑΙΝΟΥΡΓΙΑΣ ΔΗΜΟΣΙΕΥΣΗΣ – ΒΗΜΑ 3 .....	109

ΕΙΚΟΝΑ 5.3.13 ΑΝΑΖΗΤΗΣΗ ΜΕ ΒΑΣΗ ΤΗΝ ΕΡΕΥΝΗΤΙΚΗ ΠΕΡΙΟΧΗ .....	110
ΕΙΚΟΝΑ 5.3.14 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ ΜΕ ΒΑΣΗ ΤΗΝ ΕΡΕΥΝΗΤΙΚΗ ΠΕΡΙΟΧΗ .....	111
ΕΙΚΟΝΑ 5.3.15 ΕΞΕΡΕΥΝΗΣΗ ΔΗΜΟΣΙΕΥΣΕΩΝ .....	112
ΕΙΚΟΝΑ 5.3.16 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ ΜΕ ΛΕΞΗ ΚΛΕΙΔΙ .....	113
ΕΙΚΟΝΑ 5.3.17 ΕΜΦΑΝΙΣΕΙΣ ΛΕΞΗΣ ΚΛΕΙΔΙ ΣΕ ΚΕΙΜΕΝΟ ΔΗΜΟΣΙΕΥΣΗΣ .....	114
ΕΙΚΟΝΑ 5.3.18 ΕΞΕΡΕΥΝΗΣΗ ΣΥΓΓΡΑΦΕΩΝ .....	115
ΕΙΚΟΝΑ 5.3.19 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ ΣΥΓΓΡΑΦΕΩΝ .....	116
ΕΙΚΟΝΑ 5.3.20 ΕΞΕΡΕΥΝΗΣΗ ΧΡΗΣΤΩΝ ΣΥΣΤΗΜΑΤΟΣ .....	117
ΕΙΚΟΝΑ 5.3.21 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ ΧΡΗΣΤΩΝ .....	118
ΕΙΚΟΝΑ 5.3.22 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΝΑΖΗΤΗΣΗΣ ΜΕ ΒΑΣΗ ΤΗΝ ΕΤΙΚΕΤΑ .....	119
ΕΙΚΟΝΑ 5.3.23 ΠΡΟΦΙΛ ΧΡΗΣΤΗ .....	119

# 1

## *Εισαγωγή*

### *1.1 Οργάνωση και ανάκτηση πληροφορίας σε ακαδημαϊκές κοινότητες*

Στη σύγχρονη εποχή οι ακαδημαϊκές κοινότητες παράγουν ένα σημαντικό όγκο αδόμητης πληροφορίας – προερχόμενη κυρίως από τη συνεργασία και ανταλλαγή γνώσης από τα μέλη της. Ο όγκος της αδόμητης αυτής πληροφορίας δημιουργεί την ανάγκη να υπάρξει ένας τρόπος η ρευστή αυτή πληροφορία να αξιοποιηθεί.

Υπάρχουν διάφορες προσεγγίσεις προς αυτή την κατεύθυνση με πιο σχετική την ανάπτυξη σημασιολογικών μοντέλων απεικόνισης του συγκεκριμένου πεδίου γνώσης μέσω Οντολογιών που αποτελούν το φορμαλιστικό μοντέλο απεικόνισης πληροφορίας.

Για να μπορούν οι χρήστες κάθε κοινότητας να προσπελάσουν, εξερευνήσουν, διαχειριστούν και να ανακαλύψουν την αποθηκευμένη πληροφορία με τρόπο αποδοτικό, χρησιμοποιούνται δομημένα δεδομένα που περιγράφουν το πεδίο γνώσης που απασχολεί κάθε κοινότητα τα οποία ονομάζονται μεταδεδομένα. Τα μεταδεδομένα (metadata) – δεδομένα περί δεδομένων- επιτρέπει σε συστήματα να οργανώσουν την σχετική πληροφορία. Η δημιουργία μεταδεδομένων προσεγγίζεται με δυο τρόπους: δημιουργία από ειδικό (professional creation) και δημιουργία από τον δημιουργό-συγγραφέα (professional creation). Σε βιβλιοθήκες και άλλους οργανισμούς παραδοσιακά η διαδικασία αυτή ανατίθεται σε ειδικούς οι οποίοι δουλεύουν με πολύπλοκους ορισμούς, λεπτομερείς κανόνες και λεξικό όρων. Η προσέγγιση αυτή αποδεικνύεται μη πρακτική όταν επιχειρείται κλιμάκωση λόγω του τεράστιου όγκου δεδομένων που παράγεται και χρησιμοποιείται ειδικά όταν πρόκειται για υπηρεσίες που αφορούν τον παγκόσμιο ιστό (World Wide Web). Τα εργαλεία που χρησιμοποιούνται από τα

συστήματα καταλογοποίησης (cataloging Systems) είναι γενικά πολύ περίπλοκα για κάποιον χωρίς εξειδίκευση και γνώση της περιοχής.

Οι διάφορες online κοινότητες -πέραν της ακαδημαϊκής- προσπαθώντας να λύσουν το πρόβλημα της οργάνωσης, διαμοιρασμού και ανάκτησης πληροφορίας έχουν αυθόρμητα ακολουθήσει λιγότερο φερεταλιστικές μεθόδους στο πνεύμα και τη φιλοσοφία του Web 2.0 όπως είναι πχ. η ετικετοποίηση (tagging), λόγω ακριβώς των εγγενών δυσκολιών στη χρήση σημασιολογικών εργαλείων. Τέτοιες προσπάθειες καταφέρνουν να λύσουν εν μέρει το πρόβλημα ανάκτησης πληροφορίας και περαιτέρω να εκμεταλλευτούν με τρόπο αποδοτικό το συνεργατικό χαρακτήρα του διαδικτύου. Η διαμόρφωση πληροφορίας επιτυγχάνεται από ένα σύνολο ανθρώπων, προσπερνώντας έτσι τις όποιες αδυναμίες που μπορεί να υπάρξουν όταν η πληροφορία καταχωρείται με τρόπο υποκειμενικό από ένα άνθρωπο-ειδικό. Ακόμη λόγω της συμμετοχής από μια μεγάλη κοινότητα χρηστών στη διαδικασία, ο διαμοιρασμός και ανταλλαγή γνώσης γίνεται με τρόπο αποδοτικότερο ενώ μπορεί να δημιουργηθεί χωρίς ιδιαίτερο κόπο μεγαλύτερος όγκος πληροφορίας σε μικρότερο χρονικό διάστημα.

## ***1.2 Αντικείμενο διπλωματικής***

Σκοπός της εργασίας αυτής είναι να δημιουργήσει μια συνεργατική υπηρεσία όπου θα ακολουθηθεί μια άλλη προσέγγιση στο πρόβλημα της δημιουργίας μεταδεδομένων: δημιουργία από τον χρήστη (user-generated metadata) για τους πόρους που χρησιμοποιεί τόσο για δική του προσωπική του χρήση όσο και για χρήση από την κοινότητα.

Συγκεκριμένα υλοποιήθηκε μια υπηρεσία κοινωνικής οργάνωσης αναφορών (social reference management service) η οποία ακολουθεί σύγχρονους κανόνες σχεδίασης όσο αφορά τον κοινωνικό της χαρακτήρα ενώ ταυτόχρονα υποστηρίζει εγγενώς υπηρεσίες συμπερασματολογίας βασισμένη στα πρότυπα του σημασιολογικού ιστού (semantic web). Η ανάγκη που προκύπτει από τη πληθώρα των πληροφοριών μας έσπρωξε στην υιοθέτηση τεχνολογιών αναπαράστασης γνώσης ούτως ώστε να αποδοθεί σημασιολογία στο περιεχόμενο που διαχειριζόμαστε και να γίνει εφικτή η ανάπτυξη ευφυούς συμπεριφοράς του συστήματος με χρήση σύγχρονων μεθόδων αναζήτησης και εξαγωγής συμπερασμάτων. Για το σκοπό αυτό αναπτύχθηκε οντολογία η οποία περιγράφει μαθηματικά αυστηρά και τυπικά το πεδίο γνώσης που εργαστήκαμε. Λόγω του όγκου των μεταδεδομένων που αναμένεται να εισαχθούν στη βάση γνώσης του συστήματος στραφήκαμε σε βατές περιγραφικές λογικές χωρίς ωστόσο να περιοριζόμαστε σε εκφραστικότητα όσο αφορά την αναπαράσταση του πεδίου γνώσης. Επίσης λόγω του ότι αντίστοιχα σημασιολογικά εργαλεία χαρακτηρίζονται από δυσχρηστία καθώς απαιτούν εξειδίκευση από τους χρήστες, καταβλήθηκε προσπάθεια



ένας χρήστης να μπορεί να εισάγει καινούργια (μέτα)δεδομένα στη βάση γνώσης και γενικά να αξιοποιήσει πλήρως τις δυνατότητες του συστήματος με ευκολία και χωρίς κάποια εξειδίκευση.

### ***1.3 Οργάνωση κειμένου***

Στο κεφάλαιο 2 παρουσιάζονται δυο από τις πιο δημοφιλείς υπηρεσίες κοινωνικής οργάνωσης αναφορών (social reference management services) οι οποίες είναι το bibSonomy<sup>1</sup> και το citeULike<sup>2</sup> και παρουσιάζονται οι λόγοι που ώθησαν στην ανάπτυξη της εφαρμογής.

Το κεφάλαιο 3 συνοψίζει το κύριο θεωρητικό υπόβαθρο στο οποίο στηρίχτηκε η ανάπτυξη της εφαρμογής.

Στο κεφάλαιο 4 αναπτύσσονται οι προδιαγραφές του συστήματος ενώ παρουσιάζεται και η αρχιτεκτονική του.

Στο κεφάλαιο 5 αναφέρονται λεπτομέρειες υλοποίησης του συστήματος ενώ παρουσιάζεται η χρήση του συστήματος με την βοήθεια σεναρίων χρήσης.

Στο κεφάλαιο 6 συνοψίζει το περιεχόμενο της εργασίας και προτείνει μελλοντικές επεκτάσεις σε αυτή.

Τέλος στο κεφάλαιο 7 δίνεται η σχετική βιβλιογραφία που χρησιμοποιήθηκε για την συγγραφή της παρούσας διπλωματικής εργασίας.

---

<sup>1</sup> <http://www.bibsonomy.org/>

<sup>2</sup> <http://www.citeulike.org/>



# 2

## Σχετικές

### εργασίες

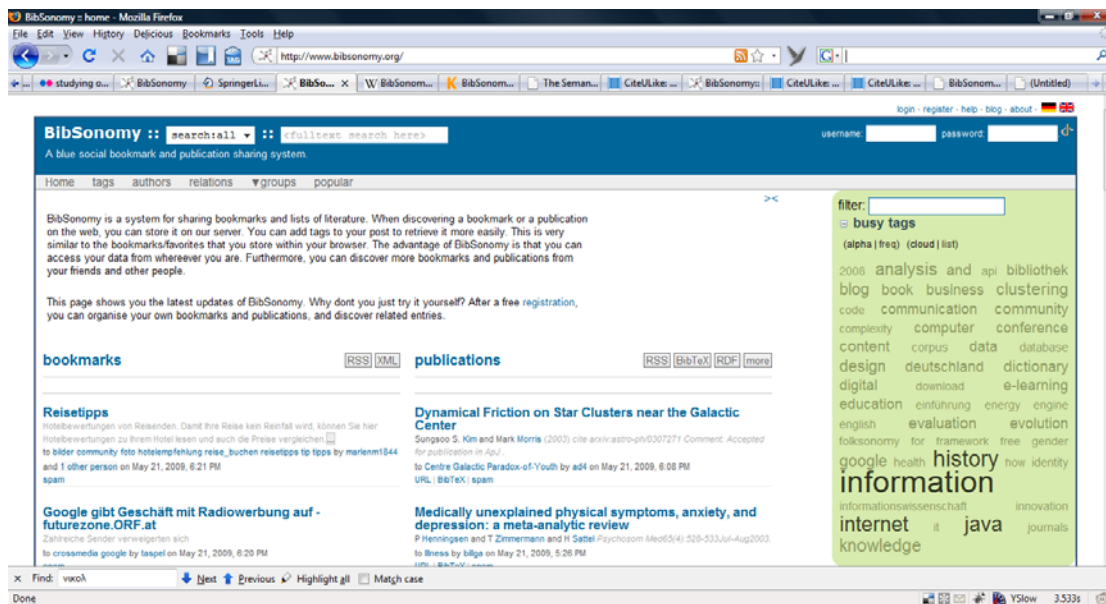
Στο κεφάλαιο αυτό παρουσιάζονται δυο από τις πιο δημοφιλείς υπηρεσίες κοινωνικής οργάνωσης αναφορών (social reference management services) οι οποίες είναι το bibSonomy και το citeULike και παρουσιάζονται οι λόγοι που ώθησαν στην ανάπτυξη της εφαρμογής.

#### **2.1 BibSonomy**

Το bibSonomy<sup>1</sup> πρόκειται για ένα σύστημα οργάνωσης επιστημονικών δημοσιεύσεων και κοινωνικής σελιδοδεικτοποίησης (social bookmarking). Προσφέρει την δυνατότητα στους χρήστες του να αποθηκεύσουν και να οργανώσουν τις δημοσιεύσεις τους και να τις μοιραστούν με άλλους χρήστες. Πέραν των δημοσιεύσεων οι χρήστες μπορούν να αποθηκεύσουν και τους σελιδοδείκτες τους (bookmarks).

---

<sup>1</sup> <http://www.bibsonomy.org/>

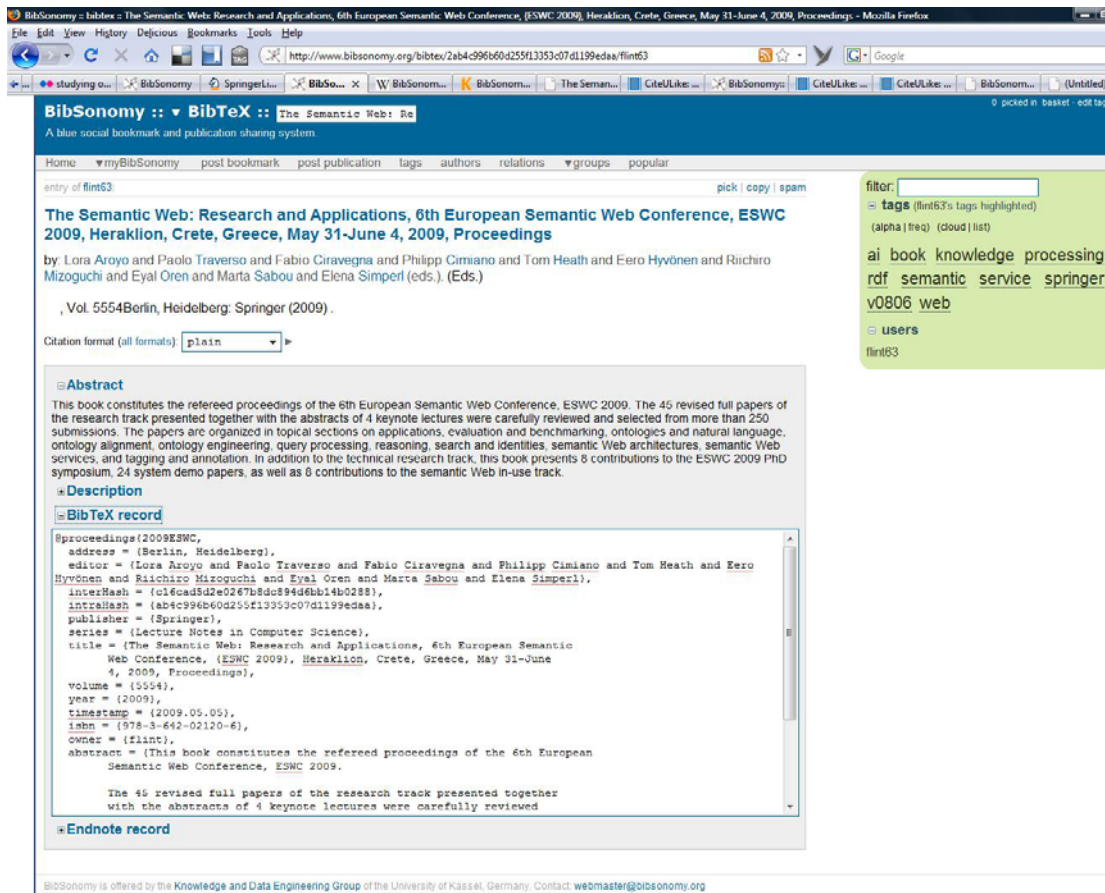


**Εικόνα 2.1.1 Η κεντρική σελίδα του bibSonomy**

Το σύστημα αναπτύχθηκε και συντηρείται από το Institute of Knowledge and Data Engineering<sup>1</sup> του πανεπιστημίου του Kassel<sup>2</sup>. Επιτρέπει, με φιλοσοφία Web 2.0, την αποθήκευση των μεταδεδομένων των επιστημονικών δημοσιεύσεων (ή σελιδοδεικτών) που οι χρήστες πιστεύουν ότι είναι ενδιαφέρον με σκοπό να τις μοιραστούν με τους υπόλοιπους χρήστες. Στις δημοσιεύσεις ( και σελιδοδείκτες) μπορούν να προστεθούν ετικέτες (tags) από διάφορους χρήστες κάτι που οδηγεί αυθόρμητα στην δημιουργία ενός μη ελεγχόμενου λεξικού όρων – folksonomy. Ως κύριο πρότυπο μεταδεδομένων που περιγράφει κάθε δημοσίευση χρησιμοποιείται το bibTex ωστόσο υπάρχει πρόνοια εξαγωγής των μεταδεδομένων και σε άλλα σχετικά πρότυπα όπως είναι το EndNote(RIS), Harvard notation, JabRef ή ακόμα και σε απλή Html και απλό κείμενο. Πέραν των προτύπων αυτών υπάρχει η δυνατότητα να γίνει εξαγωγή δημοσιεύσεων σε αρχείο RDF σύμφωνα με την οντολογία SWRC και να διαμοιραστούν τα δεδομένα σύμφωνα με το γενικό πρότυπο RSS (Really Simple Syndication) ή με το ειδικευμένο για δημοσιεύσεις πρότυπο BuRST.

<sup>1</sup> <http://www.kde.cs.uni-kassel.de/>

<sup>2</sup> <http://cms.uni-kassel.de/unicms/>



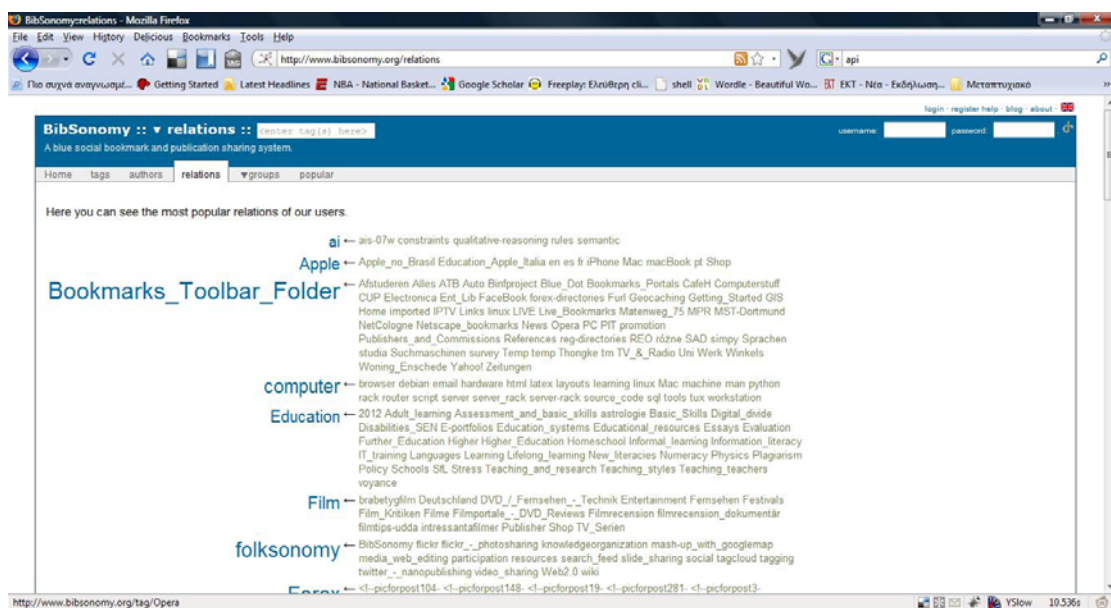
### Εικόνα 2.1.2 Η παρουσίαση μιας δημοσίευσης στο bibSonomy

Στην εφαρμογή υπάρχει υλοποιημένη διεπαφή προγραμματισμού εφαρμογών (API – Application Programming Interface) η οποία επιτρέπει την πρόσβαση των υπηρεσιών της εφαρμογής από τρίτες εφαρμογές (χωρίς την χρήση περιηγητή – browser) και ανάπτυξη εργαλείων που ενσωματώνουν τις υπηρεσίες του bibSonomy. Συγκεκριμένα χρησιμοποιώντας την αρχιτεκτονική REST (Representational state transfer) μπορεί να εκτελέσει μια σειρά από λειτουργίες όπως: ανάκτηση μεταδεδομένων δημοσίευσης (σε ένα από τα προαναφερθέντα πρότυπα), ανάκτηση δημοφιλών ετικετών, ετικετών για μια δημοσίευση, να δημιουργήσει ομάδα, να ανακτήσει τις δημοσιεύσεις μιας ομάδας κ.α. Η ύπαρξη αυτής της διεπαφής έχει οδηγήσει στην δημιουργία υπηρεσιών όπως

Πρόσθετα προγράμματα (plugins) για τον περιηγητή Firefox όπως το GoogleSonomy το οποίο ενσωματώνει αυτόματα αποτελέσματα αναζήτησης από το bibSonomy στο Google ή ενσωμάτωση των υπηρεσιών του bibSonomy στο δημοφιλές εργαλείο διαχείρισης δημοσιεύσεων του firefox Zotero.

Δημιουργία πρόσθετου προγράμματος για την δημοφιλή υπηρεσία blogging Wordpress το οποίο ανανεώνει αυτόματα τα άρθρα του χρήστη στο Wordpress στο bibSonomy ή ενσωματώνει στις ετικέτες του άρθρου αυτού τις ετικέτες που υπάρχουν στο bibSonomy.

Ενδιαφέρον παρουσιάζει η προσέγγιση του bibSonomy όσο αφορά την επέκταση του Folksonomy εισάγωντας την σχέση related μεταξύ των tags. Συγκεκριμένα εαν ένας χρήστης έχει αποδώσει σε μια δημοσίευση τα tags java και programming τότε το tag java και το tag programming σχετίζονται με το ρόλο related. Οι ερευνητές του bibSonomy έχουν αναπτύξει ειδικό αλγόριθμό FolksRank [1] ο οποίος εκμεταλλεύεται τους χαρακτηρισμούς αυτούς και αναδεικνύει την δομή του Folksonomy. Ο αλγόριθμος ο οποίος μοιάζει με τον αλγόριθμο PageRank που χρησιμοποιεί το Google και χρησιμοποιείται για να παρουσιάσει δομημένα αποτελέσματα αναζήτησης, να ανακαλύψει υφιστάμενες κοινότητες στο bibSonomy και να προτείνει ετικέτες (recommended tags) στον χρήστη.



Εικόνα 2.1.3 Δημοφιλείς συσχετίσεις ευικετών στο bibSonomy

## 2.2 CiteULike

Το CiteULike<sup>1</sup> αποτελεί την πρώτη προσπάθεια ανάπτυξης υπηρεσίας κοινωνικής σελιδοδεικτοποίησης (social bookmarking) εξειδικευμένη στις ανάγκες της ακαδημαϊκής κοινότητας και ερευνητών. Η υπηρεσία αναπτύχθηκε αρχικά από το πανεπιστήμιο του Manchester<sup>2</sup> το 2004 ωστόσο αργότερα εξελίχθηκε σε εμπορική εφαρμογή διατηρώντας όμως τον δωρεάν χαρακτήρα της.

<sup>1</sup> <http://www.citeulike.org/>

<sup>2</sup> <http://www.manchester.ac.uk/>

Όπως και το bibSonomy επιτρέπει στους χρήστες του να αποθηκεύσουν αναφορές σε δημοσιεύσεις που εκδηλώνουν ενδιαφέρον για αυτές, να τις οργανώσουν και να τις μοιραστούν με τρόπο αποδοτικό και εύκολο με άλλους συνεργάτες και χρήστες του συστήματος. Συγκεκριμένα οι χρήστες του CiteULike μπορούν να δημιουργήσουν τη δική τους λίστα από αναφορές προσθέτοντας ετικέτες σε αυτές, να συμμετάσχουν σε ομάδα (group) που ασχολείται με συγκεκριμένο θέμα, να παρακολουθήσουν τη δραστηριότητα μιας ερευνητικής ομάδας, να προσθέσουν σχόλια ή κριτικές σε αυτές ή να ανακτήσουν συναφείς δημοσιεύσεις με τις δικές τους. Η διαδικασία παρακολούθησης της δραστηριότητας που δημιουργείται γύρω από συγκεκριμένες δημοσιεύσεις γίνεται είτε με ένταξη της σε λίστα παρακολούθησης (watchlist) είτε λαμβάνοντας τη τροφοδότηση (feed) της με χρήση της τεχνολογίας RSS. Ακόμη μπορούν να εξάγουν τις δημοσιεύσεις τους στα δημοφιλή πρότυπα περιγραφής δημοσιεύσεων bibTex και RIS.

Για σκοπούς γρήγορης εισαγωγής δημοσιεύσεων στο σύστημα έχουν αναπτυχθεί εργαλεία που συνεργάζονται με δημοφιλείς σελίδες αναζήτησης και ανάκτησης δημοσιεύσεων όπως π.χ. Google Scholar, CiteSeerX, PubMed και λαμβάνουν απευθείας τη πληροφορία που περιγράφει μια δημοσίευση που ενδιαφέρει τον χρήστη του συστήματος. Ακόμη εκδότες ακαδημαϊκής βιβλιογραφίας ενθαρρύνουν τους αναγνώστες τους να προσθέσουν την συγκεκριμένη δημοσίευση στο CiteULike προσφέροντας ειδικό σύνδεσμο στην ιστοσελίδα που παρουσιάζει την εκάστοτε δημοσίευση. Η αναζήτηση δημοσιεύσεων γίνεται με βάση τα περιγραφικά χαρακτηριστικά: τίτλος, συγγραφείς, επιστημονικό περιοδικό κτλ. είτε με βάση την ετικέτα που έχει αποσωθεί σε αυτή είτε ακόμη με την ερευνητική ομάδα που

### ***2.3 Συμπεράσματα και αξιολόγηση***

Οι υπηρεσίες του bibSonomy και CiteULike που παρουσιάστηκαν εδώ μαζί με τις άλλες υπηρεσίες κοινωνικής οργάνωσης παραπομπών (online reference management services) συγκροτούν μια αξιόπιστη πηγή αναζήτησης και οργάνωσης παραπομπών (references) επιστημονικών δημοσιεύσεων. Κύριο χαρακτηριστικό τους είναι η εκμετάλλευση σε μεγάλο βαθμό των κοινωνικών χαρακτηριστικών του παγκόσμιου ιστού π.χ. με τη δημιουργία ομάδων (groups) από χρήστες που συμμετέχουν και συνεισφέρουν στο χαρακτηρισμό δημοσιεύσεων μέσω ετικετών (tags). Στο πλαίσιο αυτό έχουν αναπτυχθεί ποικίλα εργαλεία που βοηθούν τους χρήστες να οργανώσουν, να χαρακτηρίσουν αυτόματα, να εξάγουν τα μεταδεδομένα των δημοσιεύσεων που τους ενδιαφέρουν με κύριο σκοπό την ανταλλαγή πληροφορίας και την συνεργασία μεταξύ τους.

Στις υπηρεσίες που περιγράψαμε δεν ακολουθούνται πρακτικές που συναντάμε σε συστήματα γνώσης όπου ενθυλακώνονται προηγμένες υπηρεσίες αναζήτησης και εξαγωγής συμπερασμάτων. Παρόλο που οι υπηρεσίες αυτές δίνουν την δυνατότητα εξαγωγής των

μεταδεδομένων σε διεθνώς καθιερωμένα πρότυπα χαρακτηρισμού δημοσιεύσεων (bibTex,RIS κ.α.), η γνώση δεν έχει αναπαρασταθεί τυπικά με χρήση τεχνολογιών αναπαράστασης γνώσης, ούτε υπάρχει κάποια μηχανή εξαγωγής συμπερασμάτων βασισμένη στο σημασιολογικό περιεχόμενο των πόρων του συστήματος. Αποτέλεσμα αυτού είναι π.χ. η αναζήτηση υφιστάμενων πόρων να περιορίζεται σε λεξικογραφική αναζήτηση στους όρους της δημοσίευσης χωρίς να υπάρχει η δυνατότητα για έξυπνες αναζητήσεις, εξαγωγή αυτόματων συμπερασμάτων κτλ.



# 3

## Θεωρητικό

### υπόβαθρο

#### 3.1 Σημασιολογικό Διαδίκτυο

##### 3.1.1 Ο σημερινός παγκόσμιος ιστός

Ο σημερινός παγκόσμιος ιστός (World Wide Web) έχει αλλάξει τον τρόπο που οι άνθρωποι επικοινωνούν μεταξύ τους όπως επίσης και τον τρόπο που οι επιχειρήσεις λειτουργούν. Είναι ο κεντρικός άξονας μιας εξελισσόμενης επανάστασης οδηγώντας τον κόσμο μας σε μια κοινωνία της πληροφορίας (knowledge society).

Ωστόσο πέραν της τεχνολογικής προόδου που έχει επέλθει το (περισσότερο) περιεχόμενο του σημερινού διαδικτύου είναι κατάλληλο μόνο για ανθρώπινη χρήση. Η πληροφορία που βρίσκουμε στον ιστό παράγεται αυτόματα από βάσεις δεδομένων συνήθως όμως παρουσιάζεται χωρίς την αρχική δομή που συναντούμε στις βάσεις δεδομένων. Οι τυπικές χρήσεις του ιστού από τους χρήστες περιλαμβάνει την αναζήτηση και χρήση της πληροφορίας, επικοινωνία με άλλους χρήστες, επισκόπηση on-line καταλόγων και παραγγελία προϊόντων. Οι δραστηριότητες αυτές βασίζονται σε μεγάλο βαθμό στην χρήση εργαλείων λογισμικού κυρίως μηχανών αναζήτησης που βασίζονται στην αυτόματη εξαγωγή λέξεων κλειδιών – λεξικογραφική ανάλυση. Ωστόσο υπάρχουν προβλήματα που σχετίζονται με την χρήση τους όπως :

- Λίγα ή καθόλου αποτελέσματα: Καμία απάντηση στην αναζήτηση ή ανάκτηση σελίδων που δεν έχουν σχέση με το ερώτημα.
- Χαμηλή ακρίβεια στην ανάκτηση παρόλο που έχουμε πολυάριθμα αποτελέσματα :οι πιο σχετικές σελίδες ανακτούνται το γεγονός όμως ότι μαζί με αυτές ανακτούνται και

άλλες πολλές με χαμηλή συνάφεια ή καθόλου συνάφεια είναι το ίδιο κακό με το πρώτο.

- Αποτελέσματα που είναι ευαίσθητα στην ορολογία: Συχνά δεν έχουμε αποτελέσματα όταν οι σχετικές σελίδες χρησιμοποιούν διαφορετική ορολογία απ'αυτή που χρησιμοποιείται στο ερώτημα.
- Τα αποτελέσματα είναι ιστοσελίδες: Συχνά η πληροφορία που ψάχνουμε βρίσκεται σε διαφορετικά έγγραφα οπότεν πρέπει να εκτελέσουμε διαφορετικά ερωτήματα και μετά να εξάγουμε χειροκίνητα την επί μέρους πληροφορία από κάθε ένα έγγραφο και να την συγχωνεύσουμε.

Παρά την εξέλιξη των μηχανών αναζήτησης τα τελευταία χρόνια το πρόβλημα στην αναζήτηση και ανάκτηση πληροφορίας παραμένει ουσιαστικά το ίδιο: ακόμα κι αν μια αναζήτηση είναι επιτυχής ο χρήστης πρέπει να περιεργαστεί τις επιλεγμένες σελίδες και να εξάγει την πληροφορία μόνος του: μια επίπονη και όχι τόσο εύκολη διαδικασία. Το κύριο εμπόδιο στο να υπάρξει καλύτερη διαλειτουργικότητα του χρήστη στο διαδίκτυο εντοπίζεται στο γεγονός ότι το περιεχόμενο του ιστού δεν είναι machine-accessible και machine-prosecable.

### **3.1.2 Το όραμα του σημασιολογικού διαδικτύου**

Το σημασιολογικό διαδίκτυο έχει αναλάβει την πρωτοβουλία να αναπαραστήσει την υπάρχουσα πληροφορία σε μορφή που είναι κατανοητή από υπολογιστές και να χρησιμοποιήσει έξυπνες μεθόδους ούτως ώστε να εκμεταλλευτεί την αναπαράσταση αυτή και να εκτελεί αυτοματοποιημένη εξαγωγή συμπερασμάτων. Ο Tim Berners-Lee (ο εμπνευστής του Semantic Web) αναφέρει χαρακτηριστικά[1]:

“ Το Semantic Web είναι ένα όραμα: η ιδέα να έχουμε δεδομένα στο διαδίκτυο ορισμένα και διασυνδεδεμένα με τέτοιο τρόπο που να μπορούν να χρησιμοποιηθούν από μηχανές όχι μόνο για σκοπούς [οπτικής] παρουσίασης αλλά για αυτοματοποίηση, ενσωμάτωση και επαναχρησιμοποίηση των δεδομένων σε ποικίλες εφαρμογές. [...] Το σημασιολογικό διαδίκτυο είναι μια επέκταση του σημερινού διαδικτύου στο οποίο η πληροφορία έχει καλά δομημένο νόημα, δίνοντας την ικανότητα σε ανθρώπους και μηχανές να συνεργάζονται. ”

Σ'αυτό το πλαίσιο το όραμα του σημασιολογικού διαδικτύου σκοπεύει να αναδιαρθρώσει την λειτουργικότητα του διαδικτύου, μετατρέποντας το από ωκεανό ασύνδετης και αδόμητης πληροφορίας σε μια σημασιολογική βάση γνώσης. Κύριες επιδιώξεις του σημασιολογικού διαδικτύου είναι:

- Οι πράκτορες πληροφορίας (information agents) να παράγουν ιστοσελίδες με σημασιολογικό περιεχόμενο που ενσωματώνουν με τρόπο σαφή την πληροφορία.

- Η πληροφορία στο διαδίκτυο να προσφέρεται με τέτοιο τρόπο ώστε να είναι κατανοητή στον άνθρωπο και οι μηχανές να μπορούν να προσπελάσουν και επεξεργαστούν.
- Η αναζήτηση στο διαδίκτυο να διεξάγεται από πράκτορες λογισμικού (software agents). Οι άνθρωποι θα ερωτούν μηχανές αναζήτησης σε φυσική γλώσσα και θα λαμβάνουν απαντήσεις.
- Τόσο οι πράκτορες αναζήτησης όσο και οι πάροχοι πληροφοριών χρησιμοποιούν κοινές ορολογίες και δομές γνώσης που εμπεριέχονται σε οντολογίες.

### **3.1.3 Η αρχιτεκτονική του σημασιολογικού διαδικτύου**

Οι τεχνολογίες που χρησιμοποιεί το σημασιολογικό διαδίκτυο σχηματίζουν μια στοίβα τεχνολογιών-προτύπων (semantic web stack) όπου κάθε στρώμα σχετίζεται με το επίπεδο αναπαράστασης γνώσης και συλλογιστικής που εμπεριέχει η κάθε τεχνολογία. Το κάθε επίπεδο εξαρτάται από το αμέσως κατώτερο και το επεκτείνει.

Στο κατώτερο επίπεδο βρίσκεται το διεθνώς καθιερωμένο πρότυπο κωδικοποίησης χαρακτήρων **Unicode**. Το Unicode εξυπηρετεί την αναπαράσταση και επεξεργασία κειμένου σε πολλές γλώσσες. Στο ίδιο επίπεδο βρίσκεται το πρότυπο **URI Uniform Resource Identifier** URI το οποίο χρησιμεύει στο να ταυτοποιούνται μοναδικά οι διαδικτυακοί πόροι-οντότητες. Η μοναδική ταυτοποίηση χρειάζεται στα ανώτερα στρώματα για διαχείριση των πόρων.

Στο αμέσως ανώτερο στρώμα συναντούμε την **XML (Extensible Markup Language)**. Η XML είναι η γλώσσα που επιτρέπει στην δημιουργία εγγράφων από δομημένα δεδομένα. Δεν έχει κάποιο δεσμευτικό λεξιλόγιο, επιτρέποντας στον συγγραφέα να δημιουργήσει το δικό του. Επίσης δίνει την δυνατότητα να διαχωριστεί το περιεχόμενο από την παρουσίαση με αποτέλεσμα το περιεχόμενο να μπορεί να χρησιμοποιηθεί-ανταλλαγεί από άλλες εφαρμογές χωρίς να χρειάζεται μετατροπή (information exchange). Η δομή ωστόσο των εγγράφων είναι αυθαίρετη χωρίς να καθορίζεται η σημασιολογία αυτής.

Στο επόμενο στρώμα συναντούμε το πρότυπο **RDF (Resource Description Framework)**. Η γλώσσα RDF χρησιμοποιείται για την απλή περιγραφή πόρων (resources) του διαδικτύου επιτρέποντας τη δημιουργία και την ανταλλαγή μεταδεδομένων των πόρων όπως και οποιαδήποτε άλλα δεδομένα στον Ιστό. Η περιγραφή των πόρων γίνεται με δημιουργία προτάσεων της μορφής υποκείμενο(subject), ιδιότητα(property) και αντικείμενο(object) όπου την θέση του υποκειμένου καταλαμβάνει ο πόρος που περιγράφουμε. Η **RDF-S (RDF Schema)** παρέχει ένα επιπλέον λεξιλόγιο πάνω σ' αυτό της RDF το οποίο περιλαμβάνει

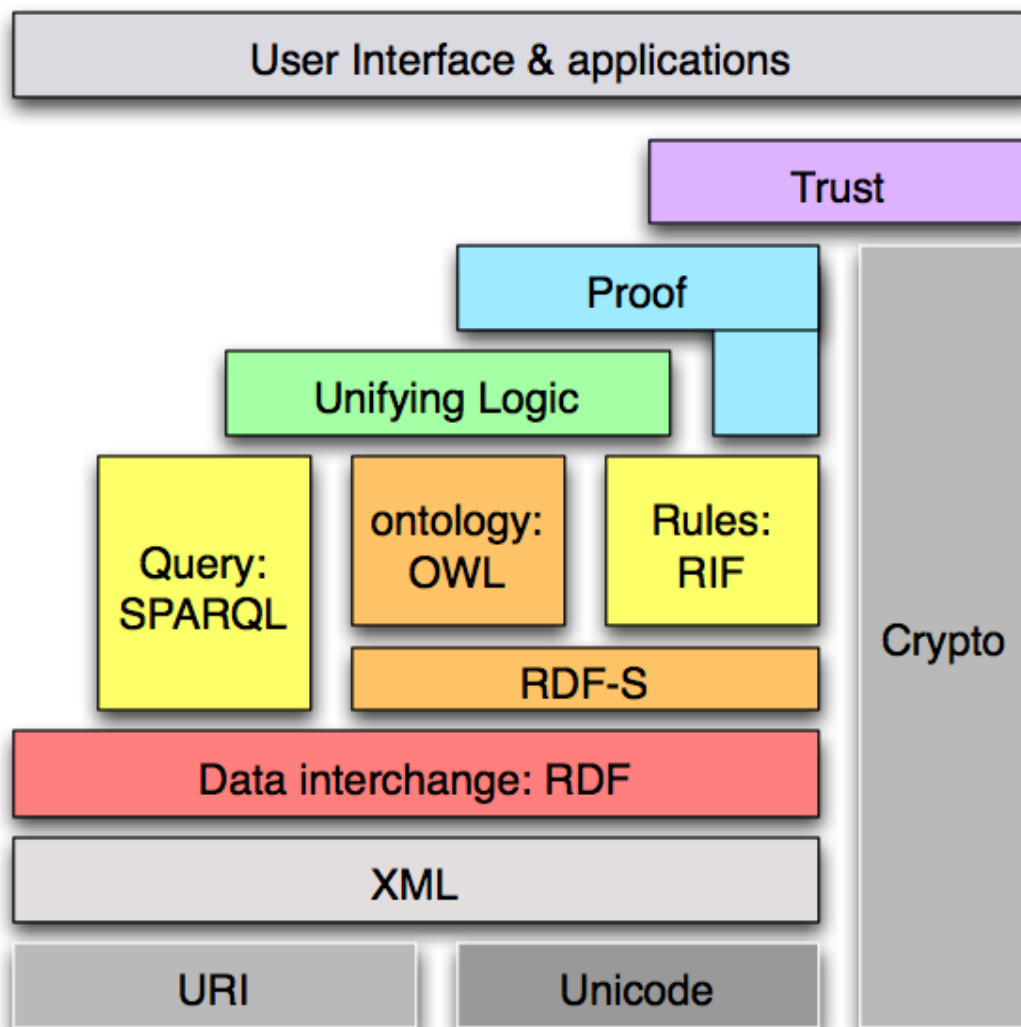
στοιχεία τα οποία προορίζονται στο να προσδώσουν επιπρόσθετες λειτουργικότητες όπως δημιουργία ιεραρχίας κλάσεων και περιορισμών στο πεδίο ορισμού και τιμών των ιδιοτήτων.

Στο ανώτερο στρώμα συναντούμε την γλώσσα **OWL (Web Ontology Language)** η οποία επεκτείνει την RDF-S προσθέτοντας περισσότερη σημασιολογία στις προτάσεις RDF. Επιτρέπει την πρόσθεση περισσότερων περιορισμών όπως περιορισμούς πληθικότητας, χαρακτηριστικά στις ιδιότητες, περιορισμούς στις τιμές, εισαγωγή αξιωμάτων κλάσεων κ.α. Είναι βασισμένη στις **Περιγραφικές Λογικές (ΠΛ)** -που αποτελούν τη αυστηρή και φορμαλιστική μέθοδο αναπαράστασης γνώσης (**Knowledge representation**) – εισάγοντας την δυνατότητα για συμπερασμό βασισμένο σε συλλογιστικές μεθόδους . Η OWL είναι η γλώσσα που χρησιμοποιείται στην σύνταξη οντολογιών για τις οποίες θα αναφερθούμε εκτενώς στην ενότητα 3.3. Στο ίδιο επίπεδο συναντούμε την **SPARQL** η οποία είναι η γλώσσα αναζήτησης των RDF δεδομένων – απαραίτητη για την ανάκτηση πληροφορίας που θα χρησιμοποιηθεί στις εφαρμογές του σημασιολογικού διαδικτύου όπως επίσης και τις γλώσσες κανόνων **RIF (Rule Interchange Format)** και **SWRL (Semantic Web Rule Language)** οι οποίες εξυπηρετούν στην περιγραφή σχέσεων που δεν μπορούν να γίνουν με χρήση περιγραφικών λογικών.

Πιο πάνω βρίσκεται το επίπεδο αποδείξεων στο οποίο γίνεται εξαγωγή συμπερασμάτων χρησιμοποιώντας τους ισχυρισμούς-δεδομένα των πιο κάτω επιπέδων όπως επίσης και αναπαράσταση των διαδικασιών απόδειξης.

Σε όλα τα επίπεδα που περιγράφηκαν είναι σημαντική η κρυπτογραφία ως μέσο ασφάλειας και επαλήθευσης ότι τα σημασιολογικά δεδομένα προέρχονται από αξιόπιστη πηγή. Αυτό μπορεί να επιτευχθεί με χρήση ψηφιακών υπογραφών

Στην κορυφή της στοίβας των τεχνολογιών είναι το επίπεδο της αξιοπιστίας (trust layer) το οποίο θα επαληθεύει ότι τα συμπεράσματα προέρχονται από αξιόπιστη πηγή και ότι η παραγόμενη πληροφορία βασίζεται σε τυπική λογική



Εικόνα 3.1.1 Η στοίβα τεχνολογιών του σημασιολογικού ιστού<sup>1</sup>

## 3.2 Web 2.0

Το **Web 2.0** είναι ο όρος που αναφέρεται σε μια παρατηρούμενη εξέλιξη του ιστού σε πλατφόρμα που ευνοεί/βοηθεί την συνεργασία, επικοινωνία, ανταλλαγή πληροφορίας μεταξύ των χρηστών (**information sharing**) και την διαλειτουργικότητα (**interoperability**) [2].

Ο όρος πρώτα εισήχθη από τον Tim O'Reilly σε συνέδριο του Web 2.0 το 2004. Κατά την περίοδο 1995-2001 παρατηρήθηκε τρομακτική άνοδος στις τιμές των μετοχών και χρηματοδότηση σε καινούργιες εταιρείες που ασχολούνταν με το διαδίκτυο. Πολλές εταιρείες προσθέτοντας απλώς το πρόθεμα e- ή την κατάληξη .com στο όνομα τους πολλαπλασίασαν

<sup>1</sup> <http://www.w3.org/DesignIssues/diagrams/sweb-stack/2006a.png>

την τιμή των μετοχών τους και έλαβαν υπέρογκη χρηματοδότηση από εταιρείες επιχειρηματικών κεφαλαίων (venture capitalists). Η άνοδος αυτή δεν αντιπροσώπευε την πραγματική αξία των εταιρειών με αποτέλεσμα στις αρχές του 2001 να παρατηρηθεί σημαντική πτώση και κατάρρευση των εταιρειών αυτών κάτι που ονομάστηκε ως dot-com bubble (η φούσκα του dot-com). Ως συνέπεια στην αγορά υπήρξε σκεπτικισμός κατά πόσο το διαδίκτυο μπορεί να χρησιμοποιηθεί ως επιχειρηματική πλατφόρμα και να οδηγήσει σε ελάχιστες ή καθόλου καινούργιες επενδύσεις στο χώρο αυτό. Ο Tim O Reilly οργάνωσε το 2001 συνέδριο που σκοπός του ήταν η απαρίθμηση των start-up εταιρειών που επιβίωσαν της κρίσης αυτής. Το συνέδριο είχε ως αποτέλεσμα να ανακαλύψει ότι όλες οι εταιρείες που επιβίωσαν της ύφεσης που παρατηρήθηκε εκμεταλλεύονταν αποτελεσματικά τα εγγενή κοινωνικά και τεχνικά χαρακτηριστικά του διαδικτύου και δεν προσπαθούσαν να μεταφυτεύσουν εκεί τις υπάρχουσες λογικές και τα ισχύοντα οικονομικά μοντέλα της πολιτισμικής βιομηχανίας ή των τηλεπικοινωνιών. Αυτό που ευνόησε αυτές τις εταιρείες στην συνέχεια ήταν ότι μπόρεσαν να εκμεταλλευτούν την περίοδο ωρίμανσης της χρήσης του μέσου από το 2004 και μετά, με την μαζική είσοδο εκατομμυρίων χρηστών με συνδέσεις υψηλής ταχύτητας, αλλά και την κατακόρυφη άνοδο των διαφημιστικών δαπανών.

Ως έννοια το Web 2.0 δεν έχει αυστηρό ορισμό, είναι ένα σύνολο από πρακτικές και φιλοσοφίες που σκοπό έχουν να αξιοποιήσουν κάποιες αρχές που καταστούν τον παγκόσμιο ιστό ένα κοινωνικό και διαδραστικό χώρο δικτύωσης [3]. Πρώτη από αυτές τις αρχές είναι να αξιοποιηθεί ο παγκόσμιος ιστός ως πλατφόρμα (web as a platform). Αυτό σημαίνει ότι οι εφαρμογές δεν πρέπει κατ' ανάγκη να υλοποιούνται για προσωπική χρήση τοπικά στον υπολογιστή του κάθε χρήστη, αλλά η αρχιτεκτονική υλοποίησης του να είναι με τέτοιο τρόπο ούτως ώστε ο χρήστης να μπορεί να αξιοποιήσει τις υπηρεσίες της κάθε εφαρμογής συνδεδεμένος μέσω του ιστού χωρίς να είναι αναγκασμένος να εγκαταστήσει ειδικό πρόγραμμα.

Μια άλλη αρχή του Web 2.0 είναι η (προσπάθεια για) τιθάσευση της συνεργατικής γνώσης. Μια υπηρεσία Web 2.0 πρέπει να βασίζεται στην γνώση που προσφέρουν οι χρήστες της παρά να επαφίεται σε άλλους μεθόδους για να εκτελέσει τις λειτουργίες της. Ως επιτυχημένα παραδείγματα αξιοποίησης αυτής της αρχής αναφέρουμε τα εξής:

- Η δημοφιλέστερη μηχανή αναζήτησης (Google) έχει επικρατήσει στην αγορά κυρίως λόγω του αλγόριθμου αναζήτησης που χρησιμοποιεί ο οποίος εκμεταλλεύεται περισσότερο την δομή των υπερσυνδέσμων που βρίσκονται στις ιστοσελίδες παρά τα χαρακτηριστικά του περιεχομένου της ιστοσελίδας.
- Η δημοφιλής εταιρεία ηλεκτρονικής πώλησης βιβλίων Amazon.com επικρατεί στην αγορά όχι γιατί το προϊόν της υπηρεσίας της είναι κάτι πολύ διαφορετικό από τους ανταγωνιστές της αλλά κυρίως γιατί προσφέρει στους χρήστες της την δυνατότητα να

εμπλακούν και να συμμετέχουν στην διαμόρφωση αποτελεσμάτων (π.χ. μέσω αξιολογήσεων – reviews του κάθε βιβλίου).

- Η online εγκυκλοπαίδεια Wikipedia βασίζεται στην αρχή ότι ο καθένας μπορεί να προσθέσει ένα άρθρο και ο καθένας μπορεί να τροποποιήσει ένα υφιστάμενο άρθρο. Έτσι το περιεχόμενο της δημιουργείται δυναμικά από τους χρήστες οδηγώντας την αυτή την στιγμή να είναι η μεγαλύτερη παγκόσμια βάση γνώσης.

Παρακάτω παραθέτουμε μερικά από τα κύρια συστατικά και δημοφιλείς εφαρμογές του Web 2.0:

### **Blogs**

Τα blogs ή ιστολογία είναι ιστοσελίδες οι οποίες ανανεώνονται συχνά από τους δημιουργούς τους και το περιεχόμενο παρουσιάζεται με αντίστροφη χρονολογική σειρά, δηλαδή από το πιο πρόσφατο στα πιο παλιά. Το περιεχόμενο των ιστολογίων ποικίλλει και κυμαίνεται από προσωπικό ημερολόγιο μέχρι επιχειρηματικό μέσο όπου προβάλλονται τα νέα της εταιρείας. Τα θέματα που ασχολούνται μπορεί να είναι σχολιασμός της καθημερινότητας, σχολιασμός της δημόσιας ζωής, πολιτικής, ενημερωτικά θέματα για τεχνολογία, αθλητισμό, χιουμοριστικά. Ουσιαστικά δεν υπάρχει περιορισμός: το θέμα και χρήση του εκάστοτε ιστολογίου επαφίεται στον δημιουργό του.

Η διάδοσή τους οφείλεται κυρίως στο ότι προσφέρουν τη δυνατότητα σε όποιον επιθυμεί να καταθέσει το σχόλιό του πάνω σε αυτό, ανοίγοντας έτσι ένα δημόσιο διαδικτυακό διάλογο με πιθανούς αποδέκτες όλους τους χρήστες. Τον Φεβρουάριο του 2008 μετρήθηκαν 112.000.000<sup>1</sup> blogs (στην Ελλάδα πάνω από 42.000). Λόγω αυτής της δημοτικότητας, της αίσθησης κοινωνικοποίησης μεταξύ των συμμετεχόντων και της επίδρασης που έχουν ακόμη και εκτός διαδικτύου, χαρακτηρίζονται από πολλούς σαν ένα νέο κοινωνικό φαινόμενο.

### **Wikis**

Τα Wikis αποτελούν χαρακτηριστικό παράδειγμα εφαρμογής ιστού όπου εκμεταλλεύεται την συμμετοχή και συνεισφορά των χρηστών της κοινότητας. Ουσιαστικά αποτελούνται από μια ιστοσελίδα (ή ένα σύνολο από ιστοσελίδες) όπου οι χρήστες της μπορούν να προσθέσουν, αφαιρέσουν και γενικά να επέμβουν στο υφιστάμενο περιεχόμενο μερικές φορές χωρίς καν να είναι ανάγκη να εγγραφούν (αν το wiki είναι δημόσιο). Έτσι αποτελούν μια επεξεργάσιμη συλλογή από σελίδες που δεν απαιτεί από τους χρήστες της να γνωρίζουν πως να συντάξουν κώδικα Html, και είναι μια γρήγορη μέθοδος για συνεργατική κοινοποίηση περιεχομένου

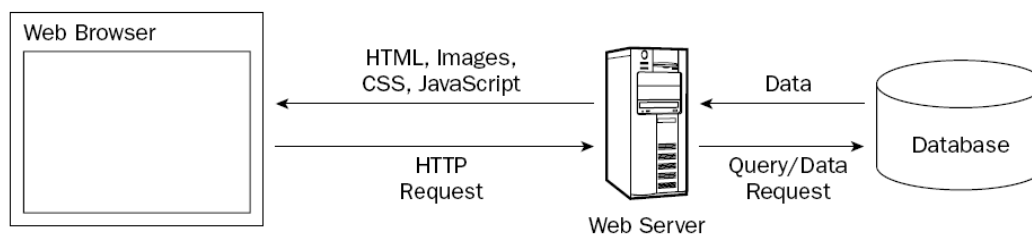
---

<sup>1</sup><http://www.blogherald.com/2008/02/11/how-many-blogs-are-there-is-someone-still-counting/>

στον ιστό. Τα wikis αποτελούν μια μορφή κοινωνικού λογισμικού (social software) καθώς επιτρέπουν την επικοινωνία και συσχέτιση των χρηστών μεταξύ τους είτε ατομικά είτε ως ομάδες και στηρίζει την δημιουργία, συντήρηση και επέκταση κοινωνικών δικτύων (social networks).

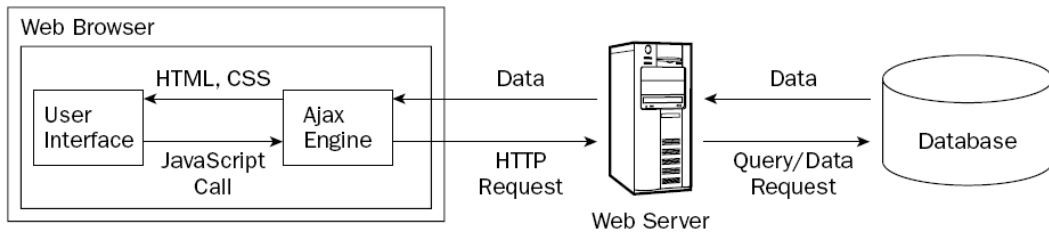
## AJAX

Ο όρος Ajax αναφέρεται σε ένα σύνολο τεχνολογιών ανάπτυξης ιστοσελίδων το οποίο χρησιμοποιείται για δημιουργία διαδραστικών (interactive) σελίδων και «πλούσιων» εφαρμογών διαδικτύου (RIA – Rich Internet Applications). Ο όρος αποτελεί ακρωνύμιο των τεχνολογιών που το απαρτίζουν: **A**synchronous **J**avascript **X**ML ωστόσο πρέπει να αναφέρουμε ότι ουσιαστικά μόνο η ασύγχρονη επικοινωνία είναι απαραίτητη για να χαρακτηριστεί ως Ajax η εφαρμογή ενώ η JavaScript και η XML μπορούν να αντικατασταθούν με άλλες συναφείς τεχνολογίες (π.χ. JSON αντί XML). Η χρήση Ajax επιτρέπει την ασύγχρονη επικοινωνία του πελάτη με τον εξυπηρετητή και ανάκτηση δεδομένων και προβολή του στο χρήστη από αυτό χωρίς την παρεμβολή του εξυπηρετητή. Το Ajax μεταλλάσσει το παραδοσιακό μοντέλο εφαρμογής ιστού όπου ο περιηγητής ήταν υπεύθυνος να εκκινήσει αιτήσεις / να επεξεργαστεί αποκρίσεις από / και προς τον εξυπηρετητή, προσφέροντας ένα ενδιάμεσο στρώμα επικοινωνίας με τον εξυπηρετητή [4]. Το στρώμα αυτό συχνά αποκαλείται μηχανή Ajax και αποτελείται από ένα αντικείμενο JavaScript το οποίο χρησιμοποιείται όποτε χρειάζεται να ανακτηθούν δεδομένα από τον χρήστη. Έτσι αντί κάθε φορά που χρειάζεται να ανακτηθεί πληροφορία από την πλευρά του εξυπηρετητή να απαιτείται να δοθεί ο υπερσύνδεσμος (link) του πόρου που το αφορά και η πληροφορία να επεξεργαστεί για να έρθει σε παρουσιάσιμη μορφή από τον εξυπηρετητή, καλείται το συγκεκριμένο αντικείμενο το οποίο αναλαμβάνει να ανακτήσει τα δεδομένα, να τα επεξεργαστεί και να τα παρουσιάσει στον χρήστη. Η επικοινωνία γίνεται ασύγχρονα εννοώντας ότι η εκτέλεση δεν χρειάζεται να περιμένει απόκριση από την άλλη πλευρά προτού προχωρήσει σε άλλες λειτουργίες.



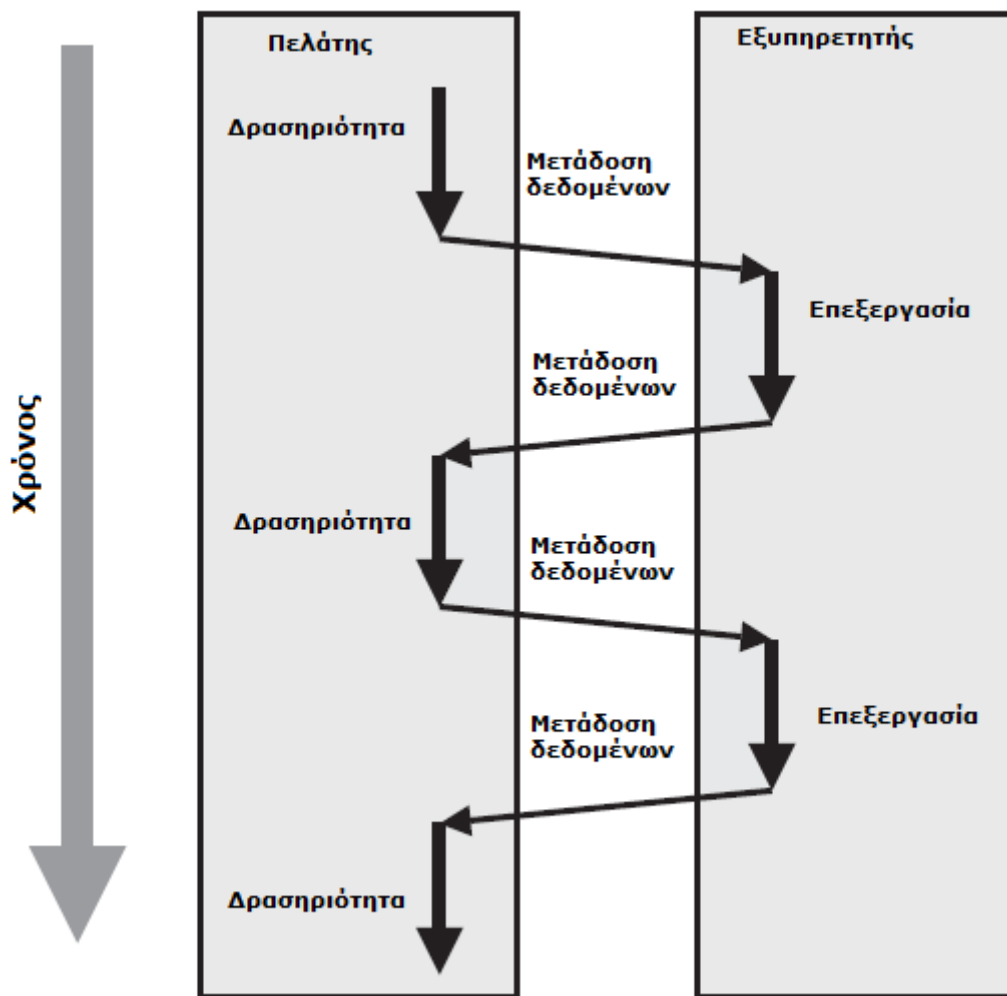
Εικόνα 3.2.1 Το παραδοσιακό μοντέλο εφαρμογής ιστού



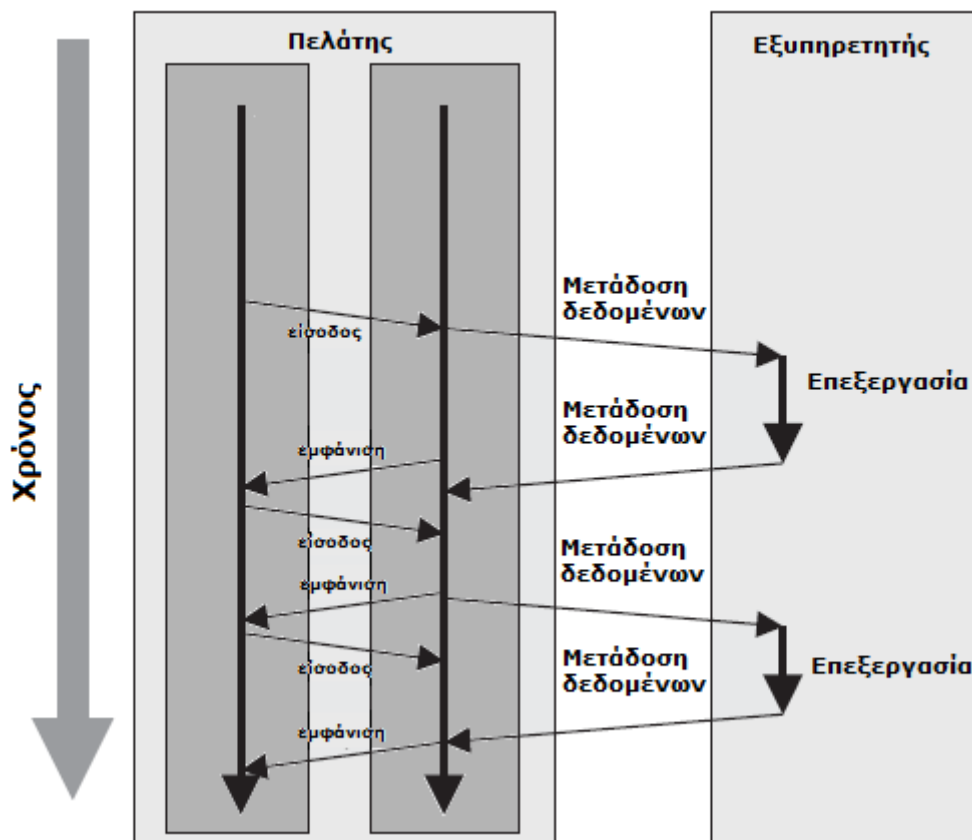


**Εικόνα 3.2.2 Το AJAX μοντέλο εφαρμογής ιστού**

Παρόλο που οι λεπτομέρειες της υλοποίησης ποικίλουν ανάλογα με την εφαρμογή, μια εφαρμογή Ajax αποτελείται από παρόμοια βήματα. Όταν ο χρήστης ενεργοποιεί ένα συγκεκριμένο συμβάν (event) η μηχανή Ajax αναλαμβάνει να το επεξεργαστεί και εκκινάει μια ασύγχρονη επικοινωνία με τον εξυπηρετητή κάνοντας χρήση του αντικειμένου XMLHttpRequest. Παράλληλα ο χρήστης μπορεί να συνεχίσει να εκτελεί άλλες λειτουργίες. Όταν ολοκληρωθεί η ανάκτηση των δεδομένων από τον εξυπηρετητή τα οποία είναι συνήθως σε ημιδομημένη μορφή (π.χ. XML), η μηχανή Ajax τα επεξεργάζεται, προσδίδει σε αυτά στιλ με χρήση γλωσσών φύλλων στιλ (CSS) και τα ενσωματώνει στην τρέχουσα σελίδα.



Εικόνα 3.2.3 Σύγχρονες αιτήσεις και αποκρίσεις



Εικόνα 3.2.4 Ασύγχρονη επικοινωνία και ανανέωση σελίδας

### Tagging

Οι μηχανές αναζήτησης (πριν το google) συχνά επαφίονταν στα μεταδεδομένα που ο δημιουργός της ιστοσελίδας είχε δημιουργήσει για να κατατάξει και κατηγοροποιήσει την σελίδα του. Επειδή όμως παρατηρήθηκε ότι πολλοί δημιουργοί εσκεμμένα πρόσθεταν πολυάριθμα ή μη συναφή μεταδεδομένα με σκοπό να αποκτήσουν υψηλότερη θέση στις μηχανές αναζήτησης, η μέθοδος αυτή απορρίφτηκε. Έκτοτε οι μηχανές αναζήτησης αναζήτησαν άλλους μηχανικούς και αυτοματοποιημένους τρόπους για να παράξουν μετα-πληροφορία, οι οποίες είναι καθόλα επιτυχημένες. Ωστόσο στα πλαίσια του Web 2.0 η ανθρώπινη δημιουργία μεταδεδομένων έχει αναγεννηθεί με την χρήση tagging. Το tagging (ετικετοποίηση) δίνει στους χρήστες την δυνατότητα να προσδώσουν σε ένα πόρο ορισμένο (συχνά απεριόριστο) αριθμό λέξεων κλειδιών-ετικετών που περιγράφουν τον πόρο αυτό. Η διαδικασία αυτή επιτρέπει την εύκολη και απρόσκοπτη αποθήκευση και αναζήτηση πόρων για προσωπική χρήση. Η διαφορά των ετικετών αυτών από τις μέτα-ετικέτες (meta-tags) που μπαίνουν στην κεφαλή (head) μιας σελίδας Html είναι ότι μπορούν να χρησιμοποιηθούν εκτός από το δημιουργό και από κοινότητες χρηστών. Έτσι οι ετικέτες είναι διαθέσιμες και ορατές σε άλλους χρήστες οι οποίοι μπορούν να υιοθετήσουν μερικές από αυτές για δική τους χρήση. Η διαδικασία αυτή δημιουργεί τις επονομαζόμενες folksonomies (συνθετικά folks-

κόσμος και taxonomy- ταξινόμια) [5] και αφορά την συνεργατική (collaborative) δημιουργία και διαχείριση ετικετών από κοινότητα χρηστών με σκοπό τον χαρακτηρισμό και κατηγοριοποίηση περιεχομένου [6, 7]. Πρόκειται για ταξινομίες που δημιουργούνται συνεργατικά από μεγάλο αριθμό χρηστών και κύριο χαρακτηριστικό τους είναι η ευκαμψία καθώς δεν περιορίζεται σε ένα στατικό λεξικό όρων αλλά ο εκάστοτε χρήστης χρησιμοποιεί το δικό του λεξιλόγιο χωρίς να περιορίζεται σε προκαθορισμένο σύνολο όρων.

#### All time most popular tags



**Εικόνα 3.2.5 Οι πιο δημοφιλείς ετικέτες στο flickr**

Ανάλογα με την δυνατότητα να ορίζει ο κάθε χρήστης δικές του ετικέτες διαφορετικές ή ίδιες με αυτές που έχει ορίσει ο δημιουργός για ένα συγκεκριμένο πόρο, οι folksonomies διαχωρίζονται σε ευρείες (broad) και στενές (narrow). Η ευρεία folksonomy είναι το αποτέλεσμα ετικετοποίησης του ίδιου αντικειμένου από μια μεγάλη κοινότητα χρηστών. Παρόλο που ο κάθε χρήστης ακολουθεί την δική του φιλοσοφία στο πως θα απονείμει tags σε συγκεκριμένο πόρο ή δικό του λεξιλόγιο περιγραφής πόρων, συναθροιστικά παρατηρείται η τάση για ένα πόρο να αναδεικνύονται λίγες δημοφιλείς ετικέτες και μεγάλη ποσότητα μη δημοφιλών ετικετών (φαινόμενο Long Tail). Χαρακτηριστικότερο παράδειγμα ευρείας folksonomy αποτελεί η υπηρεσία κοινωνικής σελιδοδεικτοποίησης (social bookmarking) del.icio.us<sup>1</sup> όπου ο ίδιος σελιδοδείκτης (bookmark) μπορεί να ετικετοποιηθεί από πολλούς χρήστες.

---

<sup>1</sup> <http://delicious.com/>

Αντίστοιχα μια στενή folksonomy είναι το αποτέλεσμα ετικετοποίησης ενός πόρου από πολύ λιγότερους χρήστες για προσωπική χρήση. Οι στενές folksonomies μπορεί να μην έχουν τα χαρακτηριστικά μαζικής διαμόρφωσης που έχουν οι ευρείες ωστόσο είναι χρήσιμες σε περιπτώσεις που τα αντικείμενα που ετικετοποιούνται δεν είναι εύκολα αναζητίσιμα με παραδοσιακές μεθόδους ή δεν μπορούν να περιγραφούν με κείμενο. Χαρακτηριστικότερο παράδειγμα στενής folksonomy αποτελεί η υπηρεσία φιλοξενίας φωτογραφιών flickr<sup>1</sup>, η οποία επιτρέπει την ετικετοποίηση φωτογραφιών από το δημιουργό της ή από ορισμένο αριθμό άλλων χρηστών (φίλους του δημιουργού).

### **Υπηρεσίες ιστού (Web services)**

Οι υπηρεσίες ιστού (Web Services) είναι αρχιτεκτονική λογισμικού η οποία είναι με τέτοιο τρόπο σχεδιασμένη ούτως ώστε να προσφέρει διαλειτουργικότητα μεταξύ μηχανών πάνω από το δίκτυο. Ουσιαστικά πρόκειται για λογισμικό το οποίο αναγνωρίζεται μοναδικά από το URI του και οι διεπαφές του μπορούν να κληθούν απομακρυσμένα μέσω ανταλλαγής μηνυμάτων σε XML μορφή. Οι υπηρεσίες ιστού μπορούν να χρησιμοποιηθούν με ποικίλους τρόπους:

Απομακρυσμένη κλήση διαδικασιών (RPC - Remote Procedure Call)

Υλοποίηση υπηρεσιοκεντρικής αρχιτεκτονικής (SOA – Service Oriented Architecture). Η χρήση αρχιτεκτονικής SOA επιτρέπει την ανταλλαγή δεδομένων μεταξύ δυο μηχανών οι οποίες δεν έχουν απαραίτητα τα ίδια χαρακτηριστικά, αρκεί η υποστήριξη του HTTP πρωτοκόλλου.

Υλοποίηση αρχιτεκτονικής REST (RESTful architecture). Η αρχιτεκτονική REST επιτρέπει την εισαγωγή, τροποποίηση, ανάκτηση και διαγραφή πόρων από απομακρυσμένο μηχάνημα με χρήση των αντίστοιχων μηνυμάτων του πρωτοκόλλου HTTP PUT, POST, GET και DELETE.

### **Κοινωνικά Δίκτυα (social networks)**

Σύμφωνα με τον Levene [8], τα κοινωνικά δίκτυα (social networks) φέρνουν μια νέα διάσταση στον παγκόσμια ιστό: πέραν από απλούς συνδέσμους μεταξύ σελίδων, προσθέτουν συνδέσμους μεταξύ χρηστών και κοινοτήτων. Σε αυτά τα δίκτυα άμεσοι σύνδεσμοι δείχνουν σε φίλους και συνεργάτες, έμμεσοι σύνδεσμοι δείχνουν σε φίλους ενός φίλου (friend of a friend) κ.ο.κ.

Τα κοινωνικά δίκτυα επικεντρώνονται στη δημιουργία online κοινοτήτων από άτομα που έχουν κοινά ενδιαφέροντα και ευνοεί την αλληλεπίδραση μεταξύ των χρηστών είτε μέσω μηνυμάτων είτε μέσω εποπτείας των δραστηριοτήτων άλλων χρηστών. Τυπικές δυνατότητες ενός χρήστη σε ένα κοινωνικό δίκτυο είναι η δημιουργία φίλων, η συμμετοχή σε ομάδες

---

<sup>1</sup> <http://www.flickr.com/>

(groups) , η συνάθροιση μιας ομάδας από φίλους με τους οποίους υπάρχει κοινωνικός δεσμός, η δημιουργία και ενημέρωση του προφίλ του, η εξωτερίκευση των φυσικών και εικονικών του δραστηριοτήτων κ.α. Παρόλο που οι δυνατότητες των κοινωνικών δικτύων έχουν εξελιχτεί και διαφοροποιούνται αναλόγως του κοινωνικού δικτύου, η ραχοκοκαλιά της αρχιτεκτονικής παραμένει η δημιουργία δημόσιου προφίλ όπου εμφανίζονται οι κοινωνικές ομάδες στις οποίες ανήκει και οι φίλοι του χρήστη. Αυτό το προφίλ δημιουργείται κατά την εγγραφή του χρήστη στο σύστημα και μπορεί να εμπλουτιστεί αργότερα με φωτογραφίες ή άλλο υλικό. Η προβολή του προφίλ μπορεί να περιοριστεί ανάλογα της επιθυμίας του χρήστη: η ορατότητα του πλήρους προφίλ μπορεί να είναι μόνο για φίλους, για μέλη μιας ομάδας, για φίλους φίλων ή εντελώς δημόσιο για όλους. Η πλειονότητα των κοινωνικών δικτύων επιτρέπει στους χρήστες να αφήσουν μήνυμα σε άλλους χρήστες στο δημόσιο προφίλ τους ή να σχολιάσουν περιεχόμενο που ανέβασε ο χρήστης (φωτογραφίες, βίντεο κτλ).

Η μαζική διείσδυση των κοινωνικών δικτύων στην καθημερινότητα των χρηστών εγείρει θέματα που άπτονται της προστασίας ιδιωτικών δεδομένων των χρηστών. Ενδεικτικά αναφέρουμε την εργασία των Gross and Acquisti [8] όπου ανέλυσαν δημόσια προφίλ στο facebook<sup>1</sup> 4000 φοιτητών του Carnegie Mellon και ανέδειξαν πιθανούς κινδύνους στη ιδιωτική ζωή των χρηστών που προκύπτουν από πληροφορίες που οι ίδιοι οι χρήστες εναπόθεσαν στις προσωπικές τους σελίδες. Τα κοινωνικά δίκτυα εγείρουν και νομικά θέματα καθώς δεν υπάρχει επαρκής και αποσαφηνισμένη νομοθεσία για αυτό το πρώιμο κοινωνικό φαινόμενο.

Συνοψίζοντας αναφέρουμε ότι οι σύγχρονες εφαρμογές ιστού στο χώρο του Web 2.0 επικεντρώνονται σε 3 κύριες διαστάσεις: λειτουργικότητα, κοινωνικότητα και (μεγάλη ποσότητα από) δεδομένα. Ως χαρακτηριστικά παραδείγματα εφαρμογών Web 2.0 αναφέρουμε την εγκυκλοπαίδεια Wikipedia<sup>2</sup> η οποία έχει μεγάλο όγκο δεδομένων και υψηλή λειτουργικότητα ενώ στερείται λίγο ίσως στο κοινωνικό χαρακτήρα, την υπηρεσία δημιουργίας προσωπικού ιστολόγου blogger<sup>3</sup> το οποίο έχει τεράστιο όγκο δεδομένων και υψηλή κοινωνικοποίηση μεταξύ των χρηστών του (ίσως στερείται λίγο σε λειτουργικότητα) και τέλος ως εφαρμογή η οποία έχει και τις τρεις διαστάσεις σε μεγάλο βαθμό την υπηρεσία φιλοξενίας φωτογραφιών flickr<sup>4</sup> (βλ. διάγραμμα 3.2.6).

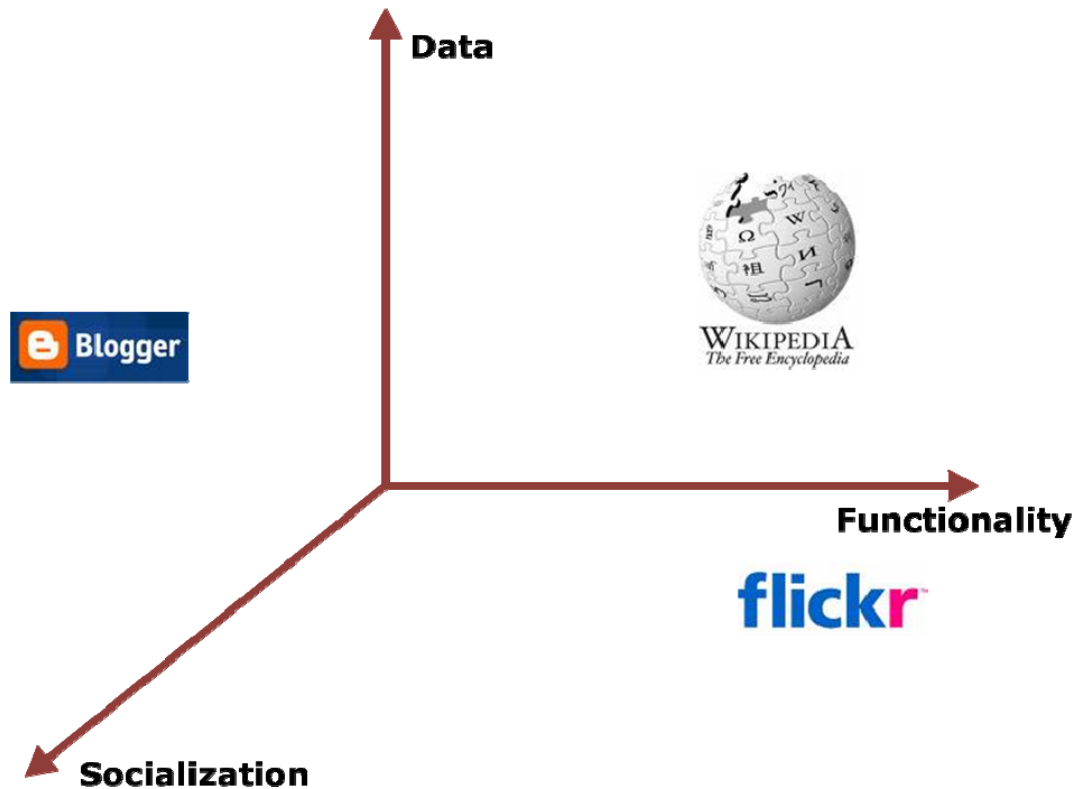
---

<sup>1</sup> <http://www.facebook.com>

<sup>2</sup> <http://www.wikipedia.org/>

<sup>3</sup> <http://www.blogger.com/>

<sup>4</sup> <http://www.flickr.com/>



Εικόνα 3.2.6 Οι τρεις διαστάσεις του Web 2.0

Είναι σημαντικό να αναφερθεί ότι παρόλο που ο όρος “Web 2.0” υπαινίσσεται ότι αναφέρεται σε μια καινούργια έκδοση του διαδικτύου, με καινούργιες ή αναβαθμισμένες τεχνολογίες στην πραγματικότητα αναφέρεται σε μια καινούργια προσέγγιση στην χρήση του ήδη υπάρχον διαδικτύου από τους τελικούς χρήστες, τους κατασκευαστές λογισμικού και τις επιχειρήσεις. Το γεγονός αυτό συμβάλει στο να υπάρχει κριτική στη χρήση του όρου από σημαντικούς ανθρώπους του διαδικτύου με χαρακτηριστικότερο παράδειγμα τον sir Tim Berners Lee (τον δημιουργό του παγκόσμιου ιστού και εισηγητή του σημασιολογικού διαδικτύου) ο οποίος αναφέρει [19]:

« Το Web 1.0 ήταν για να συνδέσει τον κόσμο. Ήταν ένας χώρος διαδραστικός και νομίζω ότι το Web 2.0 είναι ένας τεχνικός όρος, κανείς δεν ξέρει τι σημαίνει ακριβώς. Αν το Web 2.0 για σας είναι τα blogs και τα wikis τότε αυτό είναι: από ανθρώπους για ανθρώπους. Αλλά αυτό υποτίθεται ότι είναι το Web όλα αυτά τα χρόνια έτσι κι αλλιώς.»

### 3.3 Οντολογίες

Η λέξη "οντολογία" είναι ένας όρος δανεισμένος από τη φιλοσοφία, η οποία αναφέρεται στην επιστήμη της περιγραφής των διαφόρων ειδών οντοτήτων στον κόσμο και του τρόπου που

αυτές συσχετίζονται. Στην περιοχή της επιστήμης των υπολογιστών μια οντολογία ορίζει τις αρχές αναπαράστασης με τις οποίες μοντελοποιούμε ένα πεδίο γνώσης. Οι αρχές αναπαράστασης τυπικά είναι οι κλάσεις (classes) ή σύνολα, ιδιότητες (properties) και σχέσεις ή ρόλοι (μεταξύ των μελών κλάσεων). Οι ορισμοί των αρχών αναπαράστασης περιλαμβάνουν πληροφορία για το νόημα και λογικούς περιορισμούς για το πεδίο που αναπαριστούν.

Στο πλαίσιο εφαρμογής των βάσεων δεδομένων, οντολογία μπορεί να θεωρηθεί ως ένα επίπεδο αφαίρεσης των μοντέλων δεδομένων (data models), ανάλογο με τα ιεραρχικά ή σχεσιακά μοντέλα, που αποσκοπεί όμως να μοντελοποιήσει γνώση για άτομα, τις ιδιότητες τους και τις σχέσεις τους με άλλα άτομα. Οι οντολογίες συνήθως ορίζονται σε γλώσσες που επιτρέπουν αφαιρετική σχεδίαση χωρίς τους περιορισμούς που επηρεάζονται από δομές δεδομένων ή λεπτομέρειες υλοποίησης. Έτσι οι οντολογίες προσφέρουν το σημασιολογικό επίπεδο (ενώ αντιστοίχως τα σχεσιακά σχήματα των βάσεων δεδομένων αφορούν το φυσικό επίπεδο). Οι γλώσσες αυτές ονομάζονται γλώσσες αναπαράστασης γνώσης (knowledge representation languages).

Ένας αυστηρός ορισμός της οντολογίας (προτεινόμενος από τον Thomas R. Gruber [9] ) αναφέρει ότι οντολογία είναι ο ρητός (σαφής) προσδιορισμός της αντίληψης για έννοιες (explicit specification of a conceptualization). Βασιζόμενοι σ' αυτό τον ορισμό μπορούμε να διατυπώσουμε ότι οντολογία είναι ένα σύνολο ορισμών προερχόμενων από ένα αυστηρά διατυπωμένο λεξιλόγιο που μας επιτρέπει να αναπαραστήσουμε ένα πεδίο γνώσης.

Συνεχίζοντας παρουσιάζουμε τα βασικά στοιχεία των οντολογιών που είναι:

- **Άτομα** (individuals) τα οποία μπορεί να αντιστοιχούν σε πραγματικά υλικά αντικείμενα (π.χ. πρόσωπα, κτίρια, μηχανές) ή ακόμα και σε αφηρημένα αντικείμενα όπως (π.χ. ιδεολογίες, είδη μουσικής, ονόματα)
- **Κλάσεις εννοιών** (concepts) συλλογή αντικειμένων με τα ίδια χαρακτηριστικά τα οποία θεωρούνται ότι ανήκουν στην ίδια κατηγορία. Μπορούμε να ορίσουμε ιεραρχία εννοιών να ορίσουμε δηλαδή ότι η έννοια A είναι υποκλάση της έννοιας B (και αντίστοιχα η B είναι υπερκλάση της έννοιας A). Έτσι ορίζουμε ότι όλα τα άτομα της έννοιας A ανήκουν στην έννοια B, επεκτείνοντας όμως την έννοια A προσθέτοντας πρόσθετα χαρακτηριστικά στα άτομα αυτά (εξειδίκευση).
- **Συσχετίσεις** (relations) ορίζουν συσχετίσεις μεταξύ δυο εννοιών. Τέτοιοι ρόλοι μπορούν να ορίσουν τη τοποθέτηση ενός ατόμου στο κόσμο σε σχέση με μια άλλη έννοια. Οι ρόλοι δίνουν τη δυνατότητα να οριστούν μοντέλα αναπαράστασης με πιο πλούσιο σημασιολογικό περιεχόμενο.

Οι οντολογίες αναμένεται να διαδραματίσουν σημαντικό ρόλο στο όραμα του σημασιολογικού ιστού. Οι τομείς γνώσης θα μοντελοποιηθούν -με χρήση των Οντολογιών-



με τρόπο δομημένο και τυπικό σχηματίζοντας ένα κοινό λεξιλόγιο που θα μπορεί να χρησιμοποιηθεί από τα μέλη της εκάστοτε κοινότητας για περιγραφή των πόρων χωρίς να υπάρχει ο κίνδυνος για απώλεια πληροφορίας ή δημιουργία αμφισημιών και παρεξηγήσεων. Έτσι οι διαδικτυακές εφαρμογές θα μπορούν να συλλέξουν και να επεξεργαστούν περιεχόμενο βάσει του νοήματος του (και όχι απλώς με τη σύνταξη τους) καθιστώντας εφικτή την περαιτέρω ανάπτυξη σημασιολογικών υπηρεσιών όπου θα υιοθετούνται εξελιγμένες τεχνικές για αναζήτηση και ανάκτηση πληροφορίας.

### **3.4 Περιγραφικές λογικές**

#### **3.4.1 Γενικά**

Οι Περιγραφικές Λογικές είναι γλώσσες αναπαράστασης γνώσης που μπορούν να χρησιμοποιηθούν για να περιγράψουν την ορολογική γνώση ενός πεδίου εφαρμογής με δομημένο, κατανοητό και τυπικό τρόπο. Οι Περιγραφικές Λογικές αποτελούν απόγονο των λεγόμενων Δικτύων Δομημένης Κληρονομικότητας, τα οποία εισήχθησαν για να ξεπεραστούν τα διάφορα προβλήματα που είχαν προκύψει από τα πρώτα σημασιολογικά δίκτυα (semantic networks). Πέραν της περιγραφικής ικανότητας επεκτείνουν τους προαναφερθέντες μηχανισμούς αναπαράστασης γνώσης εισάγοντας μηχανισμούς εξαγωγής συμπερασμάτων (reasoning algorithms). [9],[10]

Δομικά στοιχεία του αλφάβητου (alphabet) των Περιγραφικών Λογικών είναι οι ατομικές έννοιες (atomic concepts), οι ατομικοί ρόλοι (atomic roles) και τα άτομα (individuals).

Η ατομική έννοια είναι ένα κατηγορημα πρώτου βαθμού που αποδίδει ιδιότητα σε άτομα (π.χ. Άνθρωπος , Θηλαστικό). Ο ατομικός ρόλος είναι κατηγορημα δεύτερου βαθμού που δηλώνει συσχέτιση μεταξύ δυο ατόμων (π.χ. έχειΠαιδί, είναιΑπόγονος) ενώ ως άτομο ορίζεται μια οντότητα που θέλουμε να εισάξουμε στην γνώση που περιγράφουμε π.χ. (το πρόσωπο) Λουκάς, (το URL) [Http://www.google.com](http://www.google.com), (η διεύθυνση ) Ακαδημίας 5. Κατά σύμβαση τα άτομα και ατομικές έννοιες ξεκινούν με κεφαλαίο γράμμα ενώ οι ατομικοί ρόλοι με μικρό.

Οι περιγραφικές λογικές συνθέτουν μια οικογένεια διαφορετικών λογικών σύμφωνα με το σύνολο κατασκευαστών (constructors) που διαθέτει η εκάστοτε γλώσσα ΠΛ. Οι κατασκευαστές είναι η μέθοδος η οποία χρησιμοποιείται για την σύνθεση πιο περίπλοκων εννοιών και οι οποίοι επενεργούν πάνω στις βασικές έννοιες και ρόλους.

Σύμφωνα με την σύμβαση που εισηγήθηκε από τους Schmidt-Schauss and Smolka [11], ο κάθε κατασκευαστής συνδέεται με ένα κεφαλαίο γράμμα (π.χ.  $\mathcal{U}$  για τον κατασκευαστή της ένωσης,  $\mathcal{C}$  για τον κατασκευαστή της άρνησης) και η ονομασία της γλώσσας προκύπτει από το πρόθεμα  $\mathcal{AL}$  (Attributive Language - που αποτελεί την βασική γλώσσα ΠΛ) και τα γράμματα των κατασκευαστών που διαθέτει. (π.χ.  $\mathcal{ALU}$ ,  $\mathcal{ALC}$ ). Η παρουσίαση της χρήσης των κατασκευαστών ως μεθόδους για δημιουργία πιο σύνθετων εννοιών θα γίνει με την παρουσίαση των πιο βασικών γλωσσών ΠΛ.

Η πιο βασική γλώσσα ΠΛ είναι η γλώσσα  $\mathcal{AL}$  (attributive language) [11]. Οι περιγραφές εννοιών στην  $\mathcal{AL}$  ορίζονται επαγωγικά με την ακόλουθη αφηρημένη σύνταξη ( $A$  είναι ατομική έννοια,  $C, D$  δυο περιγραφές εννοιών και  $R$  ατομικός ρόλος):

$$C, D \leftarrow A \mid \top \mid \perp \mid \neg A \mid C \sqcap D \mid \forall R.C \mid \exists R. \top$$

Οι έννοιες  $\top$  και  $\perp$  ονομάζονται καθολική έννοια (universal concept ή top concept) και κενή έννοια (bottom concept) αντίστοιχα. Η έννοια  $\forall R.C$  ονομάζεται περιορισμός τιμής (value restriction) ή αλλιώς καθολικός περιορισμός (universal restriction) και η έννοια  $\exists R. \top$  περιορισμένος υπαρξιακός περιορισμός (limited existential restriction).

Στο σημείο αυτό ορίζουμε την σημασιολογία (semantics) της κάθε γλώσσας περιγραφικής λογικής. Η σημασιολογία στις ΠΛ ορίζεται με την εισαγωγή ερμηνείας (interpretation) που αποδίδει σε κάθε έννοια νόημα και σημασία. Μαθηματικά η ερμηνεία  $\mathcal{I}$  ορίζεται από ένα ζεύγος  $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ , όπου  $\Delta^{\mathcal{I}}$  είναι ένα μη-κενό σύνολο που ονομάζεται χώρος ερμηνείας (domain of interpretation) και περιέχει στοιχεία που ονομάζονται αντικείμενα (objects), και  $\cdot^{\mathcal{I}}$  είναι μια συνάρτηση ερμηνείας (interpretation function) που ερμηνεύει κάθε ατομική έννοια  $A$  ως ένα υποσύνολο  $A^{\mathcal{I}}$  του  $\Delta^{\mathcal{I}}$  ( $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ ) και κάθε ρόλο  $R$  ως ένα υποσύνολο  $R^{\mathcal{I}}$  του  $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$  ( $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$ ).

Για παράδειγμα ας θεωρήσουμε ως χώρο ερμηνείας  $\Delta^{\mathcal{I}}$  το (πεπερασμένο) σύνολο αριθμών

$$\Delta^{\mathcal{I}} = \{1, \sqrt{3}, 2, \frac{5}{2}, e, 3, \pi, 4\}. \text{ Η ερμηνεία των εννοιών Ρητός, Άρρητος, Ακέραιος και}$$

των ρόλων μεγαλύτερος και είναι Ρίζα θα είναι:

$$\top^{\mathcal{I}} = \{1, \sqrt{3}, 2, \frac{5}{2}, e, 3, \pi, 4\}$$

$$\perp^{\mathcal{I}} = \emptyset$$

$$\text{Ρητός}^{\mathcal{I}} = \{1, 2, \frac{5}{2}, 3, 4\}$$

$$\text{Άρρητος}^{\mathcal{I}} = \{\sqrt{3}, e, \pi\}$$

$$\text{Ακέραιος}^{\mathcal{I}} = \{1, 2, 3, 4\}$$

$$\text{μεγαλύτερος}^{\mathcal{I}} = \{(\sqrt{3}, 1), (2, \sqrt{3}), (\frac{5}{2}, 2), (e, \frac{5}{2}), (3, e), (\pi, 3), (4, \pi)\}$$

$$\text{είναι Ρίζα}^{\mathcal{I}} = \{(1, 1), (\sqrt{3}, 3), (2, 4)\}$$

#### Πίνακας 3.4.1 Παράδειγμα ερμηνείας

Συγκεκριμένα για την  $\mathcal{AL}$  η συνάρτηση ερμηνείας είναι :

$$\top^{\mathcal{I}} = \Delta^{\mathcal{I}}$$

$$\perp^{\mathcal{I}} = \emptyset$$

$$(\neg A)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$$

$$(C \sqcap D)^{\mathcal{I}} = C^{\mathcal{I}} \cap D^{\mathcal{I}}$$

$$(\forall R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \forall b \in \Delta^{\mathcal{I}} . (a, b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$$

$$(\exists R. \top)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b \in \Delta^{\mathcal{I}} . (a, b) \in R^{\mathcal{I}}\}$$

#### Πίνακας 3.4.2 Συνάρτηση ερμηνείας της $\mathcal{AL}$

Η ερμηνεία της έννοιας  $\forall R.C$  είναι το σύνολο όλων των αντικειμένων του  $\Delta^{\mathcal{I}}$  τα οποία αν συμμετέχουν στην ερμηνεία του ρόλου  $R$  ( $R^{\mathcal{I}}$ ) μαζί με κάποιο άλλο αντικείμενο τότε αυτό το άλλο αντικείμενο ανήκει στην ερμηνεία της έννοιας  $C$  ( $C^{\mathcal{I}}$ ). Η ερμηνεία της έννοιας  $\exists R. \top$  είναι το σύνολο όλων των αντικειμένων του  $\Delta^{\mathcal{I}}$  τα οποία συμμετέχουν στην ερμηνεία του ρόλου  $R$  ( $R^{\mathcal{I}}$ ) μαζί με κάποιο άλλο αντικείμενο που ανήκει στον χώρο ερμηνείας  $\Delta^{\mathcal{I}}$ .

Μια πρώτη επέκταση στην  $\mathcal{AL}$  αποτελεί η δυνατότητα να υπάρξει άρνηση όχι μόνο σε ατομικές έννοιες αλλά και σε περίπλοκες έννοιες. Ο κατασκευαστής αυτός συμβολίζεται με το γράμμα  $\mathcal{C}$  και μαζί με τους κατασκευαστές της γλώσσας  $\mathcal{AL}$

οριοθετεί την γλώσσα  $\mathcal{ALC}$ . Τυπικά η συνάρτηση ερμηνείας του κατασκευαστή άρνησης είναι:

$$(\neg C)^{\mathcal{I}} = \Delta^{\mathcal{I}} \setminus C^{\mathcal{I}}.$$

Με την εισαγωγή του κατασκευαστή αυτού μπορούμε πλέον να περιγράψουμε έννοιες όπως π.χ. του άγαμου γράφωντας  $\text{Άνθρωπος} \sqcap \neg(\exists \text{έχειΣύζυγο. T})$  ή της μητέρας γράφωντας  $\text{Θηλυκό} \sqcap \exists \text{έχειΠαιδί. T}$

Ο κατασκευαστής της ένωσης, ο οποίος συμβολίζεται με το γράμμα  $\mathcal{U}$  επιτρέπει τον ορισμό μιας έννοιας ως ένωση δυο άλλων εννοιών π.χ η έννοια του γονιού μπορεί να αποδοθεί ως  $\text{Πατέρας} \sqcup \text{Μητέρα}$ . Η ερμηνεία της ένωσης δυο εννοιών σύμφωνα με την συνάρτηση ερμηνείας  $\cdot^{\mathcal{I}}$  είναι η ακόλουθη:

$$(C \sqcup D)^{\mathcal{I}} = C^{\mathcal{I}} \cup D^{\mathcal{I}}$$

Η χρήση των κατασκευαστών της  $\mathcal{AL}$  και του κατασκευαστή της ένωσης οριοθετεί την γλώσσα  $\mathcal{ALU}$ .

Ακολούθως παρουσιάζουμε τον πλήρη υπαρξιακό περιορισμό ο οποίος συμβολίζεται με το γράμμα  $\mathcal{E}$ . Η χρήση αυτού του κατασκευαστή επιτρέπει να περιγραφούν έννοιες όπως αυτή των ανθρώπων που έχουν τουλάχιστον μια κόρη με την σύνταξη  $\text{Άνθρωπος} \sqcap \exists \text{έχειΠαιδί.Θηλυκό}$ . Η έννοια αυτή διαφέρει από την έννοια  $\text{Άνθρωπος} \sqcap \forall \text{έχειΠαιδί.Θηλυκό}$  η οποία περιγράφει τους ανθρώπους όπου όλα τους τα παιδιά είναι κόρες (θηλυκά). Η ερμηνεία του περιορισμού αυτού σύμφωνα με την συνάρτηση ερμηνείας  $\cdot^{\mathcal{I}}$  είναι

$$(\exists R.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \exists b \in \Delta^{\mathcal{I}} . (a, b) \in R^{\mathcal{I}} \text{ και } b \in C^{\mathcal{I}}\}$$

Η χρήση των κατασκευαστών της  $\mathcal{AL}$  και του πλήρη υπαρξιακού περιορισμού οριοθετεί την γλώσσα  $\mathcal{ALE}$ .

Ένας άλλος κατασκευαστής είναι ο κατασκευαστικός περιορισμός πληθυκότητας (cardinality restriction) ο οποίος συμβολίζεται με το γράμμα  $\mathcal{N}$  και μαζί με τους κατασκευαστές της γλώσσας  $\mathcal{AL}$  οριοθετεί την γλώσσα  $\mathcal{ALN}$ . Ο περιορισμός αυτός δηλώνει το πλήθος των συνδέσεων ενός αντικειμένου  $a$  με άλλα αντικείμενα του χώρου ερμηνείας μέσω του ρόλου  $R$ . Μπορούμε να ορίσουμε το μέγιστο πλήθος  $n$  των συνδέσεων αυτών με τον κατασκευαστή at-most (σύνταξη  $\leq nR$ ) ή το ελάχιστο με τον κατασκευαστή at-least (σύνταξη  $\geq nR$ ) επιτρέποντας να ορίσουμε π.χ. την έννοια

του Πολύτεκνου *Άνθρωπος*  $\sqsupseteq$  4έχειΠαιδί ή την έννοια του Μοναχοπαίδι *Άνθρωπος*  $\sqcap$  1  $\leq$  είναιΠαιδί  $\sqcap$  1  $\geq$  είναιΠαιδί. Στην ειδική περίπτωση που ο περιορισμός πληθυκότητας λαμβάνει μόνο την τιμή 1 ο περιορισμός αυτός ονομάζεται συναρτησιακός περιορισμός πληθυκότητας (functional number restriction) και συμβολίζεται με το γράμμα  $\mathcal{F}$ . Τυπικά η ερμηνεία του περιορισμού πληθυκότητας σύμφωνα με την συνάρτηση ερμηνείας  $\cdot^{\mathcal{I}}$  είναι

$$(\leq nR)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \#\{b \mid (a,b) \in R^{\mathcal{I}}\} \leq n\} \text{ και}$$

$$(\geq nR)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \#\{b \mid (a,b) \in R^{\mathcal{I}}\} \geq n\}$$

ενώ του συναρτησιακού περιορισμού

$$(\text{func}(R))^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \#\{b \mid (a,b) \in R^{\mathcal{I}}\} = 1\}$$

Ο περιορισμός πληθυκότητας δεν παρέχει αρκετή εκφραστικότητα ώστε να περιορίζει σε ποιιά έννοια θα ανήκουν τα άτομα που συμμετέχουν στο ρόλο. Ο περιορισμός αυτός εισάγεται με τον κατασκευαστή του προσοντούχου περιοριστή πληθυκότητας (qualified cardinality restriction) και ο οποίος συμβολίζεται με το γράμμα  $\mathcal{Q}$ . Ο κατασκευαστής αυτός μας επιτρέπει για παράδειγμα να ορίσουμε την έννοια του πολύτεκνου όπου όλα του τα παιδιά είναι κορίτσια γράφοντας *Άνθρωπος*  $\sqsupseteq$  4έχειΠαιδί.Θηλυκό. Τυπικά η ερμηνεία του προσοντούχου περιορισμού πληθυκότητας σύμφωνα με την συνάρτηση ερμηνείας  $\cdot^{\mathcal{I}}$  είναι

$$(\leq nR.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \#\{b \mid (a,b) \in R^{\mathcal{I}} \text{ και } b \in C^{\mathcal{I}}\} \leq n\} \text{ και}$$

$$(\geq nR.C)^{\mathcal{I}} = \{a \in \Delta^{\mathcal{I}} \mid \#\{b \mid (a,b) \in R^{\mathcal{I}} \text{ και } b \in C^{\mathcal{I}}\} \geq n\}$$

Στις περιγραφικές λογικές μπορούμε να αποδώσουμε ονόματα στις περίπλοκες έννοιες που προκύπτουν από συνδιασμό απλών εννοιών και των κατασκευαστών που προσφέρει η εκάστοτε γλώσσα, αλλά και να ορίσουμε σχέσεις μεταξύ των εννοιών με την χρήση αξιωμάτων ορολογίας (terminological axioms). Συγκεκριμένα μπορούμε να ορίσουμε  $C \equiv D$ , το οποίο ονομάζεται αξίωμα ισοδυναμίας (equivalence axiom), δηλώνοντας ότι οι έννοιες C και D είναι ταυτόσημες ή να ορίσουμε  $C \sqsubseteq D$ , το οποίο ονομάζεται αξίωμα υπαγωγής (subsumption axiom ή inclusion axioms) δηλώνοντας ότι η έννοια D είναι πιο γενική από την έννοια C. Ένα σύνολο από αξιώματα αποτελεί το σώμα ορολογίας της βάσης γνώσης (Tbox terminology box).

### 3.4.2 DL-Lite

Στην ενότητα αυτή θα παρουσιάσουμε την γλώσσα ΠΛ DL-Lite. Η DL-Lite παρουσιάστηκε από τους Calvanese et al [12] ως μια γλώσσα που μπορεί να εκφράσει τα περισσότερα στοιχεία από την γλώσσα μοντελοποίησης UML και ταυτόχρονα να είναι βατή: η υπολογιστική της πολυπλοκότητα να παραμένει χαμηλή. Συγκεκριμένα τα προβλήματα συλλογιστικής στην DL-Lite μπορούν να αποφασιστούν στην χειρότερη περίπτωση (worst-case) σε πολυωνυμικό χρόνο. Αξίζει να αναφερθεί ότι προβλήματα συλλογιστικής ακόμη και σε χαμηλής εκφραστικότητας γλώσσες ΠΛ έχουν υψηλή πολυπλοκότητα. Ενδεικτικά αναφέρουμε ότι για την γλώσσα  $\mathcal{ALC}$ , η οποία επιτρέπει την χρήση μόνο των κατασκευαστών  $\neg C, C \sqcap D, C \sqcup D, \exists R.C, \forall R.C$  η πολυπλοκότητα είναι εκθετικού χρόνου (EXPTIME).

Στην γλώσσα DL-Lite οι βασικές κλάσεις (έννοιες) ορίζονται ως:

$$B ::= A \mid \exists R \mid \exists R^- \quad (0.1)$$

όπου A είναι μια ατομική έννοια και R ένας ατομικός ρόλος. Οι γενικές έννοιες ορίζονται ως:

$$C ::= B \mid \neg B \mid C_1 \sqcap C_2 \quad (0.2)$$

Επίσης ορίζονται αξιώματα υπαγωγής εννοιών

$$C \sqsubseteq D$$

όπου στο αριστερό μέρος πρέπει να υπάρχει βασική έννοια (B) και στο δεξιά μέρος μπορούμε να έχουμε γενική έννοια.

Ακόμη ορίζονται αξιώματα συναρτησιακών ρόλων  $\text{Func}(R), \text{Func}(R^-)$  όπου ο R είναι ένας ατομικός ρόλος και αξιώματα ατόμων :

$$B(a), R(a,b)$$

όπου a και b είναι άτομα(individuals).

Σημασιολογία της DL-Lite:

**Σύνταξη    Σημασιολογία**

$$\begin{array}{ll} A & A^I \subseteq \Delta^I \\ R & R^I \subseteq \Delta^I \times \Delta^I \\ \neg B & \Delta^I \setminus B^I \end{array}$$

$$\begin{aligned}
C_1 \sqcap C_2 & \quad C_1^{\exists} \cap C_2^{\exists} \\
\exists R & \quad \{c \mid \exists c'. (c, c') \in R^{\exists}\} \\
\exists R^- & \quad \{c \mid \exists c'. (c', c) \in R^{\exists}\} \\
B \sqsubseteq C & \quad B^{\exists} \sqsubseteq C^{\exists} \\
\text{Func}(R) & \quad (c, c') \in R^{\exists} \wedge (c, c'') \in R^{\exists} \rightarrow c' = c'' \\
\text{Func}(R^-) & \quad (c', c) \in R^{\exists} \wedge (c'', c) \in R^{\exists} \rightarrow c' = c''
\end{aligned}$$

### Πίνακας 3.4.3 Σημασιολογία της DL-Lite

Παρατηρούμε ότι η DL-Lite είναι αρκετά λιτή καθώς περιορίζει την χρήση των κατασκευαστών - άρνηση υπάρχει μόνο σε βασικές έννοιες και υπάρχει μόνο ο κατασκευαστής της σύζευξης και όχι διάζευξης ( $\sqcup$ ). Ακόμη η τομή εννοιών επιτρέπεται μόνο στο δεξί μέρος των αξιωμάτων. Με αυτό τον τρόπο πετυχαίνει να περιορίσει τις αλληλεπιδράσεις μεταξύ των κατασκευαστών της γλώσσας οι οποίες προκαλούν δυσβατότητες (intractabilities).

Παρόλη την χαμηλή εκφραστικότητα της η DL-Lite μπορεί να περιγράψει με επιτυχία τους περισσότερους (όχι όλους) φορμαλισμούς αναπαράστασης γνώσης που συναντάμε στα διαγράμματα κλάσεων UML και μοντέλα Οντότητας-Σχέσης (E-R models). Συγκεκριμένα με ένα σύνολο ισχυρισμών DL-Lite μπορούμε να προσδιορίσουμε:

- σχέσεις υπαγωγής (ISA), χρησιμοποιώντας ισχυρισμούς της μορφής  $A1 \sqsubseteq A2$ , δηλώνοντας ότι η κλάση  $A1$  είναι υποκλάση της κλάσης  $A2$ .
- ξένες έννοιες (disjointness), χρησιμοποιώντας ισχυρισμούς της μορφής  $A1 \sqsubseteq \neg A2$ , δηλώνοντας ότι η κλάση  $A1$  είναι ξένη της κλάσης  $A2$ .
- περιορισμούς στον τύπο του ρόλου (role-typing), χρησιμοποιώντας ισχυρισμούς της μορφής  $\exists R \sqsubseteq A_1$  (αντίστοιχα  $\exists R^- \sqsubseteq A_2$ ) δηλώνοντας ότι το πρώτο (αντίστοιχα το δεύτερο) μέλος του ρόλου  $R$  είναι τύπου  $A_1$  (αντίστοιχα  $A_2$ ).
- περιορισμό συμμετοχής (participation constraints) χρησιμοποιώντας ισχυρισμούς της μορφής  $A \sqsubseteq \exists R$  (αντίστοιχα  $A \sqsubseteq \exists R^-$ ) δηλώνοντας ότι τα

στιγμιότυπα της κλάσης A συμμετέχουν στον ρόλο R ως το πρώτο (αντίστοιχα το δεύτερο) μέλος.

- Περιορισμό μη συμμετοχής (non-participation constraints) χρησιμοποιώντας ισχυρισμούς της μορφής  $A \sqsubseteq \neg \exists R$  (αντίστοιχα  $A \sqsubseteq \neg \exists R^-$ ) δηλώνοντας ότι τα στιγμιότυπα της κλάσης A δεν συμμετέχουν στον ρόλο R ως το πρώτο (αντίστοιχα το δεύτερο) μέλος.
- συναρτησιακούς περιορισμούς (functionality restrictions), χρησιμοποιώντας ισχυρισμούς της μορφής  $\text{Func}(R)$  (αντίστοιχα  $\text{Func}(R^-)$ ) δηλώνοντας ότι ένα αντικείμενο μπορεί να είναι το πρώτο (αντίστοιχα το δεύτερο) μέλος του ρόλου R το πολύ μια φορά.

Ωστόσο με την DL-Lite δεν μπορούμε να έχουμε κάποιες δυνατότητες μοντελοποίησης που συναντάμε σε άλλες γλώσσες μοντελοποίησης (όπως π.χ. την UML). Ως παράδειγμα αναφέρουμε ότι με την DL-Lite δεν μπορούμε να έχουμε περιορισμό επικάλυψης (covering constraint), να δηλώσουμε δηλαδή ότι κάθε στιγμιότυπο κλάσης πρέπει να είναι στιγμιότυπο τουλάχιστον μιας από τις υποκλάσεις της. Ο λόγος που τέτοιες δυνατότητες κρατήθηκαν εκτός της εκφραστικής δυνατότητας της γλώσσας είναι (όπως αναφέρεται στην αρχή της ενότητας) για να κρατηθεί η υπολογιστική πολυπλοκότητα των προβλημάτων συλλογιστικής σε χαμηλά επίπεδα (πολυωνυμικού χρόνου).

Σημαντικό χαρακτηριστικό της DL-Lite είναι ότι μπορούμε να εκτελέσουμε ερωτήσεις σε γνώση που περιγράφεται από μια βάση γνώσης (knowledge base – KB) με πιο περιεκτικό τρόπο σε σύγκριση με άλλες γλώσσες περιγραφικών λογικών. Συγκεκριμένα ενώ σε άλλες οικογένειες ΠΛ μπορεί να ερωτηθεί μόνο η συμμετοχή σε μια έννοια ή ρόλο, η DL-Lite εισάγει την δυνατότητα για συνδυαστικά ερωτήματα (conjunctive queries). Η απάντηση συζευγμένων ερωτήσεων αποτελεί δύσκολο πρόβλημα για πιο εκφραστικές περιγραφικές λογικές και οι όποιοι αλγόριθμοι έχουν προταθεί για να λύσουν το πρόβλημα αυτό έχουν εν γένει μη ντετερμινιστικό χαρακτήρα [13]. Αντιθέτως στη DL-Lite το πρόβλημα αυτό είναι LOGSPACE ως προς το πλήθος των ατόμων της βάσης γνώσης.

Ένα συζευγμένο ερώτημα  $q$  πάνω σε μια βάση γνώσης  $K$  ορίζεται ως ένα ερώτημα της μορφής [14]:

$$q(\vec{x}) \leftarrow \exists \vec{y}. \text{conj}(\vec{x}, \vec{y})$$



όπου το  $\vec{x}$  είναι το διάνυσμα που περιλαμβάνει τις διακεκριμένες μεταβλητές (distinguished variables), το  $\vec{y}$  είναι το διάνυσμα που περιλαμβάνει τις υπαρκτικά προσοδεικτούμενες μεταβλητές που ονομάζονται μη-διακεκριμένες μεταβλητές (non-distinguished variables), το  $conj(\vec{x}, \vec{y})$  είναι σύζευξη από άτομα της μορφής  $B(z)$  ή  $R(z_1, z_2)$  (όπου  $B$  είναι βασική έννοια,  $R$  ρόλος και  $z, z_1, z_2$  άτομα στην βάση γνώσης).

Μια αποτίμηση (evaluation)  $[\vec{x} \rightarrow S]$  είναι μια αντικατάσταση των μεταβλητών του  $\vec{x}$  με άτομα από το σύνολο των ατόμων του  $S$ . Μια επερώτηση  $q(\vec{x}) \leftarrow \exists \vec{y}. conj(\vec{x}, \vec{y})$  ερμηνεύεται σε μια ερμηνεία  $\mathcal{I}$  ως το σύνολο  $q^{\mathcal{I}}$  των πλειάδων  $S$  τέτοιο ώστε το σώμα της επερώτησης  $\exists \vec{y}. conj(\vec{x}, \vec{y})$  αποτιμάτε σε αληθές στην αποτίμηση  $[\vec{x} \rightarrow S]$  και την ερμηνεία  $\mathcal{I}$ . Δοθείσας μιας αποτίμησης  $[\vec{x} \rightarrow S]$ , αν κάθε μοντέλο  $\mathcal{I}$  της  $O$  ικανοποιεί την  $q_{[\vec{x} \rightarrow S]}$ , λέμε ότι η  $O$  συνεπάγεται λογικά το  $q_{[\vec{x} \rightarrow S]}$ . Στην περίπτωση αυτή, το  $S$  ονομάζεται λύση της επερώτησης  $q$ . Πιο απλά, η απάντηση σε συνδυαστικά ερωτήματα είναι το σύνολο των πλειάδων (tuples) των ατόμων που όταν αντικαταστήσουν τις διακεκριμένες μεταβλητές κάνουν τις ερωτήσεις αληθείς μ.β.τ. βάση γνώσης. Για παράδειγμα το ερώτημα

$$q(x) \leftarrow \text{Άνδρας}(x) \wedge \text{Πατέρας}(y, x) \wedge (\text{Βασίλης}, y) : \text{Πατέρας}$$

(η οποία σημασιολογικά σημαίνει φέρε μου όλα τα άτομα που είναι εγγόνια – του Βασίλη) επιστρέφει όλα τα  $\text{Άτομο}_i$  που υπάρχουν στην βάση γνώσης για τα οποία η λογική παράσταση

$$\text{Άνδρας}(\text{Άτομο}_i) \wedge \text{Πατέρας}(y, \text{Άτομο}_i) \wedge (\text{Βασίλης}, y) : \text{Πατέρας}$$

αποτιμάται σε αληθές μ.β.τ. βάση γνώσης.

Η DL-Lite αποτελεί μια οικογένεια γλωσσών περιγραφικών λογικών με βάση την ΠΛ που παρουσιάσαμε εδώ. Στην περίπτωση της DL-Lite έχει δειχτεί [14] ότι με κατάλληλη επεξεργασία της οντολογίας  $O$  (κανονικοποίηση) ένα συνδυαστικό ερώτημα μπορεί να μετατραπεί σε ένωση απλών σχεσιακών ερωτημάτων (SQL queries). Η πιο βασική γλώσσα αυτής της οικογένειας είναι η  $DL-Lite_{core}$  και προκύπτει από τους κατασκευαστές που περιγράψαμε εδώ πλην του τελεστή της τομής και του συναρτησιακού τελεστή. Ερευνητικά υπήρξε προσπάθεια να επεκταθεί η εκφραστικότητα της  $DL-Lite_{core}$  προσθέτοντας κι άλλους κατασκευαστές κάτι που

οδηγεί σε δημιουργία άλλων συγγενών γλωσσών. Για παράδειγμα αν επιτρέψουμε την τομή σε έννοιες που εμφανίζονται μόνο στο αριστερό μέρος των αξιωμάτων υπαγωγής και τον συναρτησιακό τελεστή (func) τότε η γλώσσα ονομάζεται  $DL-Lite_{\mathcal{F},\Pi}$ . Αν επιτρέψουμε και τον προσοντούχο υπαρξιακό τελεστή στο δεξί μέρος τότε η γλώσσα ονομάζεται  $DL-Lite_{\mathcal{R},\Pi}$ . Η  $DL-Lite_{\mathcal{R},\Pi}$  επιτρέπει επίσης την υπαγωγή ρόλων της μορφής  $R_1 \sqsubseteq R_2$ . Αν επιτρέψουμε στην  $DL-Lite_{core}$  την ύπαρξη  $n$ -αδικής σχέσης στον υπαρξιακό τελεστή η γλώσσα ονομάζεται  $DLR-Lite_{core}$  και αντίστοιχα με την πρόσθεση της τομής και του συναρτησιακού τελεστή  $DLR-Lite_{\mathcal{F},\Pi}$  και με την περαιτέρω πρόσθεση του προσοντούχου υπαρξιακού τελεστή  $DLR-Lite_{\mathcal{R},\Pi}$ . Για την οικογένεια αυτή της DL-Lite έχει αποδεικτεί [15] ότι η απάντηση ερωτήσεων ανάγεται σε ερωτήματα λογικής πρώτης τάξης επομένως υπολογίζονται σε χώρο LOGSPACE με βάση το μέγεθος του ABOX. Ωστόσο αν επιχειρηθεί περαιτέρω ανάπτυξη της εκφραστικότητας της γλώσσας ακόμα και με μικρές αλλαγές τότε η υπολογιστική πολυπλοκότητα των ερωτημάτων αυξάνεται δραστικά. Παραδείγματος χάρη αν επιτρέψουμε την ύπαρξη καθολικού περιορισμού  $(\forall R.A)$  η πολυπλοκότητα γίνεται NLOGSPACE-hard.

### 3.4.3 Ασαφής DL-Lite (fuzzy-DL-Lite)

Στις υπηρεσίες σημασιολογικού ιστού συχνά συναντάμε πληροφορία όπου υπάρχει ασάφεια ή αβεβαιότητα. Για παράδειγμα έννοιες όπως Γρήγορο ή Κοντά είναι έννοιες όπου η συμμετοχή ενός ατόμου σε αυτές δεν μπορεί να αποδοθεί πλήρως. Οι ασαφείς γλώσσες περιγραφικής λογικής επεκτείνουν τις παραδοσιακές γλώσσες ΠΛ δίνοντας την δυνατότητα να διαχειριστούν ασαφή/ανακριβή γνώση [16]. Η κύρια ιδέα των ασαφών ΠΛ είναι ότι ένας ισχυρισμός  $a:C$ , που δηλώνει ότι το άτομο  $a$  είναι στιγμιότυπο της έννοιας  $C$ , αντί να ερμηνεύεται ως αληθές ή ψευδές να απεικονίζεται σε τιμή αληθείας  $n \in [0,1]$ . Αποτέλεσμα αυτού είναι το  $n$  να επισημαίνει τον βαθμό με τον οποίο το  $a$  ανήκει στην έννοια  $C$ .

Η σημασιολογία των ασαφών περιγραφικών λογικών ορίζεται με την βοήθεια των ασαφών ερμηνειών (fuzzy interpretations). Μια ασαφής ερμηνεία αποτελείται από ένα ζευγάρι  $(\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$  όπου ο χώρος ερμηνείας  $\mathcal{I}$  είναι ένα μη κενό σύνολο

αντικειμένων και  $\cdot^{\mathcal{I}}$  είναι μια ασαφής συνάρτηση ερμηνείας (fuzzy interpretation function), η οποία απεικονίζει

1. Ένα άτομο  $a \in \mathbf{I}$  σε ένα στοιχείο  $a^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
2. μια ατομική έννοια  $A \in \mathbf{C}$  σε μια συνάρτηση συμμετοχής  $A^{\mathcal{I}} : \Delta^{\mathcal{I}} \rightarrow [0,1]$
3. ένα ατομικό ρόλο  $R \in \mathbf{R}$  σε μια συνάρτηση συμμετοχής της μορφής  $R^{\mathcal{I}} : \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \rightarrow [0,1]$ .

Η ασαφής DL-Lite είναι η περιγραφική γλώσσα (προτάθηκε από τον Straccia [17]) που επεκτείνει την κλασική  $DL-Lite_{core}$  δίνοντας την δυνατότητα για εισαγωγή ασαφών ισχυρισμών. Όπως αποδεικνύει ο Straccia οι αλγόριθμοι συλλογιστικής στην ασαφή DL-Lite είναι παρόμοιοι με αυτούς στην κλασική DL-Lite.

### Σύνταξη Σημασιολογία

$$\top \quad \top^{\mathcal{I}}(x) = 1$$

$$\perp \quad \perp^{\mathcal{I}}(x) = 0$$

$$A \quad A^{\mathcal{I}}(x) \in [0,1]$$

$$\exists R \quad (\exists R)^{\mathcal{I}}(o_1) = \sup_{o_2 \in \Delta^{\mathcal{I}}} \{R^{\mathcal{I}}(o_1, o_2)\}$$

$$\neg B \quad (\neg B)^{\mathcal{I}}(o) = 1 - B^{\mathcal{I}}(o)$$

$$C_1 \sqcap C_2 \quad (C_1 \sqcap C_2)^{\mathcal{I}}(o) = t(C_1^{\mathcal{I}}(o), C_2^{\mathcal{I}}(o))$$

$$R^- \quad (R^-)^{\mathcal{I}}(o_2, o_1) = R^{\mathcal{I}}(o_1, o_2)$$

$$B \sqsubseteq C \quad \forall o \in \Delta^{\mathcal{I}}, B^{\mathcal{I}}(o) \leq C^{\mathcal{I}}(o)$$

$$Func(R) \quad \forall o_1 \in \Delta^{\mathcal{I}}, \#\{o_2 \mid R^{\mathcal{I}}(o_1, o_2) > 0\} = 1$$

$$B(a) \geq n \quad B^{\mathcal{I}}(a^{\mathcal{I}}) \geq n$$

$$R(a, b) \geq n \quad R^{\mathcal{I}}(a^{\mathcal{I}}, b^{\mathcal{I}}) \geq n$$

### Πίνακας 3.4.3 Σημασιολογία της fuzzy-DL-Lite

Στην fuzzy-DL-Lite χρησιμοποιείται ως ερμηνεία της άρνησης το συμπλήρωμα του Łukasiewicz δηλαδή  $c(a) = 1 - a$  ενώ για ερμηνεία της τομής χρησιμοποιείται η τ-νόρμα του Godel (ή τυπική τομή) δηλαδή  $t(a, b) = \min(a, b)$ .

Προβλήματα συλλογιστικής στην fuzzy DL Lite

Συνέπεια (consistency): Έλεγχος αν η ασαφής βάση γνώσης είναι συνεπής δηλαδή αν υπάρχει μοντέλο όπου η ερμηνεία κάθε έννοιας να μην είναι το κενό σύνολο.

Υπαγωγή (Subsumption): έλεγχος αν η έννοια  $C$  υπάγεται στην έννοια  $D$  με βάση το σώμα ορολογίας της βάσης γνώσης δηλαδή αν  $\mathcal{K} \models C \sqsubseteq D$

Συνεπαγωγή (Entailment): έλεγχος αν το άτομα  $a$  είναι στιγμιότυπο της έννοιας  $C$  σε βαθμό μεγαλύτερο του  $n$  δηλαδή αν  $\mathcal{K} \models C(a) \geq n$

Πρόβλημα εύρεσης του καλύτερου ορίου αλήθειας (Best Truth Value Bound problem): εύρεση της μεγαλύτερης κατώτερης τιμής (greatest lower bound - glb) όπου ένα άτομο συμμετέχει στην έννοια  $C$ , τυπικά  $\text{glb}(\mathcal{K}, C(a)) = \sup \{n \mid \mathcal{K} \models C(a) \geq n\}$

Ασαφής ανάκτηση (Fuzzy Retrieval): Ανάκτηση των κορυφαίων  $k$  ατόμων με την μεγαλύτερη μεγαλύτερη κατώτερη τιμή δηλαδή τα πρώτα  $k$  στοιχεία σε κατάταξη του συνόλου  $\{(a, \text{glb}(\mathcal{K}, C(a)))\}$

Τα συνδεδεικμένα ερωτήματα που παρουσιάσαμε στην DL-Lite μπορούν να επεκταθούν στην περίπτωση της ασαφής DL-Lite και να σχηματίσουν βελτιωμένες γλώσσες επερωτήσεων. Η αποτίμηση τέτοιων ερωτήσεων όπως και στην κλασική DL-Lite μπορεί με κατάλληλο επαναπροσδιορισμό να αποτιμηθούν με μεγάλη αποδοτικότητα σε σχεσιακά ερωτήματα για παραδοσιακές βάσεις δεδομένων. Οι γλώσσες επερωτήσεων που συναντούμε στην ασαφή DL-lite διακρίνονται σε:

### 1. Συζευγμένες Επερωτήσεις Κατωφλίων (conjunctive threshold queries).

Οι επερωτήσεις αυτές επεκτείνουν τα άτομα  $B(z), R(z_1, z_2)$  της συζευγμένης ερώτησης σε ασαφείς ισχυρισμούς της μορφής  $B(z) \geq t_1$  και  $R(z_1, z_2) \geq t_2$  όπου  $t_1$  και  $t_2$  τα κατώφλια. Η λύση είναι το ασαφές σύνολο των πλειάδων όπου οι ασαφείς ισχυρισμοί ικανοποιούνται ταυτόχρονα (δηλαδή μια πλειάδα είτε ανήκει είτε δεν ανήκει στο σύνολο λύσεων).

### 2. Γενικευμένες ασαφείς επερωτήσεις

Είναι ερωτήσεις που όπως και στις συζευγμένες επερωτήσεις κατωφλίων επεκτείνουν τα άτομα  $B(z), R(z_1, z_2)$  της συζευγμένης ερώτησης σε ασαφείς ισχυρισμούς της μορφής  $B(z) \geq k_1$  και  $R(z_1, z_2) \geq k_2$  όπου  $k_1$  και  $k_2$  βαθμοί. Η αποτίμηση τέτοιων ερωτήσεων είναι ένα σύνολο από τα άτομα που ικανοποιούν

την ερώτηση μαζί με το βαθμό αληθείας της ερώτησης στην ερμηνεία της ασαφής οντολογίας. Ο βαθμός υπολογίζεται ως:

$$d = \sup_{S' \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \times \dots \times \Delta^{\mathcal{I}}} \{G_{i=1}^n a(k_i, atom_i^{\mathcal{I}}(\bar{v})_{[X \rightarrow S, Y \rightarrow S']})\} \text{ όπου}$$

$G$  ο τελεστής σύζευξης των ασαφών ισχυρισμών

$a$  συνάρτηση αποτίμησης των βαθμών που έχουμε προσαρτήσει σε κάθε άτομο με αυτόν του ατόμου.

$k_i$  βαθμοί των ατόμων της ερώτησης

$atom_i(\bar{v})$  τα άτομα των συζευγμένων επερωτήσεων (είτε  $C(a) \geq k_1$  είτε  $R(a, b) \geq k_2$ ).

Ανάλογα με ποιό τελεστή  $G$  ή συνάρτηση  $a$  επιλεγεί οι γενικευμένες ασαφείς ερωτήσεις μπορεί να είναι:

α. Επερωτήσεις ασαφών κατωφλίων (fuzzy threshold queries)

Εδώ επιλέγεται για τελεστή διάζευξης η  $\tau$ -νόρμα ( $t$ ) και ως συνάρτηση αποτίμησης η R-συνεπαγωγή ( $\omega$ ). Έτσι ο βαθμός αληθείας υπολογίζεται ως:

$$d = \sup_{S' \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \times \dots \times \Delta^{\mathcal{I}}} \{t_{i=1}^n \omega_i(k_i, atom_i^{\mathcal{I}}(\bar{v})_{[X \rightarrow S, Y \rightarrow S']})\}$$

Οι ερωτήσεις αυτού του είδους έχουν την ιδιότητα να επιστρέφουν βαθμό αληθείας σε άτομα ακόμα κι αν ο βαθμός της ερώτησης ( $k_i$ ) είναι μεγαλύτερος από το βαθμό του ατόμου απλώς η συνάρτηση της R-συνεπαγωγής προσθέτει σταδιακά ποινή (ουσιαστικά ελαττώνει) στον βαθμό αληθείας της ερώτησης.

β. Επερωτήσεις ασαφών συναθροίσεων (fuzzy aggregation queries)

Ως τελεστής συνάθροισης χρησιμοποιείται η συνάρτηση ασαφούς

συνάθροισης  $G(x) = \frac{\sum_{i=1}^n x_i}{\sum_{i=1}^n k_i}$  και ως συνάρτηση αποτίμησης η συνάρτηση

$a(k_i, y) = k_i * y$  επομένως ο βαθμός αληθείας αποτιμάται ως:

$$d = \sup_{S' \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \times \dots \times \Delta^{\mathcal{I}}} \left\{ \frac{\sum_{i=1}^n k_i * atom_i^{\mathcal{I}}(\bar{v})_{[X \rightarrow S, Y \rightarrow S']}}{\sum_{i=1}^n k_i} \right\}$$



# 4

## Ανάλυση

### Απαιτήσεων Συστήματος

Το σύστημα που υλοποιήσαμε καλείται να διεκπεραιώσει μια σειρά από λειτουργίες που θα βοηθήσουν τους χρήστες της κοινότητας του εργαστηρίου να οργανώσουν και να μοιραστούν τις δημοσιεύσεις που τους ενδιαφέρουν. Υλοποιήθηκε ως μια διακτυακή εφαρμογή και όχι ως πρόγραμμα που τρέχει τοπικά σε προσωπικό υπολογιστή, ακριβώς για να είναι δυνατή η πρόσβαση ανά πάσα στιγμή από τους χρήστες χωρίς να είναι αναγκαία η εγκατάσταση οποιουδήποτε λογισμικού, αρκεί η χρήση ενός σύγχρονου περιηγητή (browser) και σύνδεση στο διαδίκτυο.

#### 4.1 Περιγραφή Λειτουργιών

Στην ενότητα αυτή περιγράφονται οι λειτουργίες του συστήματος

##### 4.1.1 Χαρακτηρισμός δημοσίευσης (annotation)

Το σύστημα δίνει την δυνατότητα στον χρήστη να ανεβάσει (upload) μια δημοσίευση που τον ενδιαφέρει και επιθυμεί να μοιραστεί με άλλους χρήστες. Η δημοσίευση αυτή πρέπει να χαρακτηριστεί (annotated) σε ποιά ερευνητική περιοχή ανήκει. Επειδή μια δημοσίευση μπορεί να ανήκει ταυτόχρονα σε πολλές ερευνητικές περιοχές και όχι πάντα σε βαθμό 1 ο χρήστης πρέπει να δύναται να χαρακτηρίσει μια δημοσίευση ότι ανήκει σε συγκεκριμένη ερευνητική περιοχή σε κάποιο βαθμό και όχι εξ'ολοκλήρου (fuzzy annotation). Πέραν του χαρακτηρισμού των ερευνητικών περιοχών, να δίνεται η δυνατότητα να προσθέσει ετικέτες (tags) στην δημοσίευση, να προσθέτει σχόλια (comments) και να αξιολογεί (rating) την δημοσίευση.

#### ***4.1.2 Αναζήτηση με βάση την ερευνητική περιοχή***

Ο χρήστης πρέπει να έχει την δυνατότητα να αναζητήσει δημοσιεύσεις με βάση την ερευνητική περιοχή που έχουν χαρακτηριστεί από άλλους χρήστες. Η αναζήτηση δεν πρέπει να περιορίζεται σε μια ερευνητική περιοχή αλλά να μπορεί να κάνει σύνθετες αναζητήσεις με πολλές ερευνητικές περιοχές και να θέτει τον βαθμό που τον ενδιαφέρει για κάθε ερευνητική περιοχή.

#### ***4.1.3 Αναζήτηση με βάση τα περιγραφικά χαρακτηριστικά της δημοσίευσης***

Πέραν της αναζήτησης με βάση την ερευνητική περιοχή που έχει χαρακτηριστεί η εκάστοτε δημοσίευση πρέπει να δίνεται η δυνατότητα στον χρήστη να αναζητήσει τις υπάρχουσες δημοσιεύσεις και με βάση τα άλλα χαρακτηριστικά της δημοσίευσης όπως είναι ο τίτλος, ο συγγραφέας, το επιστημονικό περιοδικό που έχει δημοσιευτεί κτλ. Ακόμη να δίνεται η δυνατότητα ο χρήστης να μπορεί να αναζητήσει μέσα στο περιεχόμενο – κείμενο των δημοσιεύσεων με βάση λέξη κλειδί (keyword).

#### ***4.1.4 Ανάκτηση δημοσιεύσεων με βάση την ετικέτα (προσωπική ή της κοινότητας)***

Ο χρήστης του συστήματος να μπορεί εύκολα να ανακτήσει (retrieve) τις δημοσιεύσεις στις οποίες έχει ο ίδιος προσδώσει μια συγκεκριμένη ετικέτα (tag) ή να εξερευνήσει τις δημοσιεύσεις στις οποίες η υπόλοιπη κοινότητα έχει προσδώσει μια συγκεκριμένη ετικέτα.

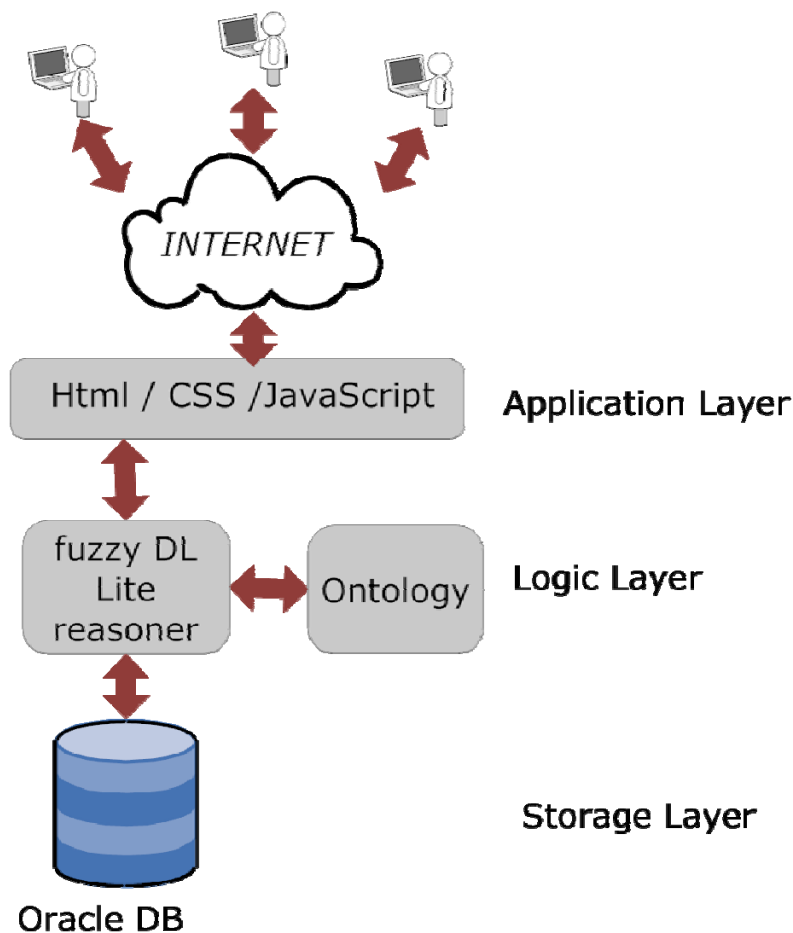
#### ***4.1.5 Περιήγηση,εποπτεία και ανάκτηση των δημοσιεύσεων άλλων χρηστών***

Ο κάθε χρήστης πρέπει να έχει την δυνατότητα να εξερευνήσει τις δημοσιεύσεις που έχουν χαρακτηρίσει οι άλλοι χρήστες ή έχουν προσδώσει σε αυτές ετικέτες. Σε κάθε δημοσίευση να μπορεί να δει τα πλήρη χαρακτηριστικά της (τίτλος, συγγραφείς, ερευνητικές περιοχές κτλ) και να μπορεί να προσθέσει σχόλια (comments), να αξιολογήσει και ο ίδιος την δημοσίευση (η καινούργια αξιολόγηση να βγαίνει από το μέσο όρο των αξιολογήσεων των άλλων χρηστών) και φυσικά να μπορεί να κατεβάσει το αρχείο της δημοσίευσης. Ακόμη αν πιστεύει ότι έχει γίνει λάθος χαρακτηρισμός μιας δημοσίευσης σε ποιες ερευνητικές περιοχές ανήκει, να δύναται να αλλάξει αυτόν τον χαρακτηρισμό.



## 4.2 Αρχιτεκτονική

Στην ενότητα αυτή παρουσιάζεται η αρχιτεκτονική του συστήματος. Όπως βλέπουμε στο διάγραμμα 4.1 η εφαρμογή διαχωρίζεται σε στρώματα διαλειτουργικότητας.



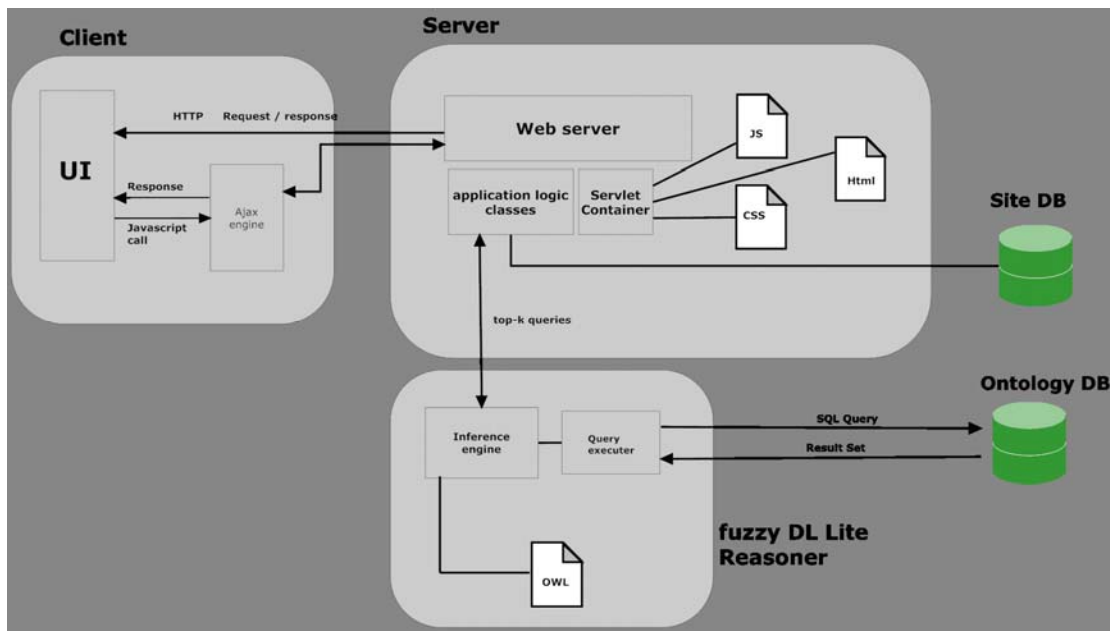
Εικόνα 4.2.1 Αρχιτεκτονική συστήματος

- Στο πάνω στρώμα βρίσκονται οι χρήστες οι οποίοι επικοινωνούν με την εφαρμογή μέσω του διαδικτύου.
- Στο αμέσως πιο κάτω επίπεδο βρίσκεται το επίπεδο της εφαρμογής η οποία αποτελείται από το γραφικό περιβάλλον με το οποίο ο χρήστης αλληλεπιδρά με τα κατώτερα λογικά στρώματα.
- Παρακάτω βρίσκεται το λογικό επίπεδο που αποτελείται από την μηχανή συμπερασματολογίας (reasoner) και την οντολογία που έχει αναπτυχθεί για να περιγράψει την γνώση του πεδίου που εργαζόμαστε.

- Τέλος στο κατώτερο στρώμα βρίσκεται το επίπεδο της αποθήκευσης (storage). Εδώ βρίσκεται μια παραδοσιακή σχεσιακή βάση δεδομένων ( στην περίπτωση μας Oracle Database).

Είναι σημαντικό να τονίσουμε ότι η αλληλεπίδραση του χρήστη με τα δεδομένα γίνεται με χρήση του μεσαίου στρώματος -όταν πρόκειται για αναζήτηση σε έννοιες της οντολογίας- και όχι απ'ευθείας. Αυτό εξασφαλίζει την ορθότητα των αναζητήσεων και διευκολύνει τον χρήστη στην εξερεύνηση της γνώσης που περιγράφεται στην οντολογία χωρίς να χρειάζεται να αναπτυχθούν εξεζητημένες τεχνικές που θα υπακούουν μόνο στο συγκεκριμένο σχήμα (schema) της βάσης δεδομένων. Με αυτό το τρόπο το σύστημα είναι αρκετά εύκαμπτο σε περίπτωση που για κάποιο λόγο η οντολογία αλλάξει: δεν χρειάζεται να γίνουν εκτενείς (και επίπονες) αλλαγές στο στρώμα της εφαρμογής.

Συνεχίζοντας βλέπουμε στο διάγραμμα 4.2 μια πιο λεπτομερή περιγραφή της αρχιτεκτονικής του συστήματος.



Εικόνα 4.2.2 Λεπτομερής αρχιτεκτονική και υποσυστήματα της εφαρμογής

Παρατηρούμε το λογικό διαχωρισμό του συστήματος στα υποσυστήματα:

#### 4.2.1 Υποσύστημα του πελάτη

Το υποσύστημα του πελάτη αποτελείται ουσιαστικά από ένα σύγχρονο περιηγητή (browser) ο οποίος μπορεί να αποδώσει (render) σελίδες διαδικτύου οι οποίες περιέχουν απλά στοιχεία Html, CSS και JavaScript. Στη μονάδα αυτή υλοποιείται μέσω JavaScript μια μηχανή Ajax η οποία μπορεί να καλέσει ασύγχρονα το υποσύστημα του εξυπηρετητή και να λάβει αποκρίσεις από δεδομένα σε αδόμητη μορφή και αφού τα επεξεργαστεί να τα εμφανίσει σε παρουσιάσιμη μορφή στον χρήστη.

#### 4.2.2 Υποσύστημα του εξυπηρετητή

Το υποσύστημα του εξυπηρετητή αποτελείται από ένα εξυπηρετητή ιστού ο οποίος μπορεί να λάβει αιτήσεις HTTP και να δημιουργήσει δυναμικά σελίδες HTML ως αποκρίσεις στο σύστημα του πελάτη. Σε αυτό βρίσκονται όλοι οι πόροι που αφορούν τις στατικές σελίδες Html, τα επικαλυπτόμενα φύλλα στυλ (CSS) και αρχεία JavaScript. Οι δυναμικές σελίδες δημιουργούνται με χρήση τεχνολογιών JSP (JavaServerPages) και JavaServlets βασιζόμενοι στις παραμέτρους που περιλαμβάνονται στις αιτήσεις του πελάτη. Επίσης σε αυτό συμπεριλαμβάνεται το σύνολο των Java κλάσεων που υλοποιούν την επικοινωνία με το υποσύστημα της μηχανής συμπερασματολογίας. Συγκεκριμένα οι κλάσεις αυτές αναλαμβάνουν να αποσαφηνίσουν τα ερωτήματα που θέτει ο πελάτης και αφού σχηματίσουν συνδεδετικά ερωτήματα εκτελούν επερωτήσεις ασαφής συνάθροισης στη μηχανή συμπερασματολογίας για να λάβουν τα κορυφαία κ-άτομα που πληρούν τα κριτήρια του συνδεδετικού ερωτήματος. Ακόμη στο υποσύστημα αυτό υλοποιείται απ'ευθείας επικοινωνία με το υποσύστημα της βάσης δεδομένων για δεδομένα που αφορούν τη λειτουργία του συστήματος (όπως είναι π.χ. οι λογαριασμοί των χρηστών, οι ετικέτες, τα σχόλια κτλ).

#### ***4.2.3 Υποσύστημα της μηχανής συμπερασματολογίας***

Το υποσύστημα της μηχανής συμπερασματολογίας υλοποιεί τη βάση γνώσης που χρησιμοποιεί το σύστημα. Χρησιμοποιεί ως σώμα ισχυρισμών (TBox) το σύνολο των ισχυρισμών που εξάγει από αρχείο .owl που ορίζει την οντολογία που χρησιμοποιείται ενώ το σώμα ισχυρισμών (ABox) αποθηκεύεται στο υποσύστημα της βάσης δεδομένων. Υλοποιεί τη μηχανή συμπερασματολογίας έχοντας τη δυνατότητα για ασαφή ερωτήματα όπως αυτά περιγράφονται στις προδιαγραφές της ασαφούς DL-Lite (βλ. Ενότητα 3.4.3).

#### ***4.2.4 Υποσύστημα της βάσης δεδομένων***

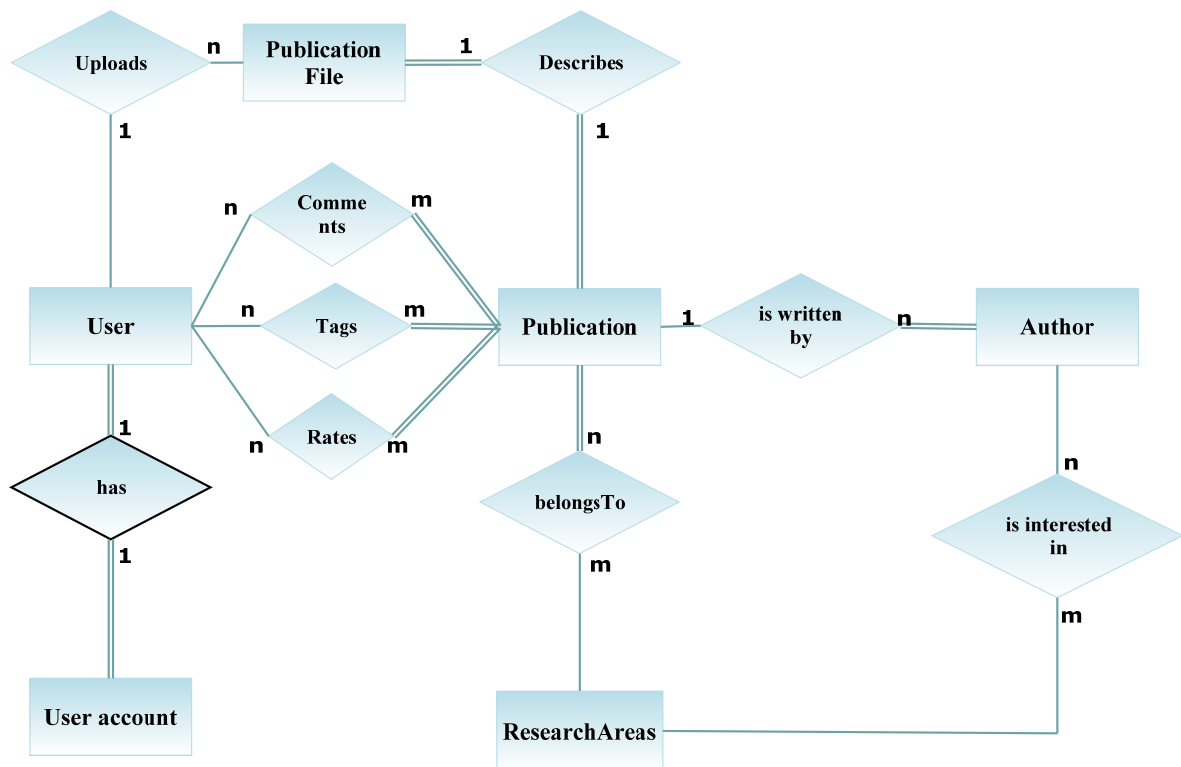
Το υποσύστημα της βάσης δεδομένων διαχωρίζεται σε δυο λογικά μέρη: το σχήμα που χρησιμοποιείται για μόνιμη αποθήκευση των ισχυρισμών της μηχανής συμπερασματολογίας και το σχήμα που αποθηκεύει δεδομένα που αφορούν τη λειτουργία της ιστοσελίδας (λογαριασμοί χρηστών, ετικέτες, σχόλια κτλ)

### ***4.3 Μοντέλο Οντοτήτων Συσχετίσεων***

Το διάγραμμα οντότητας-σχέσης (entity-relation diagram) μπορεί να εκφράσει γραφικά τη συνολική δομή μιας βάσης δεδομένων. Αποτελούν μια αφηρημένη και εννοιολογική απεικόνιση των δεδομένων. Τα διαγράμματα οντότητας-σχέσης είναι απλά και σαφή, ποιότητες στις οποίες οφείλεται η ευρεία χρήση του μοντέλου οντότητας-σχέσης. Μπορούν να εκφράσουν σύνολα οντοτήτων, ιδιότητες οντοτήτων, σχέσεις μεταξύ οντοτήτων και άλλα

ποιοτικά χαρακτηριστικά οντοτήτων και ιδιοτήτων (αδυναμία συνόλου οντότητας, ολική συμμετοχή, εξειδίκευση, γενίκευση κ.α.).[18]

Παρακάτω παρατίθεται το διάγραμμα οντοτήτων-σχέσεων που αφορά το σχήμα της βάσης δεδομένων που υλοποιήθηκε με παραδοσιακές μεθόδους. Σε αυτό ξεχωρίζουμε τις βασικές οντότητες του Χρήστη (User), Λογαριασμού Χρήστη (User Account), Δημοσίευσης (Publication), Αρχείου Δημοσίευσης (Publication file), Συγγραφέα (Author) και Ερευνητικής περιοχής (Research Area) που απεικονίζονται με ορθογώνιο. Η κάθε οντότητα ενώνεται με άλλη οντότητα μέσω σχέσεων που απεικονίζονται με ρόμβους. Έτσι έχουμε τις σχέσεις έχει(has), ανεβάζει (Uploads), περιγράφει (describes), σχολιάζει (comments), ετικετοποιεί (tags), αξιολογεί (rates), ανήκειΣε (belongsTo), γράφτηκεΑπό (is written by) και ενδιαφέρεται (is interested in). Οι ιδιότητες κάθε οντότητας και σχέσης απουσιάζουν λόγω οικονομίας χώρου.



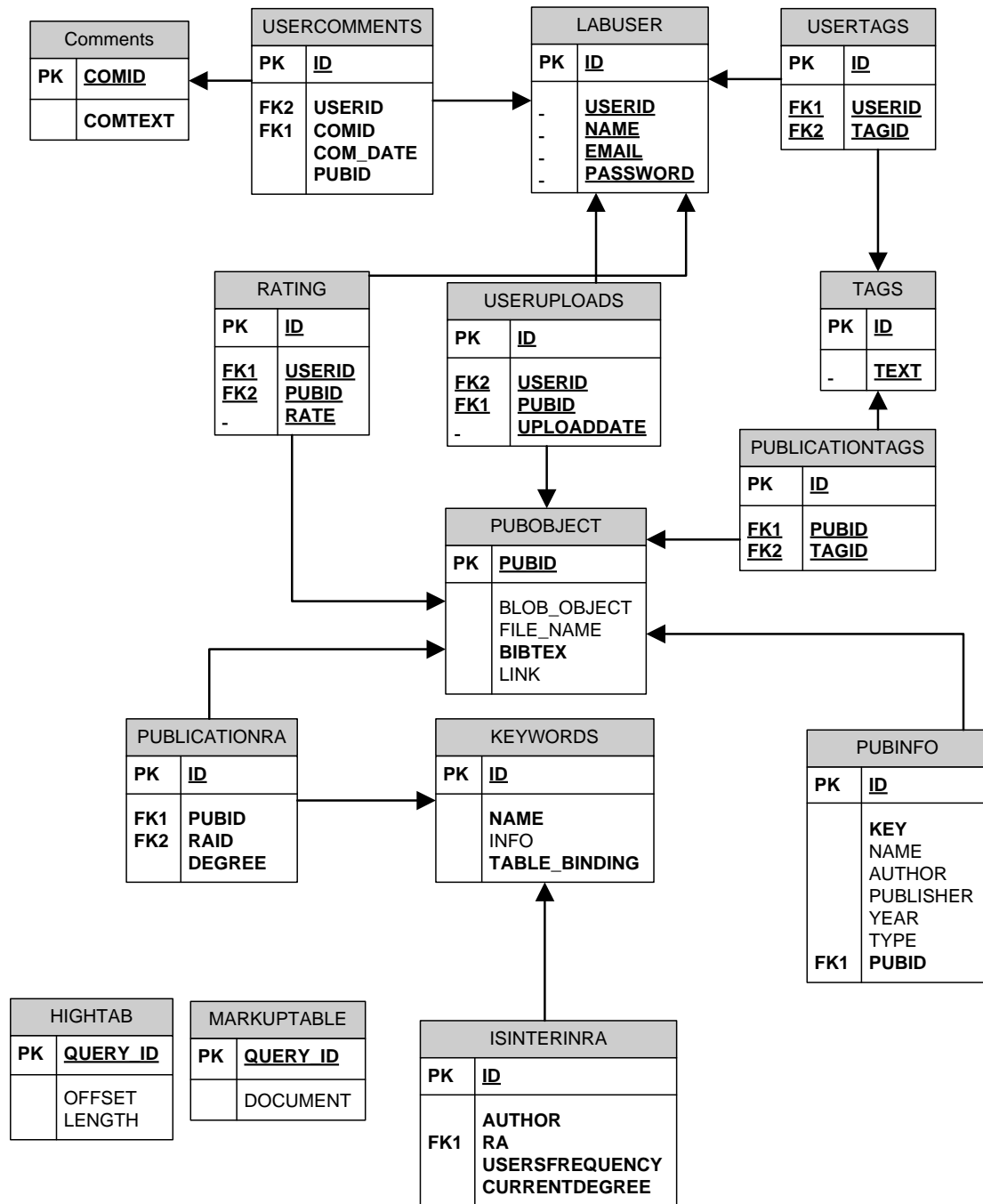
Εικόνα 4.3.1 Μοντέλο Οντοτήτων-Συσχετίσεων

#### 4.4 Σχεσιακό διάγραμμα βάσης δεδομένων

Το σχεσιακό διάγραμμα βάσης δεδομένων αποτελεί σήμερα το βασικό μοντέλο δεδομένων για εμπορικές εφαρμογές επεξεργασίας δεδομένων. Μπορεί να συγκρατήσει όλη την απαραίτητη πληροφορία όσο αφορά τη δομή των δεδομένων και διευκολύνει τον προγραμματιστή στην ανάπτυξη της εφαρμογής.

Παρακάτω παρατίθεται το σχεσιακό διάγραμμα του σχήματος της βάσης δεδομένων που χρησιμοποιήθηκε ως μοντέλο δεδομένων για τις οντότητες οι οποίες σκόπιμα δεν συμπεριλήφθηκαν στην οντολογία καθώς δεν έχουν κάποια ιδιαίτερη σημασιολογική αξία.

Ακολούθως περιγράφεται λεπτομερώς ο κάθε πίνακας που εμφανίζεται στο διάγραμμα.



Εικόνα 4.4.1 Σχεσιακό διάγραμμα

### Πίνακας LABUSER

Σε αυτόν τον πίνακα αποθηκεύονται τα στοιχεία του χρήστη (user account) και περιγράφεται η οντότητα του χρήστη του συστήματος.

#### Πεδία:

- ID** Το πεδίο που περιγράφει μοναδικά μια εγγραφή στον πίνακα (primary key). Προκύπτει από την ακολουθία labuser\_seq όπου για κάθε νέα εγγραφή στον

πίνακα αυξάνεται κατά ένα.

**USERID** Το πεδίο που περιγράφει το επιθυμητό όνομα του χρήστη (nickname). Είναι τύπου varchar2 και έχει μήκος 30 χαρακτήρων.

**NAME** Το πεδίο που περιγράφει το πραγματικό όνομα του χρήστη. Είναι τύπου varchar2 και έχει μήκος 30 χαρακτήρων.

**EMAIL** Το πεδίο που περιγράφει την ηλεκτρονική διεύθυνση του χρήστη. Είναι τύπου varchar2 και έχει μήκος 70 χαρακτήρων.

**PASSWORD** Το πεδίο που περιγράφει τον κωδικό προσπέλασης του χρήστη. Σημειώνουμε ότι δεν αποθηκεύεται στην πραγματική του μορφή για λόγους ασφαλείας αλλά μετά από κωδικοποίηση MD5. Είναι τύπου varchar2 και έχει μήκος 250 χαρακτήρων.

### Πίνακας COMMENTS

Ο πίνακας που αποθηκεύει την πληροφορία για τα σχόλια που γίνονται στις δημοσιεύσεις.

#### **Πεδία:**

**ID** Το πεδίο που περιγράφει μοναδικά μια εγγραφή στον πίνακα (primary key). Προκύπτει από την ακολουθία comments\_seq όπου για κάθε νέα εγγραφή στον πίνακα αυξάνεται κατά ένα.

**TEXT** Το πεδίο που περιγράφει το κείμενο του σχολίου. Είναι τύπου varchar2 και έχει μήκος 1000 χαρακτήρων.

### Πίνακας USERCOMMENTS

Ο πίνακας που συγκρατά την πληροφορία για τα σχόλια που έκανε ο κάθε χρήστης και σε ποιά δημοσίευση. Ο σχεσιακός αυτός πίνακας αναλαμβάνει να υλοποιήσει την σχέση στο διάγραμμα ER comments (που ενώνει την οντότητα User με Publication).

#### **Πεδία:**

**ID** Το πεδίο που περιγράφει μοναδικά μια εγγραφή στον πίνακα (primary key). Προκύπτει από την ακολουθία usercomments\_seq όπου για κάθε νέα εγγραφή στον πίνακα αυξάνεται κατά ένα.

**USERID** Το πεδίο αυτό συγκρατεί το id του χρήστη που έκανε το σχόλιο και υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα LABUSER (πρέπει δηλαδή να υπάρχει αντίστοιχη εγγραφή στον πίνακα αυτό).

**COMID** Το πεδίο αυτό συγκρατεί το id του σχόλιου που έγινε από τον συγκεκριμένο χρήστη στην συγκεκριμένη δημοσίευση. Υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα COMMENTS.

**PUBID** Το πεδίο αυτό συγκρατεί το id της δημοσίευσης στην οποία έγινε σχόλιο. Υπάρχει περιορισμός ξένου κλειδιού με το πίνακα PUBOBJECT

**COM\_DATE** Το πεδίο αυτό συγκρατεί την ημερομηνία (και ώρα) που έγινε το συγκεκριμένο σχόλιο και είναι τύπου TIMESTAMP.

### Πίνακας TAGS

Ο πίνακας αυτός χρησιμοποιείται για να συγκρατήσει την πληροφορία για τις ετικέτες (tags) που έγιναν σε δημοσιεύσεις.

#### **Πεδία:**

**ID** Το πεδίο που περιγράφει μοναδικά μια εγγραφή στον πίνακα (primary key). Προκύπτει από την ακολουθία tags\_seq όπου για κάθε νέα εγγραφή στον πίνακα αυξάνεται κατά ένα.

**TEXT** Το πεδίο αυτό περιγράφει το κείμενο της ετικέτας. Είναι τύπου varchar2 και έχει μήκος 25 χαρακτήρες.

### Πίνακας USERTAGS

Ο πίνακας που συγκρατά την πληροφορία για τις ετικέτες που προσέδωσαν οι χρήστες σε δημοσιεύσεις. Ο σχεσιακός αυτός πίνακας αναλαμβάνει να υλοποιήσει την σχέση στο διάγραμμα ER TAGS (που ενώνει την οντότητα User με Publication).

#### **Πεδία:**

**ID** Το πεδίο που περιγράφει μοναδικά μια εγγραφή στον πίνακα (primary key). Προκύπτει από την ακολουθία tags\_seq όπου για κάθε νέα εγγραφή στον πίνακα αυξάνεται κατά ένα.

**USERID** Το πεδίο αυτό συγκρατεί το id του χρήστη που προσέδωσε την ετικέτα και υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα LABUSER (πρέπει δηλαδή να υπάρχει αντίστοιχη εγγραφή στον πίνακα αυτό).

**TAGID** Το πεδίο αυτό συγκρατεί το id της ετικέτας που αποδόθηκε στην δημοσίευση. Υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα TAGS.

### Πίνακας PUBLICATIONTAGS



Ο πίνακας που συγκρατά την πληροφορία για τις ετικέτες που αποδόθηκαν από όλους τους χρήστες της κοινότητας στις δημοσιεύσεις. Η πληροφορία αυτή θα μπορούσε να προκύψει με συνδέσμους (join) από τους πίνακες USERTAGS, TAGS, PUBOBJECT ωστόσο λόγω του ότι αυτή η πληροφορία προσπελαύνεται σε μεγάλη συχνότητα, για λόγους απόδοσης κρατείται και σε αυτόν τον πίνακα.

**Πεδία:**

- ID** Το πεδίο που περιγράφει μοναδικά μια εγγραφή στον πίνακα (primary key). Προκύπτει από την ακολουθία publicationtags\_seq όπου για κάθε νέα εγγραφή στον πίνακα αυξάνεται κατά ένα.
- PUBID** Το πεδίο αυτό συγκρατεί το id της δημοσίευσης στην οποία αποδόθηκε η ετικέτα και υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα PUBOBJECT (πρέπει δηλαδή να υπάρχει αντίστοιχη εγγραφή στον πίνακα αυτό).
- TAGID** Το πεδίο αυτό συγκρατεί το id της ετικέτας που αποδόθηκε στην συγκεκριμένη δημοσίευση. Υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα TAGS.

**Πίνακας PUBOBJECT**

Ο πίνακας που περιγράφει την οντότητα δημοσίευση

**Πεδία:**

- PUBID** Το πεδίο που περιγράφει μοναδικά μια εγγραφή στον πίνακα (primary key). Προκύπτει από την ακολουθία publications\_seq όπου για κάθε νέα εγγραφή στον πίνακα αυξάνεται κατά ένα.
- BIBTEX** Το πεδίο που συγκρατεί το bibTex της δημοσίευσης. Είναι τύπου varchar2 και έχει μήκος 1000 χαρακτήρες.
- LINK** Σε περίπτωση που ο χρήστης δεν επισύναψε αρχείο μαζί με την δημοσίευση αλλά τον υπερσύνδεσμο (link) που μπορεί να ανακτηθεί, η πληροφορία συγκρατείται σε αυτό το πεδίο. Σε αντίθετη περίπτωση τίθεται σε null.
- FILENAME** Σε περίπτωση που ο χρήστης επισύναψε αρχείο μαζί με την δημοσίευση τότε το όνομα του αρχείου αποθηκεύεται σε αυτό το πεδίο. Σε αντίθετη περίπτωση τίθεται σε null.
- BLOB\_OBJECT** Σε περίπτωση που ο χρήστης επισύναψε αρχείο μαζί με την δημοσίευση τότε το περιεχόμενο του αρχείου αποθηκεύεται σε αυτό το πεδίο. Είναι τύπου BLOB (Binary Large Object). Σε αντίθετη περίπτωση τίθεται ως null.

### Πίνακας USERUPLOADS

Ο πίνακας που συγκρατά την πληροφορία για τις δημοσιεύσεις που ανέβασαν-χαρακτήρισαν οι χρήστες της κοινότητας. Ο σχεσιακός αυτός πίνακας αναλαμβάνει να υλοποιήσει την σχέση στο διάγραμμα ER annotates (που ενώνει την οντότητα User με Publication).

#### **Πεδία:**

- ID** Το πεδίο που περιγράφει μοναδικά μια εγγραφή στον πίνακα (primary key). Προκύπτει από την ακολουθία useruploads\_seq όπου για κάθε νέα εγγραφή στον πίνακα αυξάνεται κατά ένα.
- USERID** Το πεδίο αυτό συγκρατεί το id του χρήστη που ανέβασε την δημοσίευση και υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα LABUSER (πρέπει δηλαδή να υπάρχει αντίστοιχη εγγραφή στον πίνακα αυτό).
- PUBID** Το πεδίο αυτό συγκρατεί το id της δημοσίευσης και υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα PUBOBJECT.
- UPLOADDATE** Το πεδίο συγκρατεί την πληροφορία της χρονικής στιγμής που έγινε η δημοσίευση. Είναι τύπου TIMESTAMP.

### Πίνακας PUBINFO

Ο πίνακας που συγκρατά πληροφορία που περιγράφει τις δημοσιεύσεις που ανέβηκαν από τους χρήστες. Η πληροφορία αυτή θα μπορούσε να προκύψει από το πεδίο bibTex του πίνακα PUBOBJECT (με χρήση της κλάσης bibTex η οποία κάνει parsing και εξάγει τις συγκεκριμένες πληροφορίες που αποθηκεύονται εδώ) ή με χρήση του λογικού στρώματος της εφαρμογής (του fuzzy-DL-Reasoner δηλαδή), ωστόσο λόγω του ότι η πληροφορία αυτή προσπελαύνεται σε μεγάλη συχνότητα, για λόγους απόδοσης κρατείται και σε αυτόν τον πίνακα. Είναι σημαντικό να πούμε ότι η πληροφορία αυτή χρησιμοποιείται μόνο στις περιπτώσεις που ανακτείται για σκοπούς παρουσίασης. Στις περιπτώσεις όπου γίνεται αναζήτηση για πληροφορία τότε χρησιμοποιείται το λογικό στρώμα του συστήματος όπου γίνεται αναζήτηση με βάση συνδετικών ερωτημάτων στον reasoner.

#### **Πεδία:**

- ID** Το πεδίο που περιγράφει μοναδικά μια εγγραφή στον πίνακα (primary key). Προκύπτει από την ακολουθία pubinfo\_seq όπου για κάθε νέα εγγραφή στον πίνακα αυξάνεται κατά ένα.
- PUBID** Το πεδίο αυτό συγκρατεί το id της δημοσίευσης και υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα PUBOBJECT.

<b>KEY</b>	Το πεδίο που συγκρατεί το κλειδί της δημοσίευσης (bibTex key).
<b>NAME</b>	Το πεδίο που συγκρατεί τον τίτλο της δημοσίευσης
<b>AUTHOR</b>	Το πεδίο που συγκρατεί τον/τους συγγραφέα/εις της δημοσίευσης.
<b>PUBLISHER</b>	Το πεδίο που συγκρατεί τον εκδότη της δημοσίευσης.
<b>TYPE</b>	Το πεδίο που συγκρατεί τον τύπο της δημοσίευσης (π.χ. article)
<b>YEAR</b>	Το πεδίο που συγκρατεί το έτος που εκδόθηκε η δημοσίευση.

### Πίνακας RATING

Ο πίνακας που χρησιμοποιείται για να συγκρατήσει τις αξιολογήσεις που κάνουν οι χρήστες του συστήματος σε δημοσιεύσεις. Ο σχεσιακός πίνακας αυτός υλοποιεί την σχέση του ER διαγράμματος rates (ενώνει τις οντότητες USER και PUBLICATION). Η μέση αξιολόγηση της κάθε δημοσίευσης βγαίνει με χρήση της συναθροιστικής συνάρτησης AVG.

#### **Πεδία:**

- ID** Το πεδίο που περιγράφει μοναδικά μια εγγραφή στον πίνακα (primary key). Προκύπτει από την ακολουθία rating\_seq όπου για κάθε νέα εγγραφή στον πίνακα αυξάνεται κατά ένα.
- USERID** Στο πεδίο αυτό υπάρχει το id του χρήστη που έκανε την αξιολόγηση. Υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα LABUSER.
- PUBID** Στο πεδίο αυτό υπάρχει το id της δημοσίευσης στην οποία έγινε αξιολόγηση. Υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα PUBOBJECT.
- RATE** Στο πεδίο υπάρχει ο βαθμός που έβαλε ο συγκεκριμένος χρήστης στην συγκεκριμένη δημοσίευση.

### Πίνακας KEYWORDS

Στον πίνακα αυτό αποθηκεύονται οι ερευνητικές περιοχές που υπάρχουν στο σύστημα. Ο πίνακας αυτός θα μπορούσε να παραλειφθεί και η πληροφορία αυτή να ανακτείται από το κατώτερο στρώμα της εφαρμογής (πρόσβαση στις αντίστοιχες οντότητες της οντολογίας μέσω του reasoner) ωστόσο υπάρχει μια τεχνική δυσκολία: το ΣΔΒΔ της Oracle δεν μπορεί να δημιουργήσει πίνακα με μήκος ονόματος πέραν των 30 χαρακτήρων οπότεν για τις οντότητες που υπάρχουν στην οντολογία οι οποίες ξεπερνούν αυτό το όριο δεν δημιουργείται/ούνται οι αντίστοιχοι σχεσιακοί πίνακες τους οποίους χρησιμοποιεί ο reasoner για αποθήκευση των instances του ABOX. Για τον λόγο αυτό υπήρξε πρόνοια να κρατηθούν όλα τα ονομάτα των concepts στην οντολογία κάτω από 30 χαρακτήρες με συντομογράφιση του ονόματος τους (π.χ. ConStrPerAnalADesA αντί Control structure performance analysis

and design aids). Η αντιστοίχιση του πραγματικού ονόματος και της συντομογράφησης του συγκρατείται σε αυτό τον πίνακα. Επίσης συγκρατούνται οι λέξεις κλειδιά που υπάρχουν στο 4<sup>ο</sup> επίπεδο της ιεραρχίας της ACM και αντιστοιχούνται στις ερευνητικές περιοχές της οντολογίας (στην οντολογία υλοποιήθηκε η ιεραρχία της ACM μέχρι και το 3<sup>ο</sup> επίπεδο).

#### **Πεδία:**

**ID** Το πεδίο που περιγράφει μοναδικά μια εγγραφή στον πίνακα (primary key). Προκύπτει από την ακολουθία rating\_seq όπου για κάθε νέα εγγραφή στον πίνακα αυξάνεται κατά ένα.

**NAME** Το όνομα της ερευνητικής περιοχής όπως περιγράφεται στην ιεραρχία της ACM (π.χ. 'Control structure performance analysis and design aids').

**INFO** Στο πεδίο συγκρατείται το πλήρες όνομα της ερευνητικής περιοχής μαζί με τα ονομάτα των ε.π. των προηγούμενων επιπέδων (π.χ. 'Hardware / Control structures and microprogramming / Control structure performance analysis and design aids'). Η πληροφορία και πάλι θα μπορούσε να έλθει με χρήση του reasoner στο πιο κάτω επίπεδο της εφαρμογής ωστόσο επειδή ο πίνακας αυτός χρησιμοποιείται από την ajax εφαρμογή για το autocomplete feature που υπάρχει κρίθηκε απαραίτητο να κρατηθεί αυτή η πληροφορία.

**TABLE\_BINDING** Το όνομα της ερευνητικής περιοχής όπως συντομογραφήθηκε στην οντολογία (π.χ. 'ConStrPerAnalADesA').

#### **Πίνακας ISINTERINRA**

Κάθε φορά που χαρακτηρίζεται μια καινούργια δημοσίευση σε κάποιες ερευνητικές περιοχές με κάποιο βαθμό, ο ίδιος βαθμός χρησιμοποιείται για να σχηματίσει τον βαθμό στο ρόλο isInterestedIn του συγγραφέα με την ερευνητική περιοχή λαμβάνοντας υπόψιν όμως πόσες άλλες φορές έχει χαρακτηριστεί ο συγκεκριμένος συγγραφέας. Συγκεκριμένα ο καινούργιος βαθμός  $\mu'$  υπολογίζεται ως εξής:

$$\mu' = \frac{\sum_{i=1}^N d_i + d_{N+1}}{N+1} = \frac{N \times \mu + d_{N+1}}{N+1}$$

όπου  $d_i$  ο βαθμός της  $i$  δημοσίευσης του συγγραφέα,

$N$  ο αριθμός των δημοσιεύσεων του συγκεκριμένου συγγραφέα στο σύστημα,

$\mu$  ο τρέχων μέσος όρος του συγγραφέα και ο  $\mu'$  ο καινούργιος υπολογισμένος μέσος όρος του συγγραφέα. Επειδή στην οντολογία του λογικού στρώματος δεν υπάρχει πρόνοια (και δεν θα μπορούσε να υπάρχει) να συγκρατείται η πληροφορία πόσες φορές έχει χαρακτηριστεί ένας συγγραφέας ότι ενδιαφέρεται σε μια ερευνητική περιοχή (ρόλος isInterestedIn) για σκοπούς υπολογισμού της καινούργιας τιμής η πληροφορία κρατείται σε αυτόν τον πίνακα.

**Πεδία:**

**AUTHOR** Στο πεδίο αυτό υπάρχει το όνομα του συγγραφέα που αναφερόμαστε

**RA** Στο πεδίο αυτό υπάρχει το id της ερευνητικής περιοχής στην οποία αναφερόμαστε. Υπάρχει περιορισμός ξένου κλειδιού με τον πίνακα keywords.

**USERSFREQUENCY** Στο πεδίο αυτό συγκρατείται το πλήθος των χαρακτηρισμών δημοσιεύσεων όπου συγγραφέας είναι ο συγκεκριμένος. Κάθε φορά που χαρακτηρίζεται καινούργια δημοσίευση με τον συγκεκριμένο συγγραφέα το πεδίο αυτό αυξάνεται κατά ένα (N).

**CURRENTDEGREE** Στο πεδίο αυτό υπάρχει η τρέχουσα τιμή του βαθμού του ρόλου isInterestedIn ( $\mu$ ).

**Πίνακας MARKUP\_TABLE**

Ο πίνακας αυτός χρησιμοποιείται σε συνεργασία με τον πίνακα HIGH\_TAB για προσωρινή αποθήκευση σε περίπτωση αναζήτησης με λέξη κλειδί στο περιεχόμενο των εγγράφων που είναι αποθηκευμένα στην βάση δεδομένων. Δεν υλοποιεί κάποια οντότητα ή σχέση του μοντέλου οντότητας-σχέσης αλλά δρα ως βοηθητικός (και προσωρινός) πίνακας.

**Πεδία:**

**ID** Το πεδίο αυτό συγκρατεί το id της δημοσίευσης που ανήκει στο σύνολο των αποτελεσμάτων της αναζήτησης.

**DOCUMENT** Εδώ φυλάσσεται υπό τη μορφή CLOB (Character Large Object) μέρος του αρχείου των δημοσιεύσεων που ανήκουν στο σύνολο των αποτελεσμάτων της αναζήτησης.

**Πίνακας HIGH\_TAB**

Ο πίνακας αυτός χρησιμοποιείται για προσωρινή αποθήκευση σε περίπτωση αναζήτησης με λέξη κλειδί στο περιεχόμενο των εγγράφων που είναι αποθηκευμένα στην βάση δεδομένων.

**Πεδία:**

- ID** Το πεδίο αυτό συγκρατεί το id της δημοσίευσης που ανήκει στο σύνολο των αποτελεσμάτων της αναζήτησης.
- OFFSET** Εδώ υπάρχει η θέση (πόσους χαρακτήρες από την αρχή του κειμένου) της λέξης κλειδί στο κείμενο που συγκρατείται στο πεδίο DOCUMENT του πίνακα MARKUP\_TABLE
- LENGTH** Εδώ συγκρατείται το μήκος του κειμένου που συγκρατείται στο πεδίο DOCUMENT του πίνακα MARKUP\_TABLE (που περιέχει τη λέξη κλειδί).

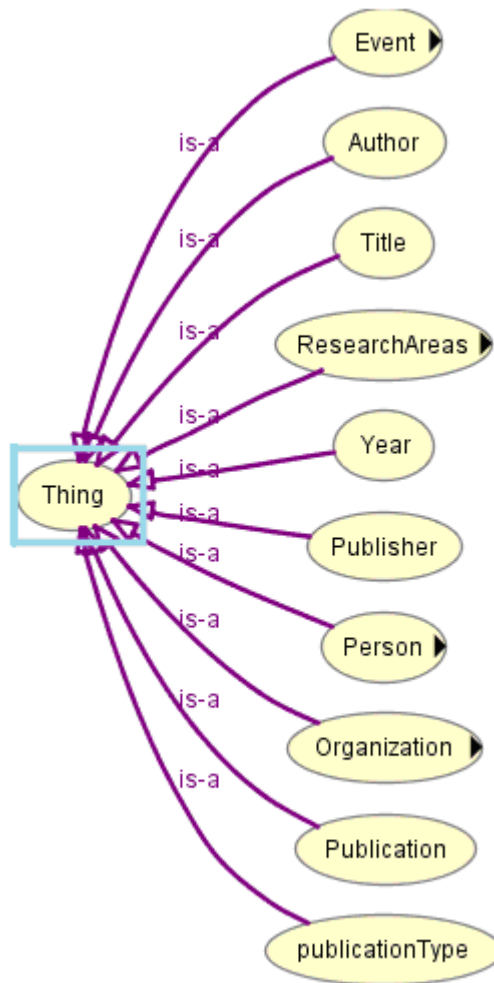
## ***4.5 Οντολογία αναπαράστασης του πεδίου της ακαδημαϊκής κοινότητας***

Στην ενότητα που ακολουθεί θα περιγραφούν οι γενικές αρχές σχεδίασης μιας Οντολογίας οι οποίες εφαρμόστηκαν κατά τη δημιουργία της Οντολογίας που περιγράφει το πεδίο των ακαδημαϊκών δημοσιεύσεων. Ακόμη θα περιγραφεί λεκτικά αλλά και διαγραμματικά και η ίδια η Οντολογία πάνω στην οποία στηρίζεται η υλοποίηση του συστήματος.

Στην Οντολογία που δημιουργήσαμε επιλέξαμε μια ιεραρχική δομή για τις έννοιες που θα αναπαρασταθούν. Καθώς η οντολογία δημιουργήθηκε με χρήση του εργαλείου σχεδίασης οντολογιών protégé – δηλαδή η σύνταξη έγινε σε γλώσσα OWL/RDF- λάβαμε ιδιαίτερη πρόνοια να μην υπερβούμε την εκφραστική ικανότητα της DL-Lite (και της fuzzy-DL-Lite γενικότερα) κάτι που θα δημιουργούσε προβλήματα αργότερα κατά την δημιουργία του αντίστοιχου σχεσιακού σχήματος από τον fuzzy-DL-Lite reasoner.

### ***4.5.1 Ανάλυση της οντολογίας***

Η οντολογία αποτελείται από 11 βασικές κλάσεις-έννοιες: Publication, PublicationType, Author, Publisher, Title, Person, Journal, Organization, Event, Year και ResearchAreas. Οι κλάσεις αυτές είναι υποκλάσεις της κλάσης owl:Thing η οποία είναι η καθολική έννοια (top-concept).



Ακολουθεί η παρουσίαση κάθε βασικής έννοιας ξεχωριστά:

**Event:**



Η έννοια περιγράφει κάθε συμβάν-γεγονός που αφορά ακαδημαϊκή δραστηριότητα.

Ως υποκλάσεις έχει τις έννοιες

**Activity:** περιγράφει μια γενική δραστηριότητα της επιστημονικής κοινότητας. Αυτή μπορεί να είναι π.χ. διάλεξη, εργαστηριακή παρακολούθηση, συνάντηση κτλ.

**Conference:** περιγράφει την έννοια του ακαδημαϊκού συνεδρίου

**Meeting:** περιγράφει την έννοια της συνάντησης μιας επιστημονικής ομάδας

**SpecialIssueEvent:** περιγράφει ένα ειδικό γεγονός που οργανώθηκε από την επιστημονική κοινότητα.

**Workshop:** περιγράφει μια εργαστηριακή συνάντηση που συνήθως λαμβάνει χώρα ως παράπλευρη δραστηριότητα σε συνέδριο.

Η έννοια είναι ξένη έννοια (disjoint concepts) με τις υπόλοιπες έννοιες.

Υπάρχουν οι ρόλοι **organizes** και ο αντίστροφος του **isOrganizedBy** που συνδέουν την έννοια **Event** με την έννοια **Organization**.

#### **Journal:**

Περιγράφει την έννοια του επιστημονικού περιοδικού. Ως επιστημονικό περιοδικό ορίζεται η περιοδική έκδοση πολλαπλών ακαδημαϊκών δημοσιεύσεων με σκοπό την ενημέρωση της ακαδημαϊκής κοινότητας για τις τρέχουσες εξελίξεις στον εκάστοτε επιστημονικό κλάδο.

Η έννοια είναι ξένη έννοια (disjoint concept) με τις υπόλοιπες έννοιες.

Υπάρχουν οι ρόλοι **isJournalOf** και (ο ανάστροφος) **hasJournal** που συσχετίζουν την έννοια **Journal** με την έννοια **Publication**.

#### **Author:**

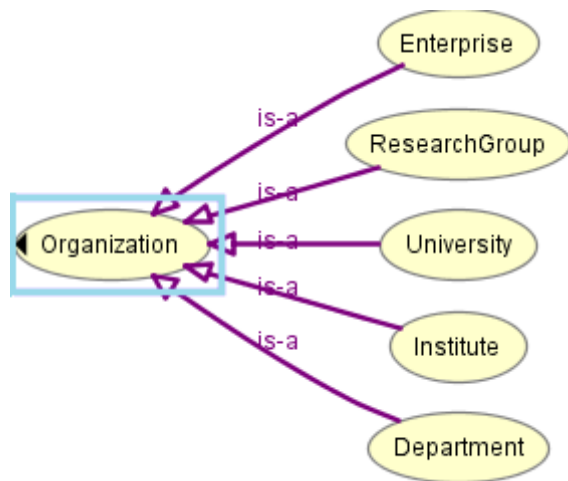
Η έννοια που περιγράφει τον συγγραφέα της εκάστοτε δημοσίευσης.

Η έννοια είναι ξένη έννοια (disjoint concept) με τις υπόλοιπες έννοιες πλην της έννοιας **Person**.

Υπάρχουν οι ρόλοι **isAuthorOf** και (ο ανάστροφος) **hasAuthor** που συσχετίζουν την έννοια **Author** με την έννοια **Publication**. Επίσης υπάρχουν οι ρόλοι **isInterestedIn** και (ο ανάστροφος) **hasauthorsInterestedIn** που συσχετίζουν την έννοια **Author** με την έννοια **ResearchAreas**.

#### **Organization:**





Η έννοια που περιγράφει ένα οργανισμό που ασχολείται με έκδοση επιστημονικών δημοσιεύσεων. Ένας οργανισμός μπορεί να είναι **University**(πανεπιστήμιο), **Department**(ακαδημαϊκό τμήμα), **Enterprise**(εταιρικός οργανισμός), **Institute** (ινστιτούτο), **ResearchGroup** (ανεξάρτητη ερευνητική ομάδα)

Η έννοια είναι ξένη έννοια (disjoint concept) με τις υπόλοιπες έννοιες.

Υπάρχουν οι ρόλοι **organizes** και ο αντίστροφος του **isOrganizedBy** που συνδέουν την έννοια **Organization** με την έννοια **Event** όπως επίσης και οι ρόλοι **worksFor** και ο αντίστροφος του **employs** που συνδέουν την έννοια με τις έννοιες **Author, Person**.

### Title:

Η έννοια που αφορά τον τίτλο μιας δημοσίευσης. Η έννοια είναι ξένη έννοια (disjoint concept) με όλες τις υπόλοιπες έννοιες της οντολογίας. Υπάρχουν οι ρόλοι **isTitleOf** και ο αντίστοιχος αντίστροφος **hasTitle** που συνδέει την έννοια με την έννοια **Publication**.

### Type:

Η έννοια που αφορά τον τύπο μιας δημοσίευσης. Παρόλο που δεν υπάρχει περιορισμός μέσω της οντολογίας για ποιού μπορεί να είναι αυτοί οι τύποι αυτοί περιορίζονται -μέσω του ανώτερου στρώματος- στο κλειστό σύνολο τύπων δημοσιεύσεων που ορίζονται στο bibTex δηλαδή το σύνολο ??έννοια είναι ξένη έννοια (disjoint concept) με όλες τις υπόλοιπες έννοιες της οντολογίας. Υπάρχουν οι ρόλοι **isTypeOf** και ο αντίστοιχος αντίστροφος **hasType** που συνδέει την έννοια με την έννοια **Publication**.

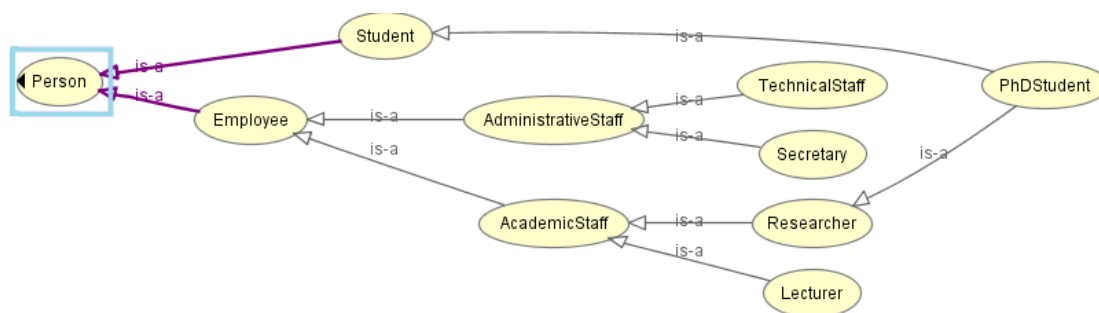
### Year

Η έννοια που αφορά το έτος έκδοσης μιας επιστημονικής δημοσίευσης. Παρόλο που δεν υπάρχει περιορισμός από την οντολογία (λόγω του ότι η εκφραστικότητα της DL-Lite δεν επιτρέπει την εισαγωγή τύπου δεδομένων) οι ισχυρισμοί που ανήκουν σε αυτή την έννοια είναι τύπου ακεραίου. Η έννοια είναι ξένη έννοια (disjoint concept) με όλες τις υπόλοιπες έννοιες της οντολογίας. Υπάρχουν οι ρόλοι **isYearOf** και ο αντίστοιχος αντίστροφος **hasYear** που συνδέει την έννοια με την έννοια **Publication**.

### **Publisher:**

Η έννοια που αφορά τον εκδότη μιας επιστημονικής δημοσίευσης. Η έννοια είναι ξένη έννοια (disjoint concept) με όλες τις υπόλοιπες έννοιες της οντολογίας. Υπάρχουν οι ρόλοι **isPublisherOf** και ο αντίστοιχος αντίστροφος **hasPublisher** που συνδέει την έννοια με την έννοια **Publication** και **Journal**.

### **Person:**



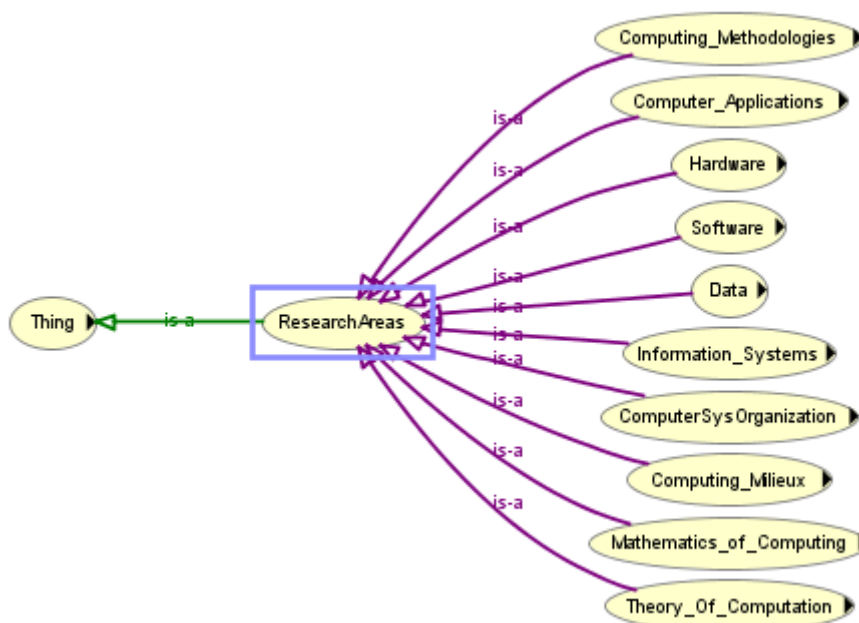
Η έννοια που περιγράφει ένα φυσικό πρόσωπο. Ένα πρόσωπο μπορεί να ανήκει στην έννοια **Employee** (προσωπικό) ή **Student** (φοιτητής). Η έννοια **Employee** επεκτείνεται περαιτέρω και έχει υποκλάσεις τις έννοιες **AcademicStaff** και **AdministrativeStaff**. Η έννοια **AcademicStaff** αφορά το ακαδημαϊκό προσωπικό και μπορεί να είναι **Lecturer** ή **Researcher** (ερευνητής). Η έννοια του ερευνητή περιλαμβάνει την έννοια του διδακτορικού φοιτητή (**PhDStudent**). Η έννοια **AdministrativeStaff** περιλαμβάνει το διοικητικό προσωπικό η έννοια του οποίου μπορεί να ειδικευτεί περαιτέρω εισάγοντας την έννοια **Secretary** και **TechnicalStaff**. Η έννοια του φοιτητή μπορεί να ειδικευτεί περαιτέρω με την εισαγωγή της έννοιας του διδακτορικού φοιτητή (**PhDStudent**). Παρατηρούμε ότι η συγκριμένη έννοια είναι υπόκλιση δυο άλλων εννοιών ταυτόχρονα (πολλαπλή κληρονομικότητα). Η έννοια **Person** είναι ξένη έννοια (disjoint concept) με όλες τις υπόλοιπες έννοιες της οντολογίας πλην της έννοιας **Author**. Υπάρχουν οι ρόλοι **worksFor**

και ο αντίστροφος του **employs** που συνδέουν την έννοια με την έννοια **Organization** και ο ρόλος **isa** που συνδέει την έννοια με την έννοια **Author**.

### Publication:

Η έννοια που αφορά τις επιστημονικές δημοσιεύσεις. Η έννοια είναι ξένη έννοια (disjoint concept) με όλες τις υπόλοιπες έννοιες της οντολογίας. Υπάρχουν οι ρόλοι **hasPublisher** και ο αντίστοιχος αντίστροφος **isPublisherOf** που συνδέει την έννοια με την έννοια **Publisher**, οι ρόλοι **hasYear** (αντ. **isYearof**) που συνδέει την έννοια με την έννοια **Year**, ο ρόλος **hasType** (**isTypeOf**) που συνδέει την έννοια με την έννοια **Type**, **hasTitle** (αντ.: **isTitleOf**) που συνδέει την έννοια με την έννοια **Title**, **hasAuthor** (αντ.: **isAuthorOf**) που συνδέει την έννοια με την έννοια **Author**, **hasJournal** (αντ.: **isJournalOf**) που συνδέει την έννοια με την έννοια **Journal**, οι ρόλοι **isOnTheArea** (αντ.: **hasPubOn**) που συνδέει την έννοια με την έννοια **ResearchAreas**.

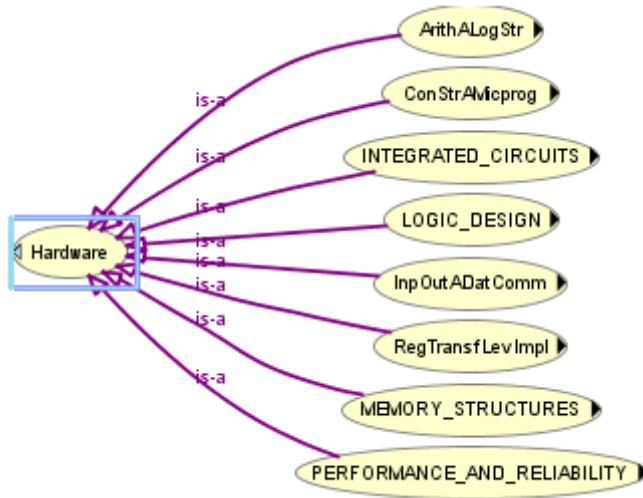
### ResearchAreas:



Η έννοια που ενθυλακώνει όλες τις ερευνητικές περιοχές που αφορούν την ειδικότητα του ηλεκτρολόγου μηχανικού και μηχανικού υπολογιστών. Η έννοια οργανώνεται και σχηματίζει μια ιεραρχία από 3 επίπεδα όπως ακριβώς αυτή έχει οριστεί επίσημα από

την ACM (Association for Computing Machinery)<sup>1</sup>. Στο πρώτο επίπεδο συναντούμε τις ερευνητικές περιοχές:

- **Hardware**



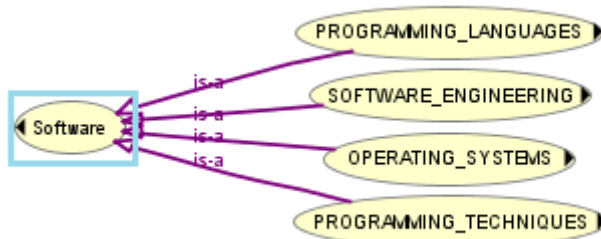
- **Computer Systems Organization**

---

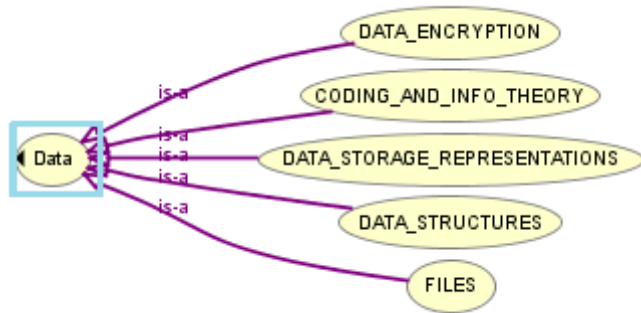
<sup>1</sup> <http://www.acm.org/>



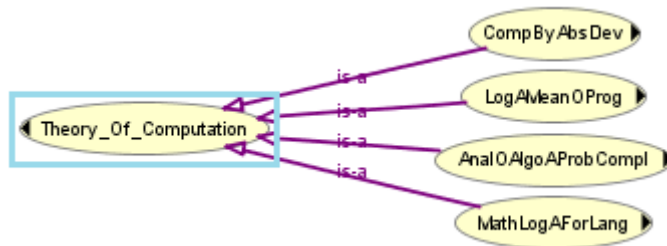
- **Software**



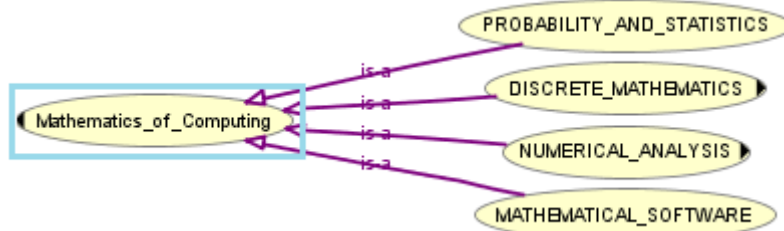
- **Data**



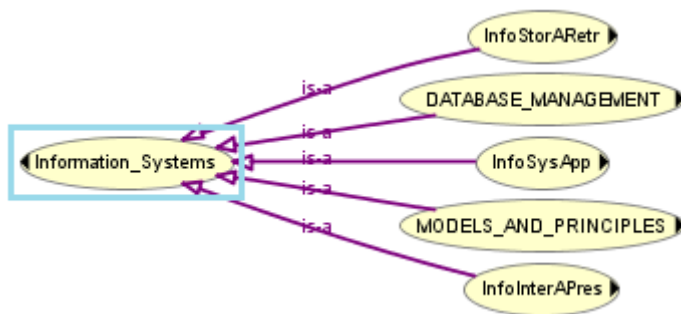
- **Theory of Computation**



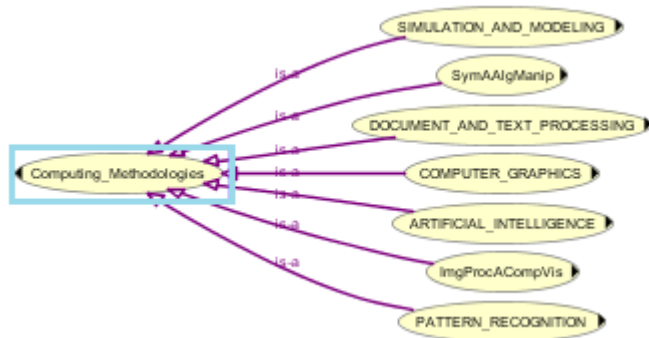
- **Mathematics of Computing**



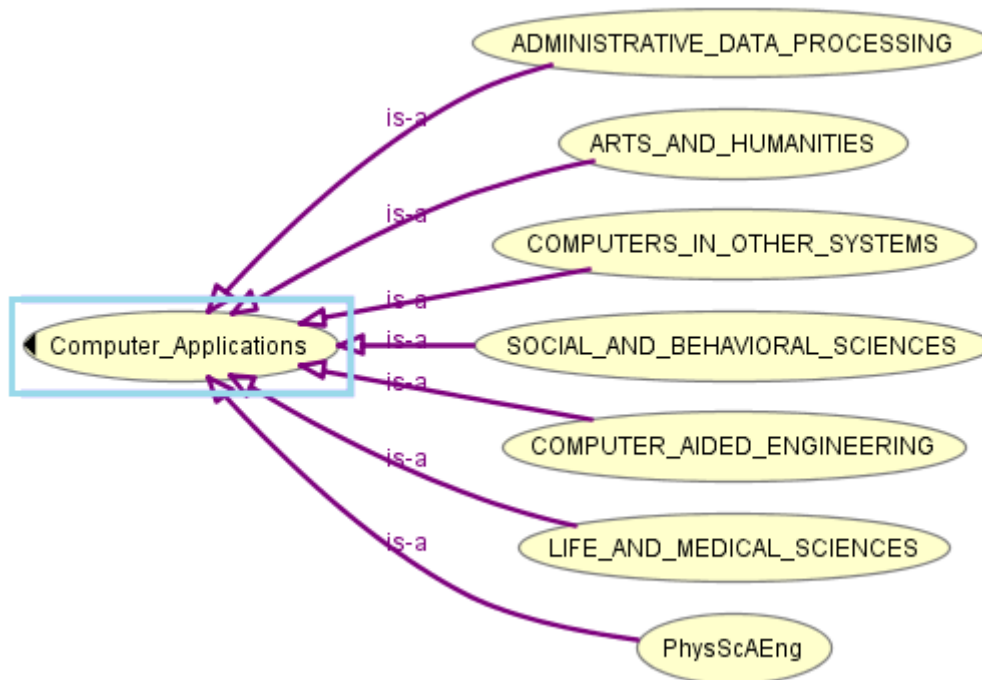
- **InformationSystem**



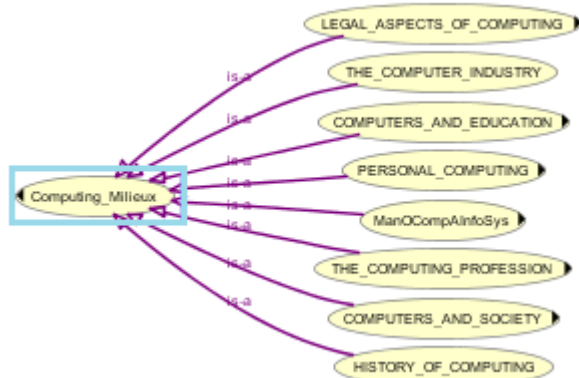
- **Computing Methodologies**



- **Computer Applications**



- **Computing Milleux**



Στη πλειονότητα των διαγραμμάτων που παρουσιάζονται εδώ παρουσιάζεται μέχρι και το 2<sup>ο</sup> επίπεδο των ερευνητικών περιοχών. Η παράλειψη του 3<sup>ου</sup> επιπέδου των ερευνητικών περιοχών έγινε για λόγους οικονομίας χώρου. Η πλήρης περιγραφή της ιεραρχίας μπορεί να

βρεθεί στην αντίστοιχη σελίδα της ACM.<sup>1</sup> Για τα ονόματα των εννοιών που υπάγονται στην γενικότερη έννοια του ResearchAreas χρησιμοποιήθηκαν συντομογραφίες αντί της πλήρους ονομασίας λόγω των τεχνικών δυσκολιών που περιγράφονται στην ενότητα 4.4

Η έννοια **ResearchAreas** είναι ξένη έννοια (disjoint concept) με όλες τις υπόλοιπες έννοιες της οντολογίας. Υπάρχουν οι ρόλοι **hasAuthorsInterestedIn** και ο αντίστροφος του **isInterestedIn** που συνδέουν την έννοια με την έννοια **Author** καθώς και οι ρόλοι **hasThePublication** και ο αντίστροφος του **isOnTheArea** που συνδέουν την έννοια με την έννοια **Publication**.

Παρατηρούμε ότι όλοι οι ρόλοι και έννοιες -πλην των ρόλων **isInterestedIn** και **isOnTheArea**-δεν αξιοποιούν την ασαφή εκφραστικότητα που προσφέρει η fuzzy DL Lite καθώς η εισαγωγή ισχυρισμών στο σώμα ισχυρισμών της οντολογίας γίνεται πάντα σε βαθμό 1. Το γεγονός αυτό είναι επακόλουθο του γεγονότος ότι γενικά στις έννοιες που αφορούν την ακαδημαϊκή κοινότητα (πλην των ρόλων που αναφέραμε) δεν υπεισέρχεται αβεβαιότητα ή ασάφεια: π.χ. δεν μπορεί να εισαχθεί βαθμός αληθείας στο κατά πόσο ένας συγγραφέας είναι πράγματι συγγραφέας μιας δημοσίευσης, ο συγγραφέας είτε θα είναι είτε δεν θα είναι ο δημιουργός μιας δημοσίευσης. Ωστόσο οι ρόλοι **isInterestedIn** και **isOnTheArea** είναι εν γενεί ασαφείς συσχετισμοί και επομένως η χρήση της ασαφούς DL-Lite κρίνεται επιβεβλημένη.

#### 4.5.2 Παράδειγμα εισαγωγής δημοσίευσης

Παρακάτω δίνουμε ένα παράδειγμα του πως σχηματίζεται η ερμηνεία του κόσμου που μοντελοποιούμε μετά την εισαγωγή μιας δημοσίευσης. Έστω ότι στο σύστημα εισάγεται η δημοσίευση που περιγράφεται από το ακόλουθο bibTex:

```
@article{horrocks2003sar,  
  title={{From SHIQ and RDF to OWL: The making of a web ontology language}},  
  author={Horrocks, I. and Patel-Schneider, P.F. and Van Harmelen, F.},  
  journal={Web semantics: science, services and agents on the World Wide Web},  
  year={2003},  
  publisher={Elsevier}  
}
```

---

<sup>1</sup> <http://www.acm.org/about/class/1998/>



Έστω ακόμη ότι ο χρήστης που εισάγει την συγκεκριμένη δημοσίευση καθορίζει ότι αυτή ανήκει στην περιοχή του Knowledge Representation κατά 0.9 και στην περιοχή του Deduction and Theorem Proving κατά 0.7. Η ενέργεια αυτή συνεπάγεται την δημιουργία των ισχυρισμών:

**Publication(horrocks2003sar) ≥ 1.0**

**Year(2003) ≥ 1.0,**

**isYearOf(2003,horrocks2003sar) ≥ 1.0,**

**hasYear(horrocks2003sar,2003) ≥ 1.0**

**Title(From SHIQ and RDF to OWL: The making of a web ontology language) ≥ 1.0,**

**isTitleOf(From SHIQ and RDF to OWL: The making of a web ontology language, horrocks2003sar) ≥ 1.0**

**hasTitle(horrocks2003sar, From SHIQ and RDF to OWL: The making of a web ontology language, horrocks2003sar) ≥ 1.0**

**Journal(Web semantics: science, services and agents on the World Wide Web) ≥ 1.0,**

**hasJournal(horrocks2003sar, Web semantics: science, services and agents on the World Wide Web) ≥ 1.0,**

**isJournalOf(Web semantics: science, services and agents on the World Wide Web, horrocks2003sar) ≥ 1.0**

**Publisher(Elsevier) ≥ 1.0**

**isPublisherOf(Elsevier, horrocks2003sar) ≥ 1.0**

**hasPublisher(horrocks2003sar, Elsevier) ≥ 1.0**

Για κάθε ένα συγγραφέα δημιουργία ισχυρισμών Author, hasAuthor, isAuthorOf

**Author(Horrocks, I.) ≥ 1.0,**

**hasAuthor(horrocks2003sar, Horrocks, I.) ≥ 1.0**

**isAuthorOf(Horrocks, I. ,horrocks2003sar) ≥ 1.0**

**Author(Patel-Schneider) ≥ 1.0**

**hasAuthor(horrocks2003sar, Patel-Schneider) ≥ 1.0**

**isAuthorOf(Patel-Schneider ,horrocks2003sar) ≥ 1.0**

**Author(Van Harmelen, F.) ≥ 1.0**

**hasAuthor(horrocks2003sar, Van Harmelen, F.) ≥ 1.0**

**isAuthorOf(Van Harmelen, F. ,horrocks2003sar) ≥ 1.0**

Δημιουργία ασαφών ισχυρισμών:

**isOnThisArea(horrocks2003sar, Knowledge Representation) ≥ 0.9**

**hasPubsOn(Knowledge Representation, horrocks2003sar)≥0.9**

**isOnThisArea(horrocks2003sar, Deduction and theorem proving)≥0.7**

**hasPubsOn(Deduction and theorem proving , horrocks2003sar)≥0.7**

Για κάθε Author αντίστοιχα

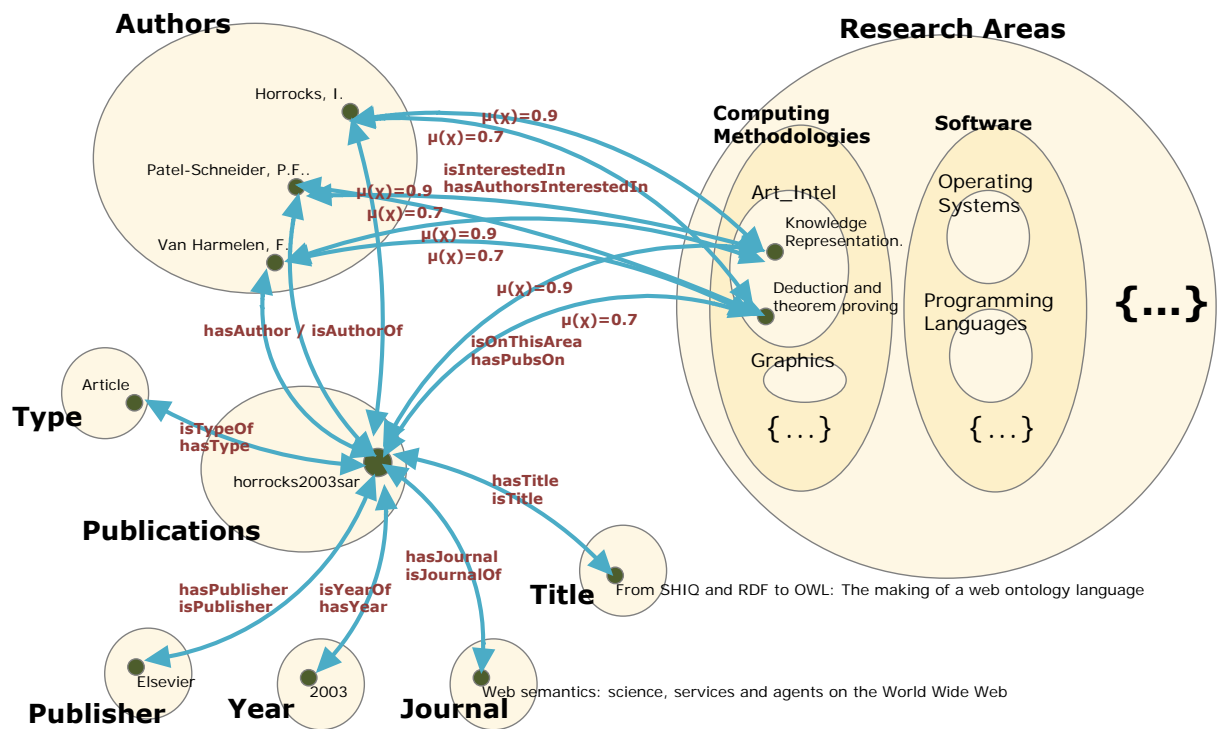
**isInterestedIn(Horrocks, I., Knowledge Representation)≥0.9**

**hasAuthorsInterestedIn (Knowledge Representation, Horrocks, I.)≥0.9**

**isInterestedIn(Horrocks, I., Knowledge Representation)≥0.9**

**hasAuthorsInterestedIn (Knowledge Representation, Horrocks, I.)≥0.9**

Στο διάγραμμα που ακολουθεί βρίσκεται μια γραφική αναπαράσταση του κόσμου που περιγράφει η οντολογία μετά την εισαγωγή της δημοσίευσης




# 5

## *Υλοποίηση*

Εδώ συζητούμε λεπτομερώς θέματα υλοποίησης του συστήματος.

### *5.1 Περιγραφή κλάσεων του συστήματος*

### 5.1.1 Κλάση *bibTex*

 <b>bibTex</b>
<p style="text-align: center;"><i>Attributes</i></p> <pre>private String entryType private String title private String authors private String journal private String publisher private String year private String month private String key private String volume private String edition private String address private String editor private String howPub private String school private String organization private String institution private String type private String number private String pages private String note private String series private String url private String crossref private String booktitle</pre>
<p style="text-align: center;"><i>Operations</i></p> <pre>public bibTex( ) public bibTex( String rawText ) public boolean isValidBibTex( String given ) public boolean isValidEntry( String entry ) public boolean isValidEntryType( String bType ) public String getCrossref( ) public void setCrossref( String crossref ) public String toString( ) public String getUrl( ) public void setUrl( String url )     { ... } public void setYear( String year ) public String getAddress( )     { ... } public String getYear( ) public void setEntryType( String e ) public String getEntryType( ) public String getBooktitle( ) public void setBooktitle( String booktitle )</pre>

### Εικόνα 5.1.1 Διάγραμμα κλάσης bibTex

Η κλάση αυτή συγκρατεί τα πεδία-οντότητες που χαρακτηρίζουν το bibTex μιας δημοσίευσης. Χρησιμοποιείται ως entity bean για τα bibTex αλλά και ως parser για να εξάγει από δεδομένη συμβολοσειρά (String) το bibTex.

Πακέτο: gr.ntua.ivml.bibTex

Πεδία :

```
private String entryType;  
private String title;  
private String authors;  
private String journal;  
private String publisher;  
private String year;  
private String month;  
private String key;  
private String volume;  
private String edition;  
private String address;  
private String editor;  
private String howPub;  
private String school;  
private String organization;  
private String institution;  
private String type;  
private String number;  
private String pages;  
private String note;  
private String series;  
private String url;  
private String crossref;  
private String booktitle;
```

Μέθοδοι και λειτουργίες

```
public bibTex(String rawText) throws Exception
```

Η μέθοδος αυτή αποτελεί τον κατασκευαστή και επιχειρεί να εκτελέσει συντακτική ανάλυση σε δεδομένη είσοδο String και να αναθέσει τιμές στα πεδία του bibTex (όσα πεδία δεν υπάρχουν παραμένουν null). Σε περίπτωση που το string είναι συντακτικά λάθος ή υπάρξει άλλο πρόβλημα εγείρεται εξαίρεση και η διαδικασία διακόπτεται.

```
public boolean isValidBibTex(String given) throws  
ArrayIndexOutOfBoundsException, Exception
```

Η μέθοδος ελέγχει την εγκυρότητα ενός δεδομένου String που αναπαριστά bibTex.

```
private boolean isValidEntry(String entry)
```

Η μέθοδος αυτή χρησιμοποιείται από το τις μεθόδους του κατασκευαστή και isValidBibTex για να ελέγξει αν μια δεδομένη οντότητα είναι έγκυρη οντότητα δηλαδή ανήκει στο σύνολο {address, annote, author, booktitle, chapter, crossref, edition, editor, eprint, howpublished,

institution, journal, key, month, number, organization, pages, publisher, school, series, title, type, url, volume, year}.

```
private boolean isValidEntryType(String bType)
```

η μέθοδος αυτή ελέγχει αν το πεδίο entryType του bibTex είναι έγκυρο ανήκει δηλαδή στο σύνολο {article,book,booklet, conference, inbook, incollection, inproceedings, manual, mastersthesis, misc, phdthesis, proceedings, techreport, unpublished}

```
public String toString()
```

Η μέθοδος επιστρέφει σε συμβολοσειρά αναπαράσταση του bibTex σύμφωνα με τις οντότητες που συγκρατεί το συγκεκριμένο αντικείμενο. Η αναπαράσταση είναι της μορφής:

```
@EntryType{[key],  
    entry1={entry1Value},  
    entry2={entry2Value},  
    ...  
    entryN={entryNValue}  
}
```

Οι υπόλοιπες μέθοδοι αποτελούν απλώς τους getters (τετριμμένες μέθοδοι που επιστρέφουν ένα συγκεκριμένο πεδίο) και setters (τετριμμένες μέθοδοι που θέτουν τιμή σε ένα συγκεκριμένο πεδίο) και απλώς παρατίθενται για σκοπούς πληρότητας.

```
public String getCrossref()
```

```
public void setCrossref(String crossref)
```

```
public String getUrl()
```

```
public void setUrl(String url)
```

```
public void setAddress(String address)
```

```
public void setAuthors(String authors)
```

```
public void setEdition(String edition)
```

```
public void setEditor(String editor)
```

```
public void setHowPub(String howPub)
```

```
public void setInstitution(String institution)
```

```
public void setJournal(String journal)
```

```
public void setKey(String key)
```

```
public void setMonth(String month)
```

```
public void setNote(String note)
```

```
public void setNumber(String number)
public void setOrganization(String organization)
public void setPages(String pages)
public void setPublisher(String publisher)
public void setSchool(String school)
public void setSeries(String series)
public void setTitle(String title)
public void setType(String type)
public void setVolume(String volume)
public void setYear(String year)
public String getAddress()
public String getAuthors()
public String getEdition()
public String getEditor()
public String getHowPub()
public String getInstitution()
public String getJournal()
public String getKey()
public String getMonth()
public String getNote()
public String getNumber()
public String getOrganization()
public String getPages()
public String getPublisher()
public String getSchool()
public String getSeries()
public String getTitle()
public String getType()
public String getVolume()
public String getYear()
public void setEntryType(String e)
public String getEntryType()
```

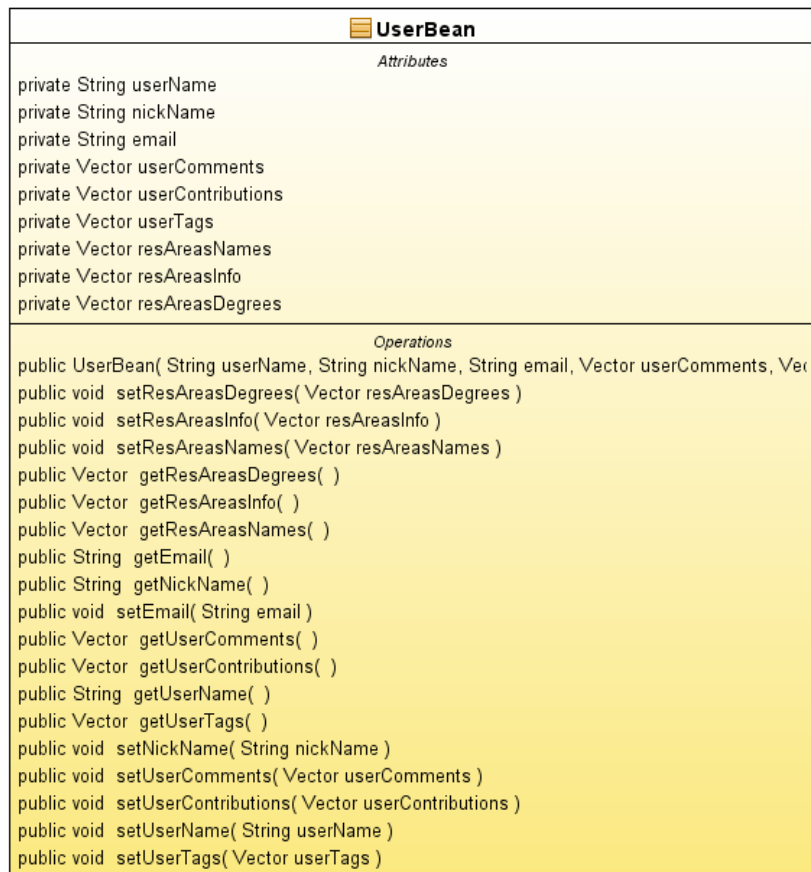
```

public String getBooktitle()

public void setBooktitle(String booktitle)

```

### 5.1.2 Κλάση UserBean



Εικόνα 5.1.2 Διάγραμμα κλάσης UserBean

Η κλάση αυτή κατακρατεί τα στοιχεία ενός χρήστη του συστήματος. Χρησιμοποιείται από το στρώμα του εξυπηρετητή για να αποθηκεύσει προσωρινά πληροφορίες για ένα χρήστη οι οποίες προήλθαν από την βάση δεδομένων και με διάδραση με το κατώτερο στρώμα της εφαρμογής.

**Πακέτο:** gr.ntua.ivml.site.EntitiesBeans

**Πεδία :**

```

private String userName;
private String nickName;
private String email;
private Vector userComments;
private Vector userContributions;
private Vector userTags;
private Vector resAreasNames;
private Vector resAreasInfo;
private Vector resAreasDegrees;

```



Μέθοδοι / Λειτουργίες :

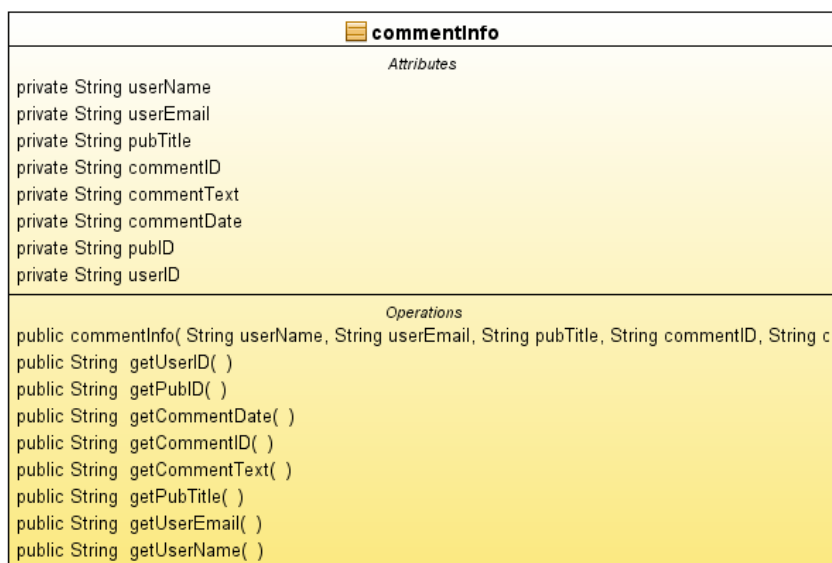
```
Public UserBean(String userName, String nickName, String email,  
Vector userComments, Vector userContributions, Vector userTags,  
Vector resAreasNames, Vector resAreasInfo, Vector resAreasDegrees)
```

Η μέθοδος αποτελεί τον κατασκευαστή της κλάσης και αναλαμβάνει να αναθέσει τιμές στα πεδία της κλάσης (όσα πεδία δεν υπάρχουν παραμένουν null) και να δημιουργήσει ένα καινούργιο αντικείμενο.

Οι υπόλοιπες μέθοδοι αποτελούν απλώς τους getters (τετριμμένες μέθοδοι που επιστρέφουν ένα συγκεκριμένο πεδίο) και setters (τετριμμένες μέθοδοι που θέτουν τιμή σε ένα συγκεκριμένο πεδίο) και απλώς παρατίθενται για σκοπούς πληρότητας.

```
public void setResAreasDegrees(Vector resAreasDegrees)  
public void setResAreasInfo(Vector resAreasInfo)  
public void setResAreasNames(Vector resAreasNames)  
public Vector getResAreasDegrees()  
public Vector getResAreasInfo()  
public Vector getResAreasNames()  
public String getEmail()  
public String getNickName()  
public void setEmail(String email)  
public Vector getUserComments()  
public Vector getUserContributions()  
public String getUsername()  
public Vector getUserTags()  
public void setNickName(String nickName)  
public void setUserComments(Vector userComments)  
public void setUserContributions(Vector userContributions)  
public void setUsername(String userName)  
public void setUserTags(Vector userTags)
```

### 5.1.3 Κλάση *commentInfo*



Εικόνα 5.1.3 Διάγραμμα κλάσης *commentInfo*

Η κλάση αυτή κατακρατεί τα στοιχεία για ένα σχόλιο που έγινε στο σύστημα. Χρησιμοποιείται από το στρώμα του εξυπηρετητή για να αποθηκεύσει προσωρινά την πληροφορία για ένα σχόλιο η οποία προήλθε από την βάση δεδομένων και με διάδραση με το κατώτερο στρώμα της εφαρμογής. Συγκεκριμένα χρησιμοποιείται από την `home.jsp` για να παρουσιάσει τα τελευταία χρονολογικά σχόλια (καλώντας την συνάρτηση `getLatestComments()` της κλάσης `dbConnection` η οποία επιστρέφει ένα διάνυμα από στιγμιότυπα της κλάσης `commentInfo`) ή στην `profile.jsp` για να παρουσιάσει το σύνολο των σχολίων του χρήστη (καλώντας αντίστοιχα την `getUserComments(String user)`).

**Πακέτο:** `gr.ntua.ivml.site.EntitiesBeans`

**Πεδία:**

```
private String userName;  
private String userEmail;  
private String pubTitle;  
private String commentID;  
private String commentText;  
private String commentDate;  
private String pubID;  
private String userID;
```

**Μέθοδοι / Λειτουργίες:**

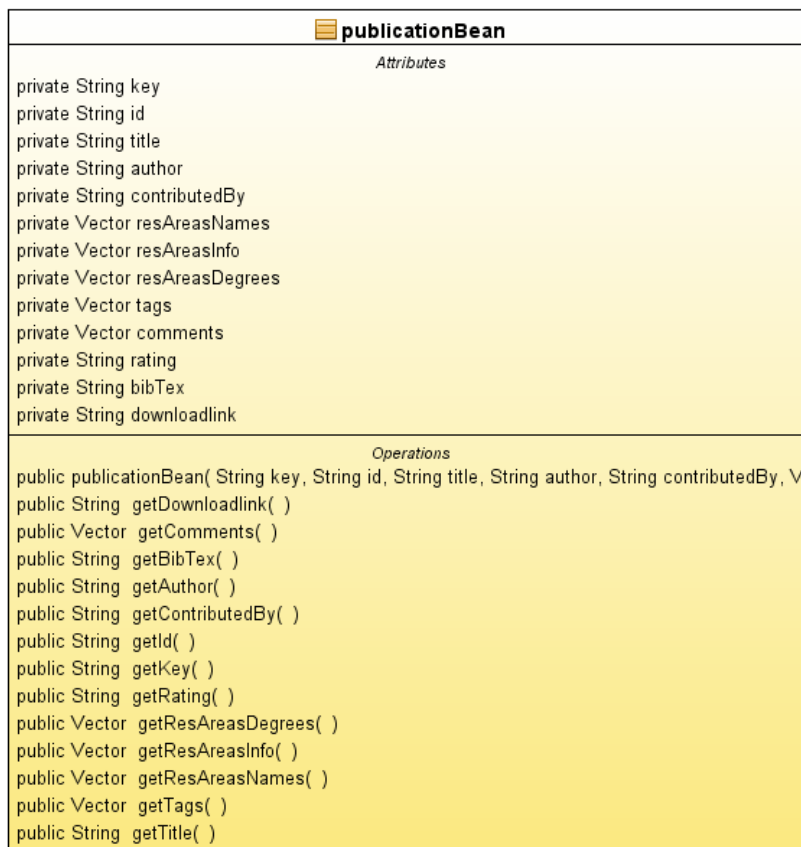
```
public commentInfo(String userName, String userEmail, String  
pubTitle, String commentID, String commentText, String commentDate,  
String pubID,String userID)
```

Η μέθοδος αποτελεί τον κατασκευαστή της κλάσης και αναλαμβάνει να αναθέσει τιμές στα πεδία της κλάσης (όσα πεδία δεν υπάρχουν παραμένουν null) και να δημιουργήσει ένα καινούργιο αντικείμενο.

Οι υπόλοιπες μέθοδοι αποτελούν απλώς τους getters (τετριμμένες μέθοδοι που επιστρέφουν ένα συγκεκριμένο πεδίο) και απλώς παρατίθενται για σκοπούς πληρότητας.

```
public String getUserID()  
public String getPubID()  
public String getCommentDate()  
public String getCommentID()  
public String getCommentText()  
public String getPubTitle()  
public String getUserEmail()  
public String getUsername()
```

#### 5.1.4 Κλάση *publicationBean*



Εικόνα 5.1.4 Διάγραμμα κλάσης *publicationBean*

Η κλάση αυτή κατακρατεί τα στοιχεία για μια δημοσίευση που υπάρχει στο σύστημα. Χρησιμοποιείται από το στρώμα του εξυπηρετητή για να αποθηκεύσει προσωρινά την πληροφορία για μια δημοσίευση η οποία προήλθε από την βάση δεδομένων και με διάδραση με το κατώτερο στρώμα της εφαρμογής. Συγκεκριμένα χρησιμοποιείται από

μα σειρά σελίδων που παρουσιάζουν πληροφορία περί δημοσιεύσεων. Ο λόγος ύπαρξης μιας τέτοιας κλάσης είναι για να ευκολύνει την διάδραση με τα κατώτερα στρώματα και την διαδικασία παρουσίασης της συγκεκριμένης πληροφορίας στο στρώμα της εφαρμογής.

**Πακέτο:** `gr.ntua.ivml.site.EntitiesBeans`

#### **Πεδία:**

```
private String key;  
private String id;  
private String title;  
private String author;  
private String contributedBy;  
private Vector resAreasNames;  
private Vector resAreasInfo;  
private Vector resAreasDegrees;  
private Vector tags;  
private Vector comments;  
private String rating;  
private String bibTex;  
private String downloadlink;
```

#### **Μέθοδοι / Λειτουργίες:**

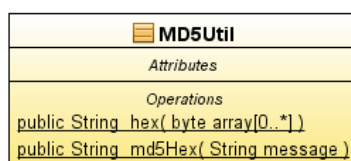
```
public publicationBean(String key, String id, String title, String  
author, String contributedBy, Vector resAreasNames, Vector  
resAreasInfo, Vector resAreasDegrees, Vector tags, String  
rating, String bibtex, Vector comments, String dlink)
```

Η μέθοδος αποτελεί τον κατασκευαστή της κλάσης και αναλαμβάνει να αναθέσει τιμές στα πεδία της κλάσης (όσα πεδία δεν υπάρχουν παραμένουν null) και να δημιουργήσει ένα καινούργιο αντικείμενο.

Οι υπόλοιπες μέθοδοι αποτελούν απλώς τους getters (τετριμμένες μέθοδοι που επιστρέφουν ένα συγκεκριμένο πεδίο) και απλώς παρατίθενται για σκοπούς πληρότητας.

```
public String getDownloadlink()  
public Vector getComments()  
public String getBibTex()  
public String getAuthor()  
public String getContributedBy()  
public String getId()  
public String getKey()  
public String getRating()  
public Vector getResAreasDegrees()  
public Vector getResAreasInfo()  
public Vector getResAreasNames()  
public Vector getTags()  
public String getTitle()
```

### 5.1.5 Κλάση MD5Util



Εικόνα 5.1.5 Διάγραμμα κλάσης MD5Util

Η κλάση προσφέρει την δυνατότητα να κρυπτογραφηθεί μια δεδομένη συμβολοσειρά με το αλγόριθμο κρυπτογράφησης MD5. Η λειτουργία της κρυπτογράφησης χρησιμοποιείται για λόγους ασφαλείας και ιδιωτικότητας των χρηστών. Παραδείγματος χάρι οι κωδικοί εισόδου των χρηστών αποθηκεύονται στη βάση κρυπτογραφημένοι καθώς αποτελούν ευαίσθητο προσωπικό στοιχείο του κάθε χρήστη.

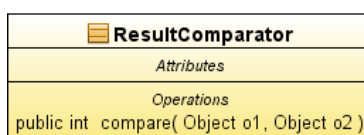
**Πακέτο:** gr.ntua.ivml.site.Utilities

**Μέθοδοι / Λειτουργίες:**

```
public static String md5Hex(String message)
```

Η μέθοδος επιστρέφει κρυπτογραφημένο κείμενο από δωθέν κείμενο σύμφωνα με τον αλγόριθμο κρυπτογράφησης MD5.

### 5.1.6 Κλάση ResultComparator



Εικόνα 5.1.6 Διάγραμμα κλάσης ResultComparator

Η κλάση αποτελεί μια διαπροσωπεία (interface) και βασικά υλοποιεί την μέθοδο compare ενός συγκριτή. Σκοπός που η συγκεκριμένη μέθοδος υπερφορτώνεται (overloaded) είναι ότι τα αποτελέσματα σε επερώτηση ασαφής συνάθροισης (fuzzy aggregation query) στον reasoner είναι σε μορφή από αταξινόμητα ζευγάρια αντικείμενο και βαθμός αληθείας (και όχι απλώς βαθμός), έτσι δεν μπορεί να γίνει χρήση της εσωτερικής συνάρτησης της Java για ταξινόμηση. Ωστόσο με χρήση της κλάσης ResultComparator μπορούμε εύκολα να ταξινομήσουμε τα αποτελέσματα ενός top-k ερωτήματος γράφοντας απλώς:

```
Collections.sort(queryResults, new ResultComparator());
```

όπου queryResults τα αποτελέσματα της ερώτησης στη μορφή λίστας από ζευγάρια αντικείμενο και βαθμός αληθείας.

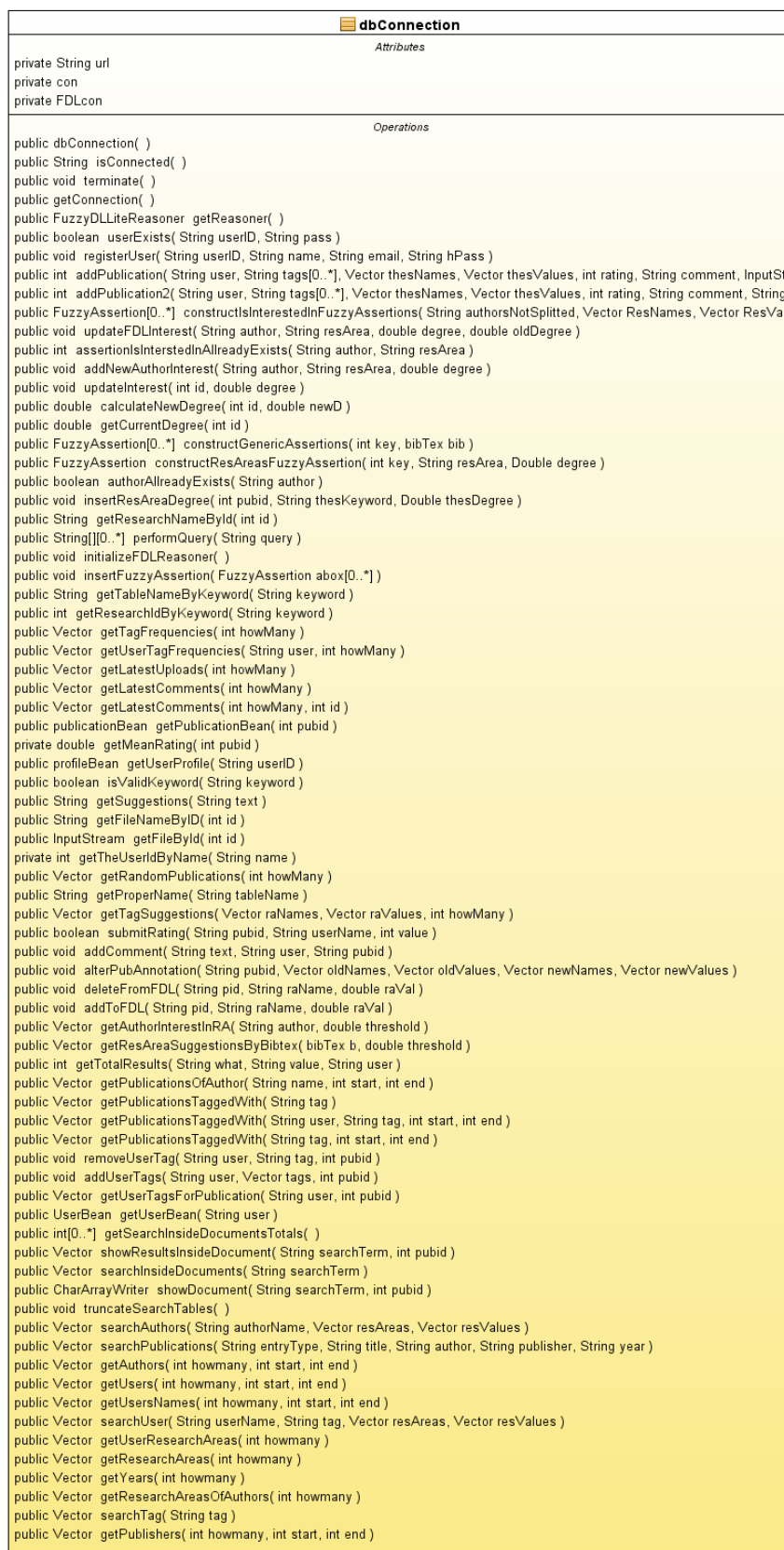
**Πακέτο:** gr.ntua.ivml.site.Utilities

**Μέθοδοι / Λειτουργίες:**

**public int** compare(Object o1, Object o2)

Η μέθοδος εκτελεί σύγκριση μεταξύ δυο αντικειμένων o1,o2 και επιστρέφει -1 εαν το o1 είναι μεγαλύτερο από το o2 και 1 εαν το o1 είναι μικρότερο από το o2.

## 5.1.7 Κλάση dbConnection



Εικόνα 5.1.7 Διάγραμμα κλάσης dbConnection

Η κλάση αυτή αποτελεί την πιο βασική μονάδα διαλειτουργικότητας του στρώματος εφαρμογής με τα κατώτερα στρώματα διαλειτουργικότητας. Συγκεκριμένα προσφέρει επικοινωνία με την (σχεσιακή) βάση δεδομένων ενώ συγκεντρώνει ένα σύνολο από μεθόδους για επίκληση (invoke) και χρήση του fuzzy-DL-Lite reasoner. Κάθε φορά που ένας χρήστης ενώνεται στο σύστημα δημιουργείται ένα στιγμιότυπο (instance) της κλάσης το οποίο και αποθηκεύεται στην μνήμη του εξυπηρετητή ενώ το χαρακτηριστικό id (jsessionId) του αντικειμένου αυτού αποθηκεύεται ως cookie στον χρήστη-πελάτη. Έτσι καθόλη την διάρκεια της συνεδρίας ενός χρήστη δημιουργείται μόνο ένα στιγμιότυπο αυτού του αντικειμένου το οποίο και κάθε φορά που ο χρήστης απαιτεί την διάδραση με το κατώτερο στρώμα της εφαρμογής ανακτείται με την βοήθεια του jsessionid. Αυτό έγινε για σκοπούς απόδοσης του συστήματος ούτως ώστε να μην απαιτείται κάθε φορά που ο χρήστης θα εκκινά μια καινούργια δραστηριότητα να αρχικοποιείται ένα καινούργιο αντικείμενο και ακολούθως να αποδεσμεύεται με αποτέλεσμα την κατακράτηση αρκετών πόρων του συστήματος.

**Πακέτο:** gr.ntua.ivml.site.Utilities

Πεδία:

Τα βασικά πεδία της κλάσης είναι τα δυο αντικείμενα επικοινωνίας με τα δυο σχήματα της βάσης δεδομένων (ένα το σχεσιακό και ένα που χρησιμοποιείται από τον reasoner) καθώς και το αντικείμενο του reasoner. Υπάρχουν και άλλα βοηθητικά πεδία τα οποία μπορούν να παραμετροποιηθούν ανάλογα με το περιβάλλον εγκατάστασης όπως είναι τα usernames/passwords των χρηστών των σχημάτων της βάσης δεδομένων, το URI της τοποθεσίας που βρίσκεται το .owl αρχείο που περιγράφει την οντολογία πάνω στην οποία βασίζεται το σύστημα καθώς και την τοποθεσία που είναι εγκατεστημένο το σύστημα (host), η πόρτα επικοινωνίας (port) και υπηρεσία επικοινωνίας (service).

```
private String host  
private String port  
private String service  
private String username  
private String username2  
private String password  
private String url  
private String ontologyFileLocation  
private Connection con  
private Connection FDLcon  
private FuzzyDLLiteReasoner dLiteReasoner
```

**Μέθοδοι / Λειτουργίες:**



**public void** initializeFDLReasoner()

Η μέθοδος αυτή αναλαμβάνει να φορτώσει το .owl αρχείο στο σύστημα, να εκτελέσει συντακτική ανάλυση σε αυτό (parsing) ούτως ώστε να εξάγει το σώμα ορολογίας (Tbox) και αφού δημιουργήσει ένα νέο αντικείμενο FuzzyDLLiteReasoner να αναθέσει το Tbox που περιγράφεται στο αρχείο ως το Tbox του reasoner. Σε περίπτωση που δημιουργηθεί κάποιο πρόβλημα (π.χ. δεν βρέθηκε το .owl αρχείο ή η συντακτική ανάλυση απέτυχε) εγείρεται εξαίρεση και η διαδικασία τερματίζεται.

**public** dbConnection()

Αποτελεί τον κατασκευαστή της κλάσης. Συγκεκριμένα εδώ εγκαθιστούνται οι δυο επικοινωνίες (connections) με την βάση δεδομένων ενώ αρχικοποιείται και ο reasoner με κλήση της initializeFDLReasoner(). Σε περίπτωση που δημιουργηθεί πρόβλημα είτε στην εγκατάσταση της επικοινωνίας είτε στην αρχικοποίηση του reasoner εγείρεται εξαίρεση και η διαδικασία τερματίζεται.

**public** boolean isConnected()

Απλή μέθοδος που εκτελεί έλεγχο κατά πόσο υπάρχει επικοινωνία με τα δυο σχήματα της βάσης δεδομένων.

**public void** terminate()

Κατα την λήξη της συνεδρίας του χρήστη (logout.jsp) καλείται αυτή η μέθοδος, η οποία αναλαμβάνει να αποδεσμεύσει τους πόρους του συστήματος.

**public boolean** userExists(String userID, String pass)

Η μέθοδος καλείται κατά την σύνδεση του χρήστη για να ελέγξει την ύπαρξη των στοιχείων (όνομα και κωδικός του χρήστη) στην βάση δεδομένων.

**public void** registerUser(String userID, String name, String email, String hPass)

Η μέθοδος καλείται κατά την εγγραφή του χρήστη στο σύστημα και αναλαμβάνει να εγγράψει μια καινούργια πλειάδα (tuple) στον πίνακα labuser με τα στοιχεία του χρήστη.

**public int** addPublication(String user, String[] tags, Vector thesNames, Vector thesValues, **int** rating, String comment, InputStream is, String fileName, String key, String pubName, String bibTex, String pubAuthors)

Η μέθοδος αυτή καλείται κατά τον χαρακτηρισμό (annotation) μιας καινούργιας δημοσίευσης στο σύστημα η οποία συνοδεύεται και με αρχείο. Η μέθοδος αναλαμβάνει να εγγράψει τα στοιχεία στους κατάλληλους πίνακες του σχεσιακού

σχήματος (tags, usertags, comments, usercomments, pubobject, pubinfo, rating, userpuloads) ενώ με διαδοχική κλήση των `constructResAreasFuzzyAssertion()`, `constructGenericAssertions()` και `constructIsInterestedInFuzzyAssertions()` κατασκευάζει ένα διάνυσμα από ασαφείς ισχυρισμούς (FuzzyAssertions) το οποίο και εισάγει στο ABox της βάσης γνώσης καλώντας την `insertFuzzyAssertion()`. Όπως και στις προηγούμενες μεθόδους αν υπάρξει κάποιο πρόβλημα εγείρεται εξαίρεση και η διαδικασία τερματίζεται.

**public int** addPublication2(String user, String[] tags, Vector thesNames, Vector thesValues, **int** rating, String comment, String link, String key, String pubName, String bibTex, String pubAuthors)

Η μέθοδος αυτή όπως και η `addPublication` καλείται κατά τον χαρακτηρισμό μιας καινούργιας δημοσίευσης με μόνη διαφορά ότι δεν συνοδεύεται από αρχείο ο χαρακτηρισμός αυτός αλλά μόνο από τον υπερσύνδεσμο(link) της τοποθεσίας που μπορεί ο χρήστης να ανακτήσει το αρχείο. Εκτελεί ακριβώς τις ίδιες λειτουργίες όπως και η `addPublication` με μόνη διαφορά την εγγραφή στον (σχεσιακό) πίνακα PUBOBJECT.

**public** ArrayList<FuzzyAssertion> constructIsInterestedInFuzzyAssertions(String authorsNotSplitted, Vector ResNames, Vector ResValues)

Η μέθοδος αυτή καλείται κατά τον χαρακτηρισμό μιας καινούργιας δημοσίευσης από την `addPublication()` ή την `addPublication2()`. Ως ορίσματα παίρνει τους συγγραφείς και τις ερευνητικές περιοχές που αποδόθηκαν στην δημοσίευση. Αναλαμβάνει να δημιουργήσει για κάθε ένα συγγραφέα ( $Author_i$ ) και ερευνητική περιοχή ( $ResearchArea_j$ ) τον ασαφή ισχυρισμό

$isInterestedIn(Author_i, ResearchArea_j) \geq k_j$  όπου

$Author_i$  ο i-οστός συγγραφέας,

$ResearchArea_j$  η j-οστή ερευνητική περιοχή που αποδόθηκε στην δημοσίευση και

$k_j$  ο j-οστός βαθμός της συγκεκριμένης ερευνητικής περιοχής

Προτού δημιουργήσει τον εκάστοτε ισχυρισμό εκτελεί έλεγχο αν υπάρχει ήδη ο συγκεκριμένος ισχυρισμός στο σώμα ισχυρισμών της βάσης γνώσης (ABox) (καλώντας την `assertionIsInterestedInAlreadyExists()`) και αν ναι τότε υπολογίζει τον καινούργιο βαθμό του ισχυρισμού ως τον μέσο όρο των υφισταμένων βαθμών και ενημερώνει τον υφιστάμενο ισχυρισμό καλώντας την `updateFDLInterest()`.

Η μέθοδος αφού κατασκευάσει επιτυχώς τους ασαφείς αυτούς ισχυρισμούς τους επιστρέφει ως λίστα στην καλούσα μέθοδο.

**public int** assertionIsInterestedInAlreadyExists(String author, String resArea)  
 η μέθοδος καλείται από constructIsInterestedInFuzzyAssertions() για να ελέγξει αν στο σώμα ισχυρισμών (ABox) υπάρχει ήδη ισχυρισμός που αφορά τον δωθέντα συγγραφέα και ερευνητική περιοχή.

**public void** updateFDLInterest(String author, String resArea, double degree, double oldDegree)  
 Στην περίπτωση που στην constructIsInterestedInFuzzyAssertions() προηγήθηκε ο ασαφής ισχυρισμός απαιτείται η ενημέρωση του υφιστάμενου ισχυρισμού σύμφωνα με τα νέα δεδομένα. Ως ορίσματα η μέθοδος δέχεται την τριάδα που απαρτίζουν τον καινούργιο ασαφή ισχυρισμό (συγγραφέας, ερευνητική περιοχή, ενημερωμένος βαθμός) καθώς και τον παλιό βαθμό για να ανευρεθεί ο ισχυρισμός προς ενημέρωση.

**public double** calculateNewDegree(int id, double newD)  
 Η συνάρτηση που αναλαμβάνει να υπολογίσει τον καινούργιο βαθμό του ισχυρισμού isInterestedIn.  
 Ο νέος βαθμός ( $\mu'$ ) υπολογίζεται ως:

$$\mu' = \frac{\sum_{i=1}^N d_i + d_{N+1}}{N+1} = \frac{N \times \mu + d_{N+1}}{N+1}$$

όπου  $d_i$  ο βαθμός της  $i$  δημοσίευσης του συγγραφέα,

$N$  ο αριθμός των δημοσιεύσεων του συγκεκριμένου συγγραφέα στο σύστημα,

$\mu$  ο τρέχων μέσος όρος του συγγραφέα και ο  $\mu'$  ο καινούργιος υπολογισμένος μέσος όρος του συγγραφέα.

**public** ArrayList<FuzzyAssertion> constructGenericAssertions(int key, bibTex bib)  
 η μέθοδος αυτή αναλαμβάνει να κατασκευάσει τους υπόλοιπους γενικούς ισχυρισμούς που αφορούν μια καινούργια δημοσίευση που μπαίνει στην βάση γνώσης. Ως όρισμα δίνεται το κλειδί της δημοσίευσης και το bibTex της με βάση το οποίο θα εξαχθούν όλα τα πεδία που αφορούν την δημοσίευση. Συγκεκριμένα δημιουργούνται οι εξής ισχυρισμοί:

isTypeOf(key,type)  $\geq 1.0$  (όπου type={article,book,phdthesis,...})

Year(year)  $\geq 1.0$

isYearOf(key,year)  $\geq 1.0$

Title(title)  $\geq 1.0$

isTitleOf(key,title)  $\geq 1.0$

Publisher(publisher)  $\geq 1.0$

isPublisherOf(key,publisher)  $\geq 1.0$

Και για κάθε ένα συγγραφέα οι ισχυρισμοί:

Author(author<sub>*j*</sub>)  $\geq 1.0$

isAuthorOf(key,author<sub>*j*</sub>)  $\geq 1.0$

Αφού κατασκευαστεί το σύνολο των ισχυρισμών επιτυχώς επιστρέφονται ως λίστα στην καλούσα συνάρτηση.

**public** FuzzyAssertion constructResAreasFuzzyAssertion(**int** key, String resArea, Double degree)

Η μέθοδος αυτή καλείται διαδοχικά από την addPublication() (ή addPublication2() ) διαδοχικά για κάθε ερευνητική περιοχή που αποδώθηκε στην δημοσίευση με σκοπό να δημιουργήσει τον ασαφή ισχυρισμό:

$isOnThisArea(key, researchArea_i) \geq k_i$

**public void** insertFuzzyAssertion(ArrayList<FuzzyAssertion> abox)

Αφού κατασκευαστεί η πλήρης λίστα με τους ασαφείς ισχυρισμούς που αφορούν τον χαρακτηρισμό μιας καινούργιας δημοσίευσης καλείται η παρούσα μέθοδος που αναλαμβάνει να επικοινωνήσει με τον reasoner και να αποθηκεύσει το καινούργιο σύνολο ασαφών ισχυρισμών στο υφιστάμενο σώμα ισχυρισμών της βάσης γνώσης.

**public** String getTableNameByKeyword(String keyword)

Η μέθοδος επιστρέφει το (συντομογραφημένο) όνομα έννοιας που υπάρχει στην οντολογία σύμφωνα με την αντίστοιχη λέξη κλειδί.

**public int** getResearchIdByKeyword(String keyword)

Η μέθοδος επιστρέφει το id της ερευνητικής περιοχής από τον πίνακα KEYWORDS σύμφωνα με την αντίστοιχη λέξη κλειδί

**public** Vector getTagFrequencies(**int** howMany)

η μέθοδος επιστρέφει ένα διάνυσμα που περιέχει τις πιο συχνές ετικέτες (tags) μαζί με την συχνότητα τους. Οι πιο συχνές ετικέτες ανακτούνται με ομαδοποίηση (group by) στον πίνακα TAGS και χρήση της συναθροιστικής συνάρτησης COUNT.

**public** Vector getUserTagFrequencies(String user, **int** howMany)

όπως και η getTagFrequencies(int howMany) επιστρέφει διάνυσμα που περιέχει τις πιο συχνές ετικέτες και συχνότητα που αφορούν όμως ένα συγκεκριμένο χρήστη. Αυτό γίνεται πάλι με ομαδοποίηση αντί όμως στον πίνακα TAGS, στον πίνακα που προκύπτει από την σύνδεση (join) των πινάκων TAGS και USERTAGS.

**public** Vector getLatestUploads(**int** howMany)

Η μέθοδος επιστρέφει τις πιο πρόσφατες χρονολογικά συνεισφορές των χρηστών στο σύστημα υπό την μορφή διανύσματος από αντικείμενα uploadInfo. Καλείται από την home.jsp για να αναδείξει την πρόσφατη δραστηριότητα στο σύστημα.

**public** Vector getLatestComments(**int** howMany)

Η μέθοδος επιστρέφει τα πιο πρόσφατα χρονολογικά σχόλια των χρηστών στο σύστημα υπό την μορφή διανύσματος από αντικείμενα commentInfo. Όπως και η getLatestUploads καλείται από την home.jsp για να αναδείξει την πρόσφατη δραστηριότητα στο σύστημα.

**public** Vector getLatestComments(**int** howMany, **int** pubid)

Η μέθοδος επιστρέφει τα πιο πρόσφατα χρονολογικά σχόλια των χρηστών στο σύστημα υπό την μορφή διανύσματος από αντικείμενα commentInfo για μια συγκεκριμένη δημοσίευση. Καλείται από την getPublicationBean() για να αναδείξει το πιο πρόσφατο σχολιασμό από την κοινότητα των χρηστών σε μια συγκεκριμένη δημοσίευση.

**public** publicationBean getPublicationBean(**int** pubid)

Η μέθοδος καλείται από την `publication.jsp` και επιστρέφει τα πλήρη στοιχεία μια δημοσίευσης υπο την μορφή αντικειμένου της κλάσης `publicationBean`.

**private double** `getMeanRating(int pubid)`

Η συνάρτηση καλείται από την `getPublicationBean()` με σκοπό να ανακτηθεί ο μέσος όρος αξιολόγησης μιας δημοσίευσης από τους χρήστες. Αυτό γίνεται με χρήση της συναθροιστικής συνάρτησης `AVG` στον σχεσιακό πίνακα `RATING`.

**public boolean** `isValidKeyword(String keyword)`

Η μέθοδος καλείται ασύγχρονα κατά την διάρκεια του χαρακτηρισμού μιας καινούργιας δημοσίευσης και ελέγχει αν οι ερευνητικές περιοχές που έθεσε ο χρήστης ανήκουν στο κλειστό λεξιλόγιο (`closed vocabulary`) των ερευνητικών περιοχών που σχηματίζουν την ιεραρχία στην οντολογία. Σε περίπτωση που ο χρήστης αυθαίρετα έθεσε μια δική του ονομασία που δεν υπάρχει σε αυτό το κλειστό σύνολο επιστρέφεται λάθος και ενημερώνεται με σχετικό μήνυμα.

**public String** `getSuggestions(String text)`

Η μέθοδος καλείται ασύγχρονα κατά τον χαρακτηρισμό μιας καινούργιας δημοσίευσης για να βοηθήσει τον χρήστη να συμπληρώσει στα αντίστοιχα πεδία τις ερευνητικές περιοχές που αποδίδει στην δημοσίευση. Αφού εκτελεστεί ερώτημα στον πίνακα `KEYWORDS` για εύρεση παρόμοιων εγγραφών με το δωθέν κείμενο (συμβολοσειρά) επιστρέφεται το σύνολο αυτών των εγγραφών στη μορφή `XML`:

```
<results>
  <rs id="id1" info="info1">
    Name1
  </rs>
  <rs id="id2" info="info2">
    Name2
  </rs>
  ...
  <rs id="idN" info="infoN">
    NameN
  </rs>
</results>
```

Το `XML` αυτό επεξεργάζεται από την `JavaScript` και το φέρνει σε παρουσιάσιμη μορφή για να υποδείξει στον χρήστη τις ερευνητικές περιοχές που μπορεί να χρησιμοποιήσει.

**public InputStream** `getFileById(int id)`

Η συνάρτηση καλείται από την `download.jsp` και επιστρέφει σε μορφή συνεχούς ροής χαρακτήρων (`stream`) το αρχείο της δημοσίευσης.

**public String** `getFileNameById(int id)`

Αποτελεί παραφερής μέθοδο με την `getFileById`. Καλείται και αυτή από την `download.jsp` και επιστρέφει το όνομα του αρχείου με το οποίο αποθηκεύτηκε στο σύστημα το αρχείο.

**private int** `getTheUserIdByName(String name)`

Επιστρέφει το `id` του χρήστη σύμφωνα με το όνομα του από τον πίνακα `LABUSER`

**public Vector** `getRandomPublications(int howMany)`

Σε περίπτωση που ο χρήστης δεν έχει αναπτύξει δραστηριότητα στο σύστημα (δεν έχει ανεβάσει κάποια δημοσίευση) καλείται η μέθοδος αυτή που επιστρέφει τυχαία ένα αριθμό από δημοσιεύσεις στην μορφή διανύσματος από αντικείμενα `publicationBean` με σκοπό να τα προβάλει ως εισηγήσεις στον χρήστη.

**public** Vector getRecomendedPubs(**int** howMany, String user)

Σε περίπτωση που ο χρήστης έχει αναπτύξει δραστηριότητα στο σύστημα τότε οι εισηγήσεις μπορούν να είναι πιο στοχευμένες. Αφού βρεθεί η ερευνητική περιοχή που τον ενδιαφέρει πιο πολύ εκτελείται ερώτημα μέσω του `reasoner` για ανάκτηση δημοσιεύσεων που ανήκουν σε αυτή την ερευνητική περιοχή. Οι προτεινόμενες δημοσιεύσεις επιστρέφονται σε μορφή διανύσματος από αντικείμενα της κλάσης `publicationBean`. Η μέθοδος καλείται από την `home.jsp` (κεντρική σελίδα του συστήματος) και από την `publication.jsp`.

**public** Vector getTagSuggestions(Vector raNames, Vector raValues, **int** howMany)

Η μέθοδος καλείται κατά το τελευταίο βήμα χαρακτηρισμού μιας νέας δημοσίευσης (αφού έχουν δοθεί οι ερευνητικές περιοχές) και επιστρέφει ένα διάνυσμα από προτεινόμενες ετικέτες. Οι ετικέτες αυτές πορκύπτουν από τις πιο δημοφιλείς ετικέτες για τις συγκεκριμένες ερευνητικές περιοχές δηλαδή από την σύνδεση των πινάκων TAGS, PUBLICATIONRA και PUBLICATIONTAGS με συνάθροιση των βαθμών των ετικετών και κατάταξη τους σύμφωνα με αυτό τον βαθμό.

**public boolean** submitRating(String pubid, String userName, **int** value)

Η μέθοδος καλείται ασύγχρονα από την `publication.jsp` όταν ο χρήστης αξιολογεί μια δημοσίευση. Σε περίπτωση που ο χρήστης έχει ήδη αξιολογήσει την συγκεκριμένη δημοσίευση επιστρέφεται λάθος και εμφανίζεται σχετικό μήνυμα. Σε αντίθετη περίπτωση εγγράφεται μια νέα εγγραφή στον πίνακα RATING και επαναυπολογίζεται ο μέσος όρος αξιολόγησης μέσω της `getMeanRating()`.

**public void** addComment(String text, String user, String pubid)

Η μέθοδος προσθέτει ένα καινούργιο σχόλιο ενός χρήστη σε μια υφιστάμενη δημοσίευση. Συγκεκριμένα εγγράφει νέες εγγραφές στους πίνακες COMMENTS και USERCOMMENTS

**public void** alterPubAnnotation(String pubid, Vector oldNames, Vector oldValues, Vector newNames, Vector newValues)

η μέθοδος καλείται κατά την τροποποίηση του χαρακτηρισμού δημοσίευσης σε σχέση με τις ερευνητικές περιοχές. Αναλαμβάνει να διαγράψει τους υφιστάμενους ισχυρισμούς από την βάση γνώσης και να προσθέσει τους καινούργιους.

Συγκεκριμένα για κάθε υφιστάμενο ισχυρισμό καλείται η `deleteFromFDL()` και για κάθε καινούργιο καλείται η `addToFDL()`. Επιλέχθηκε η μέθοδος διαγραφή-επανεισαγωγή αντί της απλής ενημέρωσης γιατί δύναται ο αριθμός (και τα μέλη) των καινούργιων ισχυρισμών να διαφέρει από τον υφιστάμενο.

**public void** deleteFromFDL(String pid, String raName, **double** raVal)

Αναλαμβάνει να διαγράψει από το σώμα ισχυρισμών (ABox) τον υφιστάμενο

ισχυρισμό  $isOnThisArea(key, researchArea_i) \geq k_i$  για την συγκεκριμένη δημοσίευση, ερευνητική περιοχή και βαθμό συμμετοχής.



**public void** addToFDL(String pid, String raName, **double** raVal)

Αναλαμβάνει να προσθέσει στο σώμα ισχυρισμών (ABox) τον καινούργιο ισχυρισμό  $isOnThisArea(key, researchArea_i) \geq k_i$  για την συγκεκριμένη δημοσίευση, ερευνητική περιοχή και βαθμό συμμετοχής.

**public** Vector getAuthorInterestInRA(String author, **double** threshold)

Η μέθοδος καλείται από την author.jsp και αναλαμβάνει να βρει τις ερευνητικές περιοχές που ενδιαφέρουν τον συγκεκριμένο συγγραφέα.

Συγκεκριμένα εκτελεί ερώτηση ασαφής συνάθροισης (fuzzy aggregation query) μέσω του reasoner της μορφής:

$$q(x) \leftarrow Author(author) \geq 1.0 \wedge isInterestedIn(author, x) \geq th \wedge ResearchArea(x) \geq 1.0$$

Όπου author ο συγκεκριμένος συγγραφέας και th ένα κατώφλι στην περιοχή (0,1] που χρησιμοποιείται ως περιορισμός και βελτιστοποιητής του ερωτήματος. Τα αποτελέσματα επιστρέφονται στην μορφή διανύσματος από ζευγάρια της μορφής (ερευνητική περιοχή, βαθμός αληθείας).

**public** Vector getResAreaSuggestionsByBibtex(bibTex b, **double** threshold)

η μέθοδος καλείται στο δεύτερο βήμα του χαρακτηρισμού μιας καινούργιας δημοσίευσης (όπου έχει ήδη δοθεί το bibTex και συνεχώς οι συγγραφείς της δημοσίευσης) και αναλαμβάνει να βρει προτεινόμενες ερευνητικές περιοχές σύμφωνα με τους συγγραφείς της δημοσίευσης. Συγκεκριμένα εκτελεί μέσω του reasoner επερώτηση ασαφής συνάθροισης (fuzzy aggregation query) της μορφής:

$$q(x) \leftarrow ResearchAreas(x) \geq 1.0 \wedge isInterestedIn(author_1, x) \geq th \wedge \dots \wedge isInterestedIn(author_N, x) \geq th$$

Όπου  $author_i$  οι συγγραφείς που δωθήκαν σύμφωνα με το bibTex,  $1 \leq i \leq N$  και th ένα κατώφλι στην περιοχή (0,1] που χρησιμοποιείται ως περιορισμός και βελτιστοποιητής του ερωτήματος. Τα αποτελέσματα επιστρέφονται στην μορφή διανύσματος από ζευγάρια της μορφής (ερευνητική περιοχή, βαθμός αληθείας) και παρουσιάζονται στον χρήστη ως προτεινόμενες περιοχές με ένδειξη του βαθμού αληθείας.

**public int** isPubAlreadyUploaded(String title)

Η μέθοδος καλείται κατά το δεύτερο βήμα χαρακτηρισμού καινούργιας δημοσίευσης και εκτελεί έλεγχο κατά πόσο η δημοσίευση υπάρχει ήδη στην βάση γνώσης. Αν ναι τότε επιστρέφει το id της δημοσίευσης και στον χρήστη εμφανίζεται σχετικό μήνυμα. Ο έλεγχος περιορίζεται στον τίτλο της δημοσίευσης.

**public int** getTotalResults(String what, String value, String user)

η μέθοδος καλείται είτε για να βρεθεί ο συνολικός αριθμός δημοσιεύσεων στις οποίες έχει αποδοθεί μια συγκεκριμένη ετικέτα (η παραμέτρος what τίθεται σε tag, ο user σε null) είτε τον συνολικό αριθμό δημοσιεύσεων στις οποίες έχει αποδώσει μια συγκεκριμένη ετικέτα ένας συγκεκριμένος χρήστης (η παραμέτρος what τίθεται σε tag2) είτε για εύρεση του συνολικού αριθμού δημοσιεύσεων ενός συγκεκριμένου συγγραφέα (η παράμετρος what τίθεται σε name, ο user σε null).

**public** Vector getPublicationsOfAuthor(String name, **int** start, **int** end)

Επιστρέφει σε διάνυσμα από αντικείμενα publicationBean τις δημοσιεύσεις ενός συγγραφέα.

```
public Vector getPublicationsTaggedWith(String user, String tag, int start, int end)
```

Επιστρέφει σε διάνυσμα από αντικείμενα publicationBean τις δημοσιεύσεις στις οποίες έχει αποδοθεί μια συγκεκριμένη ετικέτα. Η ίδια μέθοδος μπορεί να υπερφορτωθεί (overloaded) για να επιστρέψει τις δημοσιεύσεις στις οποίες έχει αποδοθεί μια συγκεκριμένη ετικέτα από ένα συγκεκριμένο χρήστη.

```
public void removeUserTag(String user, String tag, int pubid)
```

Η μέθοδος καλείται ασύγχρονα από την publication.jsp στην περίπτωση που ένας χρήστης θέλει να αφαιρέσει μια δική του ετικέτα από την δημοσίευση. Συγκεκριμένα αφαιρεί τις κατάλληλες εγγραφές από τους πίνακες TAGS, PUBLICATIONTAGS και USERTAGS.

```
public void addUserTags(String user, Vector tags, int pubid)
```

Αντίστοιχα με την removeUserTag η μέθοδος καλείται επίσης ασύγχρονα από την publication.jsp στην περίπτωση που ένας χρήστης θέλει να προσθέσει μια δική του ετικέτα στην δημοσίευση και αναλαμβάνει να εγγράψει τις κατάλληλες εγγραφές στους πίνακες TAGS, PUBLICATIONTAGS και USERTAGS.

```
public Vector getUserTagsForPublication(String user, int pubid)
```

Η μέθοδος καλείται από την publication.jsp και επιστρέφει το σύνολο των ετικετών του χρήστη για την συγκεκριμένη δημοσίευση (το οποίο αποτελεί υποσύνολο του συνόλου των ετικετών από όλους τους χρήστες).

```
public UserBean getUserBean(String user)
```

Η μέθοδος καλείται από την profile.jsp και επιστρέφει σε μορφή αντικειμένου της κλάσης UserBean όλα τα στοιχεία ενός συγκεκριμένου χρήστη. Συγκεκριμένα πέραν του ονόματος, email κτλ το αντικείμενο αυτό συγκερατεί πληροφορία για τις ετικέτες που χρησιμοποιεί ο χρήστης, τις δημοσιεύσεις που έχει συνεισφέρει καθώς και τα σχόλια που έχει κάνει.

```
public int[] getSearchInsideDocumentsTotals()
```

Η μέθοδος καλείται από την searchDocs.jsp (η οποία εκτελεί αναζήτηση μέσα στο κείμενο των αρχείων βάση λέξης/ων κλειδί) και επιστρέφει σε πίνακα τον αριθμό των εγγράφων που υπάρχει η λέξη κλειδί και τις συνολικές εμφανίσεις της σε αυτά. Αυτό επιτυγχάνεται απλά με συναθροιστική συνάρτηση COUNT πάνω στους πίνακες MARKUP\_TABLE και HIGHTAB οι οποίοι αποθηκεύουν προσωρινά τα αποτελέσματα τέτοιας αναζήτησης.

```
public Vector searchInsideDocuments(String searchTerm)
```

η μέθοδος καλείται από την searchDocs.jsp και εμφανίζει τα αποτελέσματα αναζήτησης σε έγγραφα δημοσιεύσεων με βάση λέξη κλειδί. Συγκεκριμένα επιστρέφει το id της δημοσίευσης που περιέχει την λέξη κλειδί και τον αριθμό εμφανίσεων της λέξης κλειδί σε αυτή.



**public** Vector showResultsInsideDocument(String searchTerm, **int** pubid)  
η μέθοδος καλείται από την showOccurrences.jsp και αναλαμβάνει να εμφανίσει τα αποτελέσματα στο περιεχόμενο του εγγράφου της δημοσίευσης μετά από αναζήτηση με λέξη κλειδί. Πρώτα καλεί την διαδικασία της βάσης δεδομένων write\_to\_markup\_table() η οποία εγγράφει στους πίνακες MARKUP\_TABLE και HIGHTAB το κείμενο που εμφανίζεται η λέξη κλειδί και την θέση της λέξης κλειδί σε αυτό το κείμενο αντίστοιχα.

Ακολούθως εκτελεί σύνδεση των δυο πινάκων και ανακτά το κείμενο με την λέξη κλειδί, με αυτή πλέον να περικλείεται με την HTML ετικέτα <span>. Το κείμενο παρουσιάζεται στον χρήστη όπου με βοήθεια της γλώσσας CSS η λέξη κλειδί επισημαίνεται.

**public** CharArrayWriter showDocument(String searchTerm, **int** pubid)  
η μέθοδος καλείται από την showOccurrences.jsp στην περίπτωση που ο χρήστης θέλει να κοιτάξει το έγγραφο της δημοσίευσης χωρίς να το κατεβάσει μέσα από τον περιηγητή (browser). Συγκεκριμένα το έγγραφο λαμβάνεται από την βάση δεδομένων και μετατρέπεται σε HTML μορφή όπου και παρουσιάζεται στον χρήστη σε νέο παράθυρο.

**public void** truncateSearchTables()  
η μέθοδος αναλαμβάνει να αφαιρέσει όλα τα στοιχεία από τους πίνακες MARKUP\_TABLE και HIGHTAB οι οποίοι αποθήκευαν προσωρινά τα αποτελέσματα αναζήτησης με βάση λέξη κλειδί.

**public** Vector searchAuthors(String authorName, Vector resAreas, Vector resValues)

Η μέθοδος αναλαμβάνει να αναζητήσει συγγραφείς είτε με το όνομα τους είτε με τις ερευνητικές περιοχές που τον ενδιαφέρουν. Στην πρώτη περίπτωση εκτελείται ερώτηση στο σχεσιακό σχήμα της βάσης δεδομένων ενώ στην δεύτερη εκτελεί ερώτηση ασαφής συνάθροισης (fuzzy aggregation query) μέσω του reasoner της μορφής:

$$q(x) \leftarrow Author(x) \geq 1.0 \wedge isInterestedIn(x, researcharea) \geq k \wedge ResearchArea(researcharea) \geq 1.0$$

Όπου επιστρέφονται οι καλύτεροι κ (top-k) συγγραφείς σύμφωνα με τον βαθμό αληθείας τους.

**public** Vector searchPublications(String entryType, String title, String author, String publisher, String year)

Η μέθοδος επιτρέπει την αναζήτηση δημοσιεύσεων σύμφωνα με τα μεταδιδόμενα που την χαρακτηρίζουν (τύπος, τίτλος, συγγραφέας, εκδότης, έτος έκδοσης). Εκτελεί ερώτηση για όσα στοιχεία έχουν οριστεί και ακολούθως εκτελεί ένωση των αποτελεσμάτων, τα οποία και επιστρέφει ως διάνυσμα από αντικείμενα publicationBean. Το ερώτημα γίνεται στο πίνακα PUBINFO αν και θα μπορούσε να χρησιμοποιηθεί ο reasoner για πιο σημασιολογική αναζήτηση. Ο λόγος που επιλέγουμε να μην χρησιμοποιήσουμε τον reasoner έγκειται στην αδυναμία του να υποστηρίξει τον τελεστή LIKE ούτως ώστε να ανακτούνται παρόμοια αποτελέσματα με το κριτήριο αναζήτησης.

**public** Vector getAuthors(**int** howmany, **int** start, **int** end)

Καλείται από την exploreAuthors.jsp και επιστρέφει τα ονόματα των δημοφιλέστερων συγγραφέων (με βάση το πλήθος των δημοσιεύσεων τους που υπάρχουν στο σύστημα) καθώς και το πλήθος των δημοσιεύσεων αυτών.

**public void** removePublication(**int** pubid)

Η μέθοδος αναλαμβάνει να αφαιρέσει από την βάση γνώσης μια δημοσίευση μαζί με όλα τα μεταδιδόμενα που περιλαμβάνει αυτή. Συγκεκριμένα αφαιρεί τις κατάλληλες εγγραφές από τους πίνακες TAGS, USERTAGS, USERUPLOADS, PUBLICATIONTAGS, COMMENTS, USERCOMMENTS, RATING, PUBLICATIONRA, PUBINFO και PUBOBJECT ενώ αφαιρεί και από το σώμα ισχυρισμών όλους τους ισχυρισμούς που αφορούν την δημοσίευση με μόνη διαφοροποίηση την ενημέρωση (αντί διαγραφή) του ρόλου isInterestedIn για τους συγγραφείς σε περίπτωση που αυτοί έχουν κι άλλες δημοσιεύσεις.

**public** Vector searchUser(String userName, String tag, Vector resAreas, Vector resValues)

Η μέθοδος αυτή αναλαμβάνει να αναζητήσει χρήστες στο σύστημα σύμφωνα με το όνομα τους, ετικέτα που χρησιμοποιούν, ερευνητικές περιοχές που τους ενδιαφέρουν. Το αποτέλεσμα επιστρέφεται υπό τη μορφή διανύσματος από τα ονόματα των χρηστών που πληρούν τα κριτήρια.

**public** Vector getUserResearchAreas(**int** howmany)

Η μέθοδος καλείται από την exploreUsers.jsp και επιστρέφει τις πιο δημοφιλείς ερευνητικές περιοχές που ενδιαφέρουν τους χρήστες της κοινότητας σε μορφή διανύσματος από ζευγάρια: ερευνητική περιοχή και βαθμό ενασχόλησης σε αυτή.

**public** Vector getResearchAreas(**int** howmany)

Η μέθοδος καλείται από την explorePubs.jsp και επιστρέφει υπό την μορφή διανύσματος τις πιο δημοφιλείς ερευνητικές περιοχές που έχουν συσχετιστεί με τις υπάρχουσες δημοσιεύσεις στο σύστημα.

**public** Vector getAuthors(**int** howmany, **int** start, **int** end)

Η μέθοδος καλείται από την explorePubs.jsp και επιστρέφει υπό την μορφή διανύσματος τους πιο δημοφιλείς συγγραφείς δημοσιεύσεων (υπό την έννοια του πλήθους των δημοσιεύσεων του εκάστοτε συγγραφέα).

**public** Vector getUsers(**int** howmany, **int** start, **int** end)

Η μέθοδος καλείται από την exploreUsers.jsp και επιστρέφει τους πλέον ενεργούς χρήστες (υπό την έννοια του πλήθους των δημοσιεύσεων που έχουν συνεισφέρει στο σύστημα) υπό την μορφή διανύσματος από τα ονόματα των χρηστών.

**public** Vector getUsersNames(**int** howmany, **int** start, **int** end)

Η μέθοδος καλείται από την exploreUsers.jsp και επιστρέφει ένα υποσύνολο από τα ονομάτα των χρηστών υπό τη μορφή διανύσματος.

**public** Vector getYears(**int** howmany)

Η μέθοδος καλείται από την explorePubs.jsp και σκοπό έχει να επιστρέψει τα πιο δημοφιλή (υπό την έννοια ότι έχουν τις περισσότερες δημοσιεύσεις) έτη που υπάρχουν στο σύστημα.

**public** Vector getResearchAreasOfAuthors(**int** howmany)

Η μέθοδος καλείται από την exploreAuthors.jsp και επιστρέφει υπό την μορφή διανύσματος τις πιο δημοφιλείς ερευνητικές περιοχές που ενδιαφέρουν τους υπάρχοντες συγγραφείς στο σύστημα.

**public** Vector searchTag(String tag)

Η μέθοδος αναλαμβάνει να αναζητήσει στο σύνολο των υπαρχουσών ετικετών ετικέτες που μοιάζουν με τη δωθέν συμβολοσειρά. Το αποτέλεσμα επιστρέφεται υπό τη μορφή διανύσματος από ετικέτες που πληρούν τα κριτήρια.

```
public Vector getPublishers(int howmany, int start, int end)
```

Η μέθοδος καλείται από την explorePubs.jsp και σκοπό έχει να επιστρέψει τους πιο δημοφιλείς (υπό την έννοια ότι έχουν τις περισσότερες δημοσιεύσεις) εκδότες που υπάρχουν στο σύστημα.

## 5.2 Διαδικασία εγκατάστασης του συστήματος

Για την εγκατάσταση του συστήματος απαιτείται ένας Web container που θα αναλάβει να εξυπηρετεί τις HTTP αιτήσεις των χρηστών χρησιμοποιώντας τις Java κλάσεις και η εγκατάσταση σχεσιακής βάσης δεδομένων που θα υλοποιεί το στρώμα αποθήκευσης της εφαρμογής. Στην εφαρμογή που υλοποιήσαμε χρησιμοποιούμε τον Apache Tomcat ως εξυπηρετητή εφαρμογής (application server) και την Oracle Database 11g ως το σύστημα διαχείρισης βάσης δεδομένων (DBMS – DataBase Management System). Είναι σημαντικό να τονίσουμε ότι η επιλογή της Oracle ως ΣΔΒΔ μας περιορίζει στη συγγραφή κώδικα SQL καθώς αυτός γράφεται σε σύνταξη PL/SQL (η διάλεκτος SQL που χρησιμοποιεί η Oracle) και ο οποίος διαφέρει σε αρκετά σημεία από τη προτυποποιημένη σύνταξη SQL.

Εφόσον ολοκληρωθεί τη συγγραφή των .jsp αρχείων και .java αρχείων δημιουργούμε ένα .war αρχείο που περιέχει το σύνολο των αρχείων που απαιτούνται για τη λειτουργία του συστήματος. Αυτό μπορεί να γίνει με τρόπο αυτοματοποιημένο από το περιβάλλον του NetBeans (το IDE που χρησιμοποιήθηκε για την ανάπτυξη της εφαρμογής) εκτελώντας

Run > Clean and build main project. Στο υποφάκελο dist του φακέλου του project βρίσκεται το war αρχείο που μας ενδιαφέρει. Αφού το ανεβάσουμε στην ορισμένη θέση του εξυπηρετητή όπου θα στεγάζεται η εφαρμογή εκτελούμε την εντολή:

```
labcite@webapps:~/vhosts/labcite/webapps$ jar xf ~/labcite.war
```

η οποία αποσυμπιέζει το .war αρχείο και δημιουργεί την δεντρική μορφή που αναμένει ο Apache Tomcat

**labCiteDir**

```
<jsp files>
<other folders>
META-INF
    context.xml
WEB-INF
    web.xml
    classes
        <class files>
lib
```

<.jar files>

Όπως αναφέρθηκε στην ενότητα που αναφέρεται στην αρχιτεκτονική του συστήματος απαιτούνται δυο διακριτά σχήματα βάσης δεδομένων όπου θα αποτελούν το λογικό διαχωρισμό μεταξύ του σώματος ισχυρισμών της βάσης γνώσης και των δεδομένων που χρησιμοποιούνται στη διαχείριση των σελίδων του συστήματος.

Δημιουργία σχεσιακού σχήματος που αναπαριστά το την Οντολογία:

Για την δημιουργία του σχεσιακού σχήματος που αναπαριστά την Οντολογία και αποθηκεύει όλους τους ισχυρισμούς της βάσης γνώσης υπάρχουν δυο δυνατές επιλογές

Εφόσον εγκατασταθεί το .war αρχείο στον εξυπηρετητή και αποσυμπιεστεί κατευθυνόμενα στο φάκελο WEB-INF/classes/gr/ntua/ivml/site/Utilities και εκτελούμε την εντολή στο κέλυφος (shell):

```
java CreateFDLSchema <όρισμα1> <όρισμα2> <όρισμα3> <όρισμα4>
```

όπου όρισμα1 είναι η (απόλυτη) θέση του αρχείου .owl που περιγράφει την οντολογία στο σύστημα

όρισμα2 είναι το URL της σύνδεσης με την βάση δεδομένων

όρισμα3 είναι το όνομα (username) του χρήστη του σχήματος

όρισμα4 είναι ο κωδικός (password) του χρήστη του σχήματος

Π.χ.

```
labcite@webapps:~/WEB-INF/classes/gr/ntua/ivml/site/Utilities$ java CreateFDLSchema  
file:///home/labcite/vhosts/labcite/webapps/o.owl
```

```
jdbc:oracle:thin:@oracle.image.ece.ntua.gr:1521:oracle labCiteFDL qw easdzxc
```

Η εκτέλεση του προγράμματος CreateFDLSchema αναλαμβάνει να φορτώσει το αρχείο .owl, να εκτελέσει συντακτική ανάλυση (parsing) και να εξάγει τις έννοιες και ρόλους της οντολογίας και να δημιουργήσει τους αντίστοιχους πίνακες στο σχεσιακό σχήμα που δίνεται.

Δεύτερη επιλογή είναι να τρέξουμε μέσω ενός εργαλείου ανάπτυξης εφαρμογών SQL (π.χ. Oracle Sql Developer) το sql script FDLReasoner schema.sql το οποίο περιέχει τα DDL (Data Definition Language) που δημιουργούν το σχεσιακό σχήμα που εξυπηρετεί τον reasoner. Το συγκεκριμένο script δημιουργήθηκε βάση την τελευταία έκδοση της οντολογίας όπως αυτή περιγράφεται στην ενότητα 4.5. Σε περίπτωση ωστόσο που κάνουμε οποιαδήποτε αλλαγή στην Οντολογία (η οποία βρίσκεται στο .owl αρχείο) τότε πρέπει να ακολουθήσουμε την πρώτη μέθοδο καθώς οι αλλαγές δεν θα αποτυπωθούν στο σχήμα που θα δημιουργηθεί εφαρμόζοντας την δεύτερη μέθοδο.

Δημιουργία του δεύτερου σχήματος

Για να δημιουργήσουμε το δεύτερο σχήμα εκτελούμε μέσω Sql Developer το script create\_site\_schema.sql το οποίο περιέχει τα DDLs που δημιουργούν τους πίνακες και διαδικασίες του σχήματος όπως αυτοί έχουν οριστεί στην ενότητα 4.4. Εφόσον δημιουργηθεί το σχήμα μπορούμε να εκτελέσουμε και το script insert\_keywords.sql που αναλαμβάνει να συμπληρώσει τον σχεσιακό πίνακα KEYWORDS με τις αντιστοιχίσεις των συντομογραφήσεων των εννοιών της οντολογίας με τα κανονικά τους ονόματα.

### **5.3 Παρουσίαση χρήσης του συστήματος με τη βοήθεια σεναρίων**

Στην ενότητα αυτή παρουσιάζεται η χρήση του συστήματος με την βοήθεια τυπικών σεναρίων χρήσης.

#### **5.3.1 Εγγραφή, είσοδος χρήστη και εκκίνηση καινούργιας συνεδρίας**

Αν ο χρήστης προσπαθήσει να ανακτήσει οποιοδήποτε σελίδα του συστήματος χωρίς να έχει εκκινήσει μια συνεδρία (session) τότε θα ανακατευθυνθεί στην ακόλουθη σελίδα (welcome.jsp)

**Login page**

You've tried to access an unauthorized page  
Please login before trying to access unauthorized pages

userID What's your name?

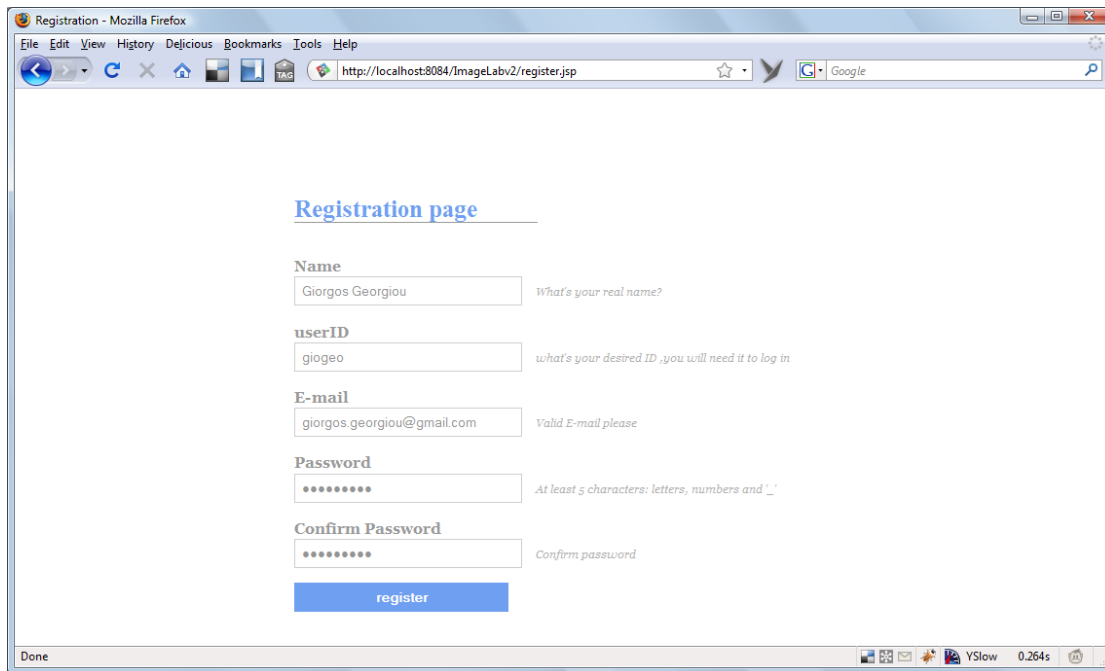
password Your password

login

Not a user yet? Go ahead and [register](#)

**Εικόνα 5.3.1 Η σελίδα welcome.jsp**

όπου και προτρέπεται να εισάγει το userID και κωδικό του για να εισέλθει στο σύστημα. Σε περίπτωση που ο χρήστης δεν διαθέτει λογαριασμό θα πρέπει να δημιουργήσει ένα κάνοντας κλικ στον σύνδεσμο register.



**Εικόνα 5.3.2 Η σελίδα registration.jsp**

Στην σελίδα register.jsp καλείται να συμπληρώσει μια φόρμα με το πραγματικό του όνομα, το επιθυμητό userID, την ηλεκτρονική του διεύθυνση και τον κωδικό (password) που επιθυμεί να έχει. Μετά την υποβολή της φόρμας (και εφόσον δεν υπάρχει κάποιο πρόβλημα π.χ. κενό πεδίο, λάθος επιβεβαίωση του κωδικού) θα ανακατευθυνθεί στην σελίδα καλωσορίσματος όπου καλείται να δώσει το userID και password για να συνδεθεί στο σύστημα.

## Login page

Your registration was succesful

Go ahead and log in.

userID

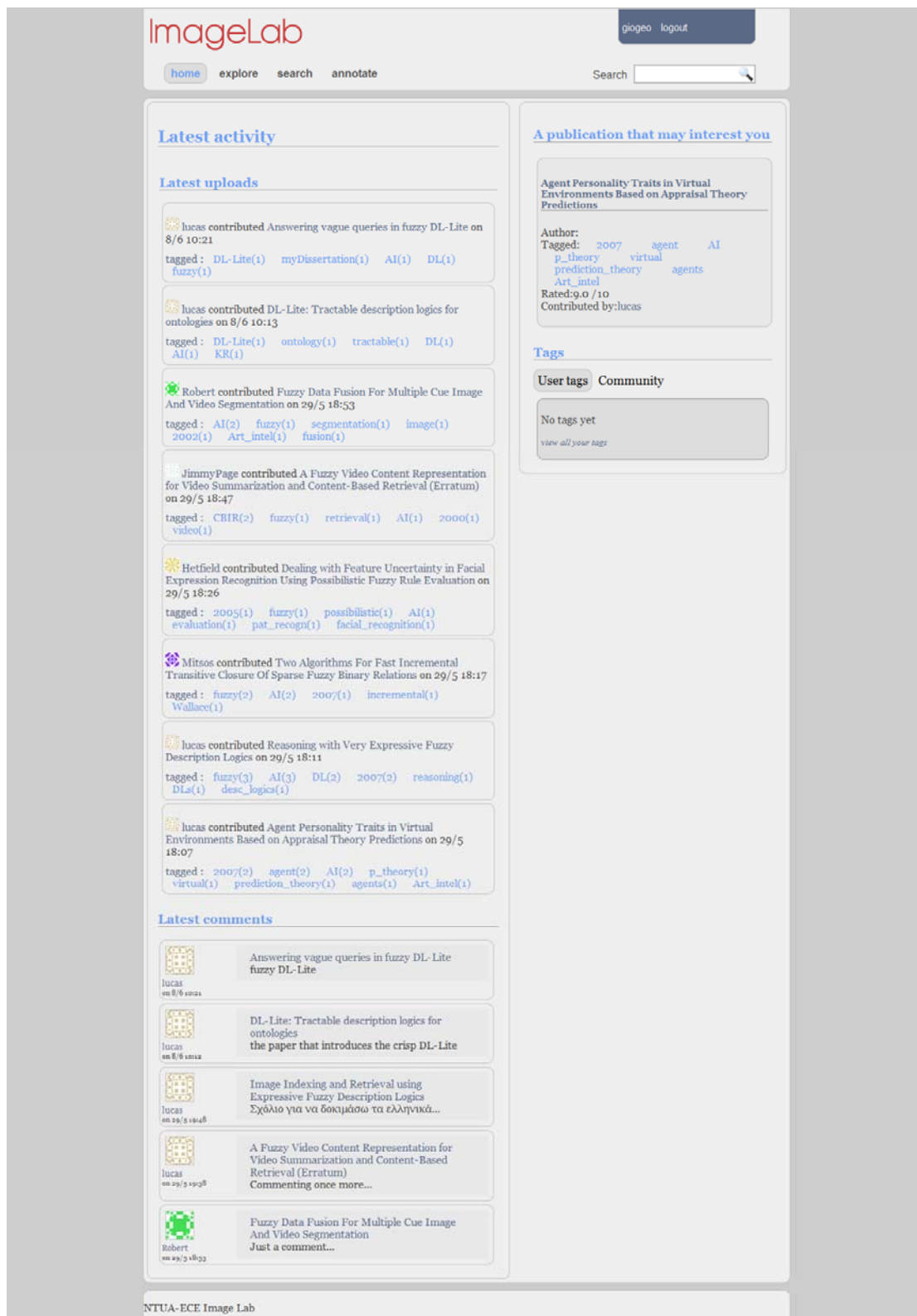
*What's your name?*

password

*Your password*

Not a user yet? Go ahead and [register](#)

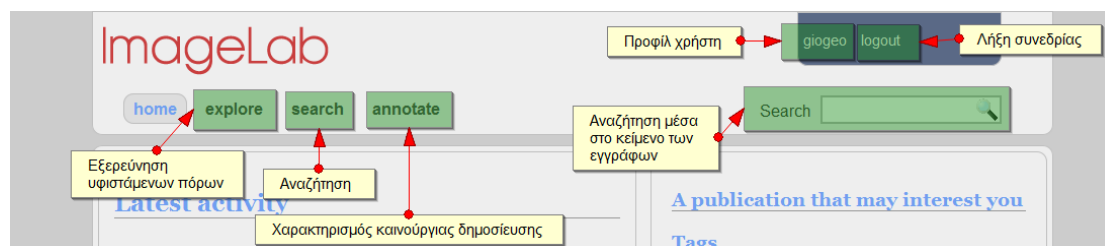
Εφόσον ο χρήστης δώσει σωστό userID και password (υπάρχει στην ΒΔ λογαριασμός με τα στοιχεία του) θα ανακατευθυνθεί στην κεντρική σελίδα του συστήματος.



Εικόνα 5.3.3 Η σελίδα home.jsp

Η κεντρική σελίδα του συστήματος αποτελείται από 3 κυρίες μονάδες: το μενού επιλογών, η αριστερή στήλη και η δεξιά στήλη.

Μενού επιλογών:



**Εικόνα 5.3.4 το μενού επιλογών**

Το μενού δίνει επιλογές στον χρήστη να περιηγηθεί στο σύστημα, συγκεκριμένα να εξερευνήσει τους υφιστάμενους πόρους (explore) δηλαδή τις υφιστάμενες δημοσιεύσεις, συγγραφείς, ετικέτες, χρήστες και ερευνητικές περιοχές. Επίσης μπορεί να αναζητήσει δημοσιεύσεις επιλέγοντας search είτε να χαρακτηρίσει μια καινούργια δημοσίευση επιλέγοντας annotate. Ακόμη μπορεί να δει το προφίλ του επιλέγοντας το όνομα του στην άνω δεξιά γωνία ή να τερματίσει την συνεδρία του επιλέγοντας logout. Επίσης δίνεται η δυνατότητα να εκτελέσει αναζήτηση μέσα στο κείμενο των εγγράφων των υφιστάμενων δημοσιεύσεων απ' ευθείας από το σχετικό πεδίο στα δεξιά του μενού.

Αριστερή στήλη:



**Latest activity**

**Latest uploads** ← Λίστα με τις τελευταίες συνεισφορές χρηστών

- lucas contributed Answering vague queries in fuzzy DL-Lite on 8/6 10:21  
tagged : DL-Lite(1) myDisse Τίτλος δημοσίευσης L(1) fuzzy(1)
- Χρήστης → lucas contributed DL-Lite: Tractable description logics for ontologies on 8/6 10:13  
tagged : DL-Lite(1) ontology(1) tractable(1) DL(1) AI(1) KR(1) ← Ετικέτες πρόσφατης δημοσίευσης
- Robert contributed Fuzzy Data Fusion For Multiple Cue Image And Video Segmentation on 29/5 18:53  
tagged : AI(2) fuzzy(1) segmentation(1) image(1) 2002(1) Art\_intel(1) fusion(1)
- lucas contributed Reasoning with Very Expressive Fuzzy Description Logics on 29/5 18:11  
tagged : fuzzy(3) AI(3) DL(2) 2007(2) reasoning(1) DLs(1) desc\_logics(1)
- lucas contributed Agent Personality Traits in Virtual Environments Based on Appraisal Theory Predictions on 29/5 18:07  
tagged : 2007(2) agent(2) AI(2) p\_theory(1) virtual(1) prediction\_theory(1) agents(1) Art\_intel(1)

**Latest comments** ← Λίστα με τον πιο πρόσφατο σχολιασμό

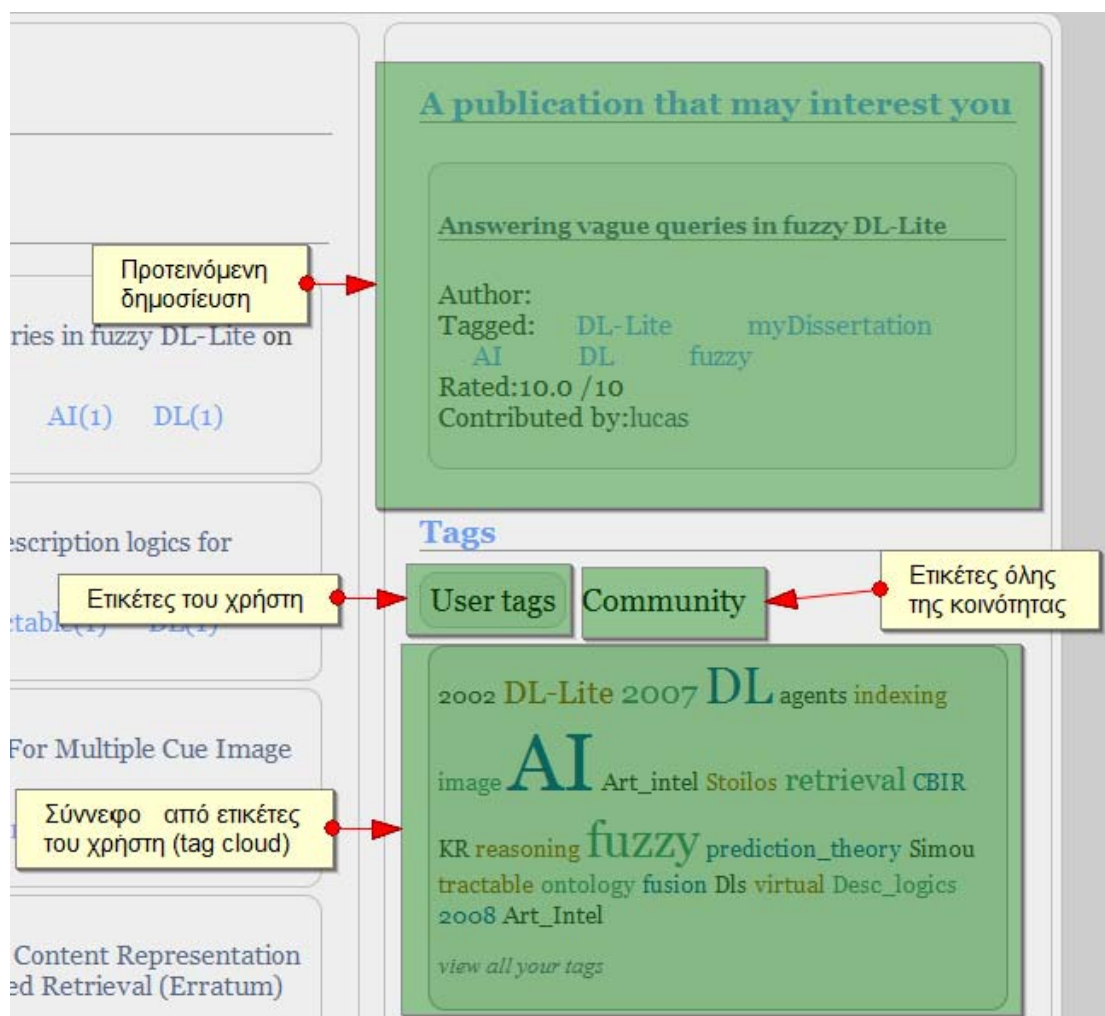
- lucas on 8/6 10:21 Answering vague queries in fuzzy DL-Lite fuzzy DL-Lite
- Χρήστης και ημερομηνία σχολιασμού → lucas on 8/6 10:12 DL-Lite: Tractable description logics for ontologies the paper that introduces the crisp DL-Lite → Τίτλος δημοσίευσης → Κείμενο σχολιασμού
- Robert on 29/5 18:53 Fuzzy Data Fusion For Multiple Cue Image And Video Segmentation Just a comment...

**Εικόνα 5.3.5 Η αριστερή στήλη της κεντρικής σελίδας**

Η αριστερή στήλη εμφανίζει την πιο πρόσφατη δραστηριότητα των χρηστών στο σύστημα. Συγκεκριμένα εμφανίζονται οι τελευταίοι χρονολογικά χαρακτηρισμοί δημοσιεύσεων, παρουσιάζοντας τον χρήστη που έκανε τον χαρακτηρισμό, τον τίτλο της δημοσίευσης και τις ετικέτες τις οποίες απέδωσε σε αυτή. Ακόμη εμφανίζονται τα τελευταία χρονολογικά σχόλια παρουσιάζοντας το χρήστη που έκανε το σχόλιο, το τίτλο της δημοσίευσης που έγινε το σχόλιο και το κείμενο του σχολίου. Ο χρήστης μπορεί να περιηγηθεί στις συγκεκριμένες δημοσιεύσεις, να δει το προφίλ των χρηστών και να εξερευνήσει τις ετικέτες που

παρουσιάζονται κατ' ευθείαν αφού όλοι οι πόροι εμφανίζονται με τον σχετικό υπερσύνδεσμο (link) που οδηγεί στη συγκριμένη σελίδα.

Δεξιά στήλη:



Εικόνα 5.3.6 Η δεξιά στήλη της κεντρικής σελίδας

Στην δεξιά στήλη παρουσιάζεται μια προτεινόμενη δημοσίευση για τον συγκεκριμένο χρήστη και το σύνολο των ετικετών του χρήστη αλλά και συνολικά όλης της κοινότητας. Η εισήγηση για τον χρήστη βγαίνει από ένα απλό αλγόριθμο: αν ο χρήστης έχει αναπτύξει δραστηριότητα (έχει χαρακτηρίσει έστω και μια δημοσίευση) τότε ανακτείται η ερευνητική περιοχή που τον ενδιαφέρει πιο πολύ και ανακτώνται ακολούθως δημοσιεύσεις που αφορούν την συγκεκριμένη ερευνητική περιοχή. Σε περίπτωση που δεν έχει αναπτύξει οιοδήποτε δραστηριότητα τότε ανακτείται μια δημοσίευση με εντελώς τυχαίο τρόπο. Το σύννεφο από ετικέτες δίνει την δυνατότητα στο χρήστη να ανακτήσει εύκολα όλες τις δημοσιεύσεις στις οποίες έχει αποδώσει μια συγκεκριμένη ετικέτα ο ίδιος ή αντίστοιχα η υπόλοιπη κοινότητα.

### 5.3.2 Εισαγωγή και χαρακτηρισμός νέας δημοσίευσης

Ο χρήστης μπορεί να χαρακτηρίσει μια καινούργια δημοσίευση επιλέγοντας από το μενού επιλογών `annotate` ή αν κατευθυνθεί απ' ευθείας στην σελίδα `annotation1.jsp` ( εφόσον φυσικά έχει εκκινήσει μια καινούργια συνεδρία). Ο χαρακτηρισμός γίνεται σε 3 βήματα.

### Βήμα 1:

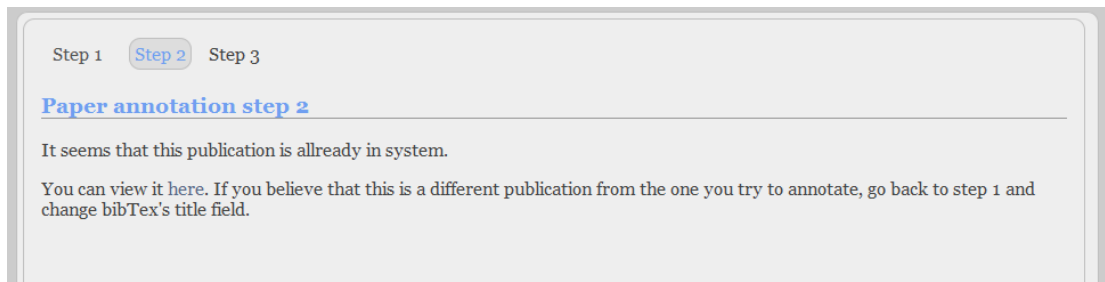
Εικόνα 5.3.7 Χαρακτηρισμός καινούργιας δημοσίευσης – Βήμα 1

Στο πρώτο βήμα ο χρήστης καλείται να εισάγει τα χαρακτηριστικά που συνθέτουν το `bibTex` της δημοσίευσης. Αυτό μπορεί να γίνει μηχανικά, εισάγοντας δηλαδή κάθε πεδίο ξεχωριστά, είτε να επικολλήσει απ' ευθείας το κείμενο του `bibTex`.

Εικόνα 5.3.8 Χαρακτηρισμός καινούργιας δημοσίευσης – Βήμα 1β

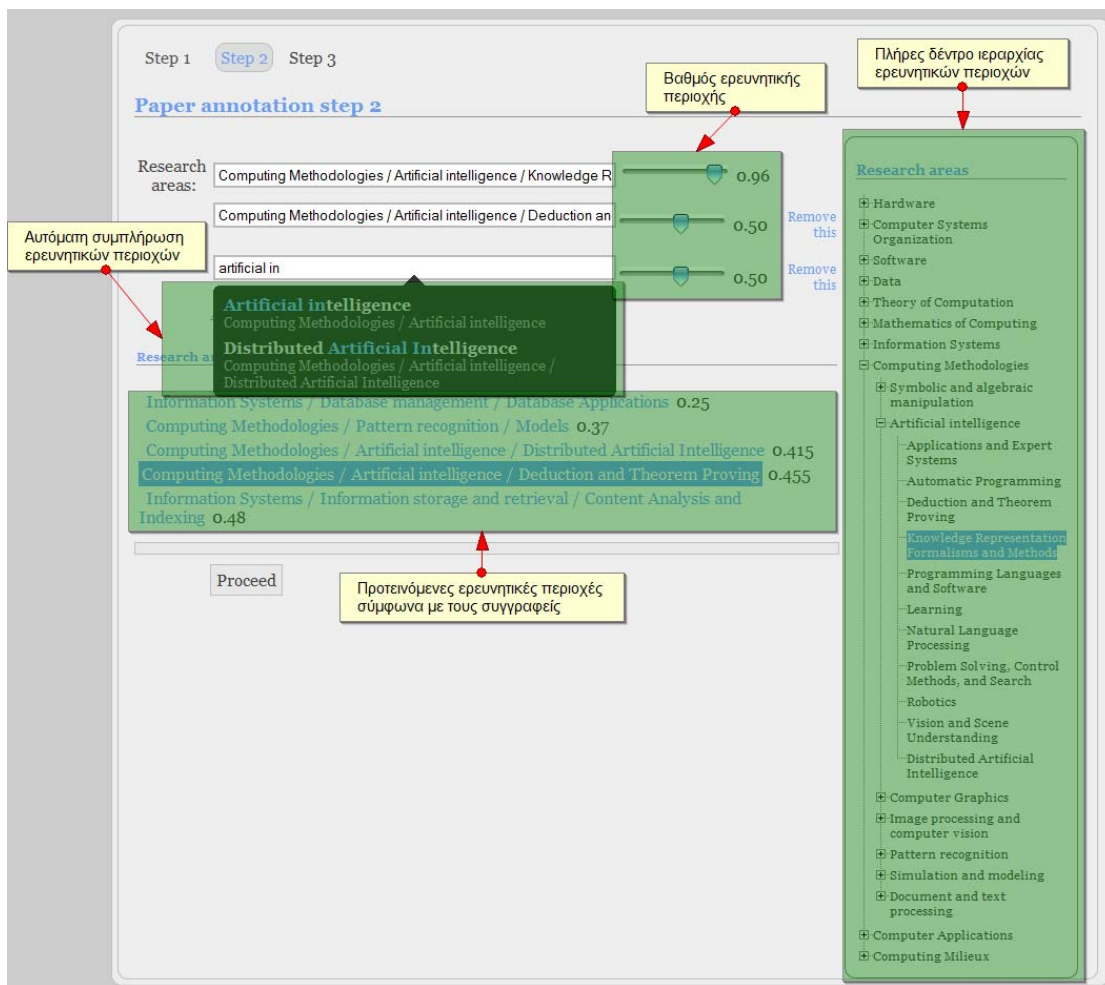
### Βήμα 2:

Στο δεύτερο βήμα εκτελείται έλεγχος κατά πόσο η δημοσίευση υπάρχει ήδη στο σύστημα. Σε τέτοια περίπτωση εμφανίζεται σχετικό μήνυμα:



**Εικόνα 5.3.9 Χαρακτηρισμός καινούργιας δημοσίευσης – Βήμα 2**

Σε αντίθετη περίπτωση ο χρήστης προτρέπεται να εισάγει τις ερευνητικές περιοχές που πιστεύει ότι η δημοσίευση ανήκει και το βαθμό που συμβαίνει αυτό.

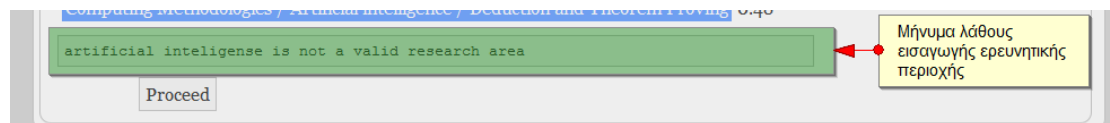


**Εικόνα 5.3.10 Χαρακτηρισμός καινούργιας δημοσίευσης – Βήμα 2β**

Για να διευκολυνθεί ο χρήστης στην εισαγωγή των ερευνητικών περιοχών υπάρχει μια σειρά από ευκολίες: Καθώς πληκτρολογεί την ερευνητική περιοχή στο αντίστοιχο πεδίο, εκτελείται ασύγχρονα (με χρήση τεχνολογίας AJAX) αναζήτηση στην βάση γνώσης και εμφανίζεται μια λίστα από ερευνητικές περιοχές που το όνομα τους μοιάζει με αυτό που εισήγαγε μέχρι στιγμής ο χρήστης. Ο χρήστης μπορεί τότε να επιλέξει την ερευνητική περιοχή που τον ενδιαφέρει από τη λίστα που του προσφέρεται. Ακόμη με το που φορτώνει η σελίδα εμφανίζεται μια λίστα από προτεινόμενες περιοχές σύμφωνα με τους συγγραφείς που ο

χρήστης έθεσε στο προηγούμενο βήμα (η λίστα διαμορφώνεται με συνδυαστικό ερώτημα στη βάση γνώσης της μορφής

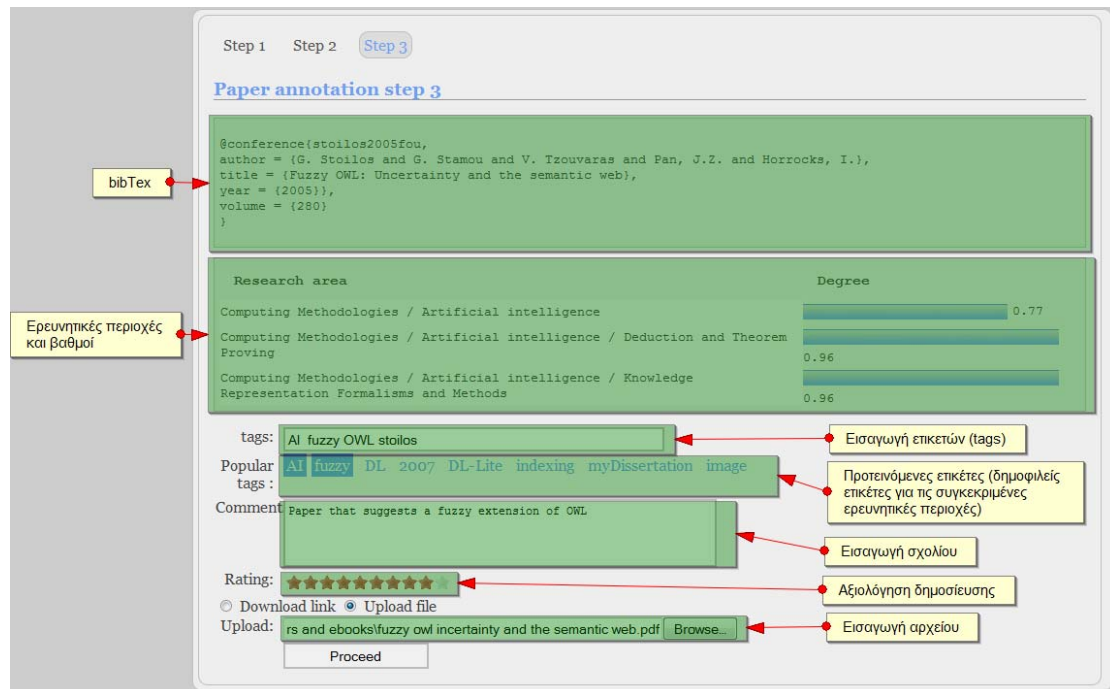
$q(x) \leftarrow ResearchAreas(x) \geq 1.0 \wedge isInterestedIn(author_1, x) \geq th \wedge \dots \wedge isInterestedIn(author_N, x) \geq th$ ) όπου ο χρήστης μπορεί με ένα απλό κλικ να εντάξει στο σύνολο των ερευνητικών περιοχών. Επίσης προσφέρεται μια οπτική απεικόνιση του δέντρου ιεραρχίας των ερευνητικών περιοχών που ο χρήστης μπορεί εύκολα να εξερευνήσει και να επιλέξει με απλό κλικ την ερευνητική περιοχή που τον ενδιαφέρει. Πατώντας το κουμπί proceed μπορεί να κατευθυνθεί στο τρίτο βήμα. Προτού όμως γίνει αυτό γίνεται έλεγχος κατά πόσο οι ερευνητικές περιοχές που έθεσε ανήκουν στο κλειστό λεξιλόγιο (closed vocabulary) του συστήματος. Σε περίπτωση που αυτό δεν συμβαίνει τότε εμφανίζεται σχετικό μήνυμα.



Εικόνα 5.3.11 Χαρακτηρισμός καινούργιας δημοσίευσης – Βήμα 2γ

### Βήμα 3:

Στο τρίτο και τελικό βήμα υπενθυμίζεται στο χρήστη τα μέχρι στιγμής δεδομένα που έχει εισάγει (τα οποία και μπορεί να επεξεργαστεί αν επιθυμεί επιλέγοντας τους συνδέσμους για τα προηγούμενα βήματα) και καλείται να συμπληρώσει μια σειρά από άλλα χαρακτηριστικά για να ολοκληρωθεί η διαδικασία χαρακτηρισμού



Εικόνα 5.3.12 Χαρακτηρισμός καινούργιας δημοσίευσης – Βήμα 3

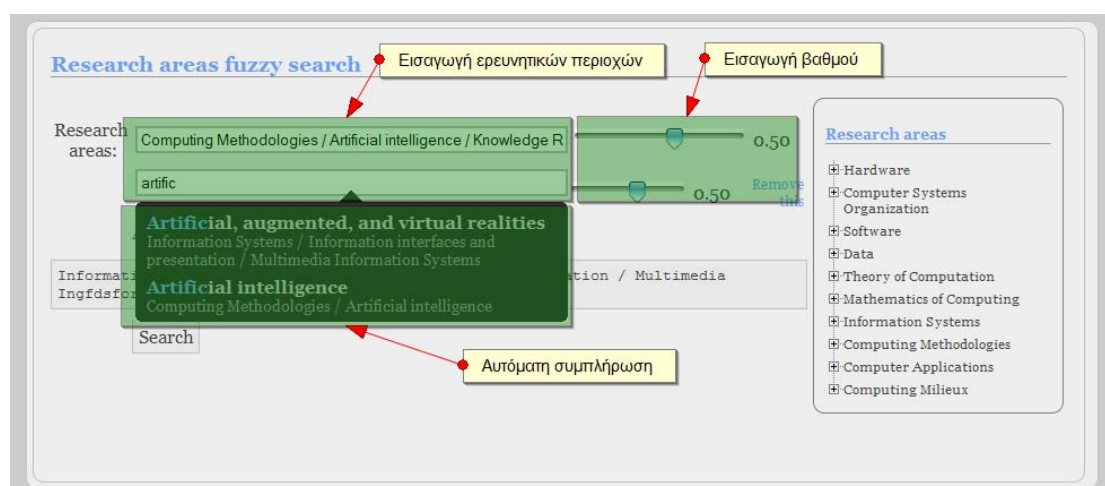
Συγκεκριμένα ο χρήστης κατ' αρχή προτρέπει να εισάγει ετικέτες (tags) για την δημοσίευση. Μπορεί να εισάγει απεριόριστο αριθμό ετικετών αρκεί να τις ξεχωρίσει με ένα



κενό διάστημα. Για διευκόλυνση του εμφανίζονται προτεινόμενες ετικέτες οι οποίες είναι οι πλέον δημοφιλείς ετικέτες για τις συγκεκριμένες ερευνητικές περιοχές που έχουν εισαχθεί στο σύστημα. Ακολούθως καλείται να προσθέσει ένα σύντομο σχόλιο καθώς και μια αξιολόγηση από το 1 ως το 10 για την δημοσίευση που εισάγει. Ως τελευταίο βήμα ο χρήστης καλείται να εισάγει το αρχείο του εγγράφου της δημοσίευσης ή τον σύνδεσμο (link) που μπορεί να ανακτηθεί η συγκεκριμένη δημοσίευση. Μετά την επιτυχή ολοκλήρωση της διαδικασίας ο χρήστης ανακατευθύνεται στην σελίδα publication.jsp όπου βλέπει συγκεντρωτικά τα στοιχεία της δημοσίευσης που μόλις έχει χαρακτηρίσει

### 5.3.3 Αναζήτηση με βάση την ερευνητική περιοχή

Στην περίπτωση που ο χρήστης θέλει να ανακτήσει μια δημοσίευση με βάση την ερευνητική περιοχή τότε πρέπει να κατευθυνθεί στην σελίδα search.jsp



Εικόνα 5.3.13 Αναζήτηση με βάση την ερευνητική περιοχή

Στη σελίδα αυτή καλείται να εισάγει την (ή τις) ερευνητική περιοχή που επιθυμεί να ανήκει η δημοσίευση καθώς και τον βαθμό. Όπως και στο βήμα 2 του χαρακτηρισμού μιας καινούργιας δημοσίευσης υπάρχει η ευκολία της αυτόματης συμπλήρωσης (με χρήση τεχνολογίας AJAX) καθώς και το πλήρες δέντρο της ιεραρχίας των ερευνητικών περιοχών. Ο χρήστης μπορεί να συμπληρώσει απεριόριστο αριθμό ερευνητικών περιοχών ούτως ώστε να συγκεκριμενοποιήσει όσο θέλει την αναζήτηση. Μετά την επιτυχή εισαγωγή των ερευνητικών περιοχών κατευθύνεται στην σελίδα των αποτελεσμάτων:

**Search results**

**Two Algorithms For Fast Incremental Transitive Closure Of Sparse Fuzzy Binary Relations**

Author: [M. Wallace](#) [S. Kollias](#)

Contributed by: [Mitsos](#)

Tags: [fuzzy\(2\)](#) [AI\(2\)](#) [2007\(1\)](#) [incremental\(1\)](#) [Wallace\(1\)](#)

**Matching degree 86.2 %** Βαθμός ταιριάσματος

Research area	Degree
Knowledge Representation Formalisms and Methods	0.5
Artificial intelligence	0.5

**DL-Lite: Tractable description logics for ontologies**

Author: [Calvanese, D.](#) [De Giacomo, G.](#) [Lembo, D.](#) [Lenzerini, M.](#) [Rosati, R.](#)

Contributed by: [lucas](#)

Tags: [DL-Lite\(1\)](#) [ontology\(1\)](#) [tractable\(1\)](#) [DL\(1\)](#) [AI\(1\)](#) [KR\(1\)](#)

**Matching degree 97.0 %** Βαθμός ταιριάσματος

Research area	Degree
Knowledge Representation Formalisms and Methods	0.98
Artificial intelligence	0.92
Deduction and Theorem Proving	0.37

**Answering vague queries in fuzzy DL-Lite**

Author: [Straccia, U.](#)

Contributed by: [lucas](#)

Tags: [DL-Lite\(1\)](#) [myDissertation\(1\)](#) [AI\(1\)](#) [DL\(1\)](#) [fuzzy\(1\)](#)

**Matching degree 97.0 %** Βαθμός ταιριάσματος

Research area	Degree
Knowledge Representation Formalisms and Methods	0.98
Deduction and Theorem Proving	0.82
Artificial intelligence	0.76

**Εικόνα 5.3.14 Αποτελέσματα αναζήτησης με βάση την ερευνητική περιοχή**

Στην σελίδα των αποτελεσμάτων παρουσιάζεται οι δημοσιεύσεις που πληρούν τα κριτήρια της αναζήτησης (στην εικόνα παρουσιάζεται το αποτέλεσμα για αναζήτηση Deduction and theorem proving κατά 0.8 και Knowledge representation κατά 0.75). Εκτός των ποιοτικών χαρακτηριστικών της εκάστοτε δημοσίευσης (τίτλος, συγγραφείς, χρήστης που την ανέβασε) παρουσιάζεται επίσης ο βαθμός ταιριάσματος (matching degree) και οι 3 ερευνητικές περιοχές με το μεγαλύτερο βαθμό. Ακόμη για κάθε αποτέλεσμα εμφανίζονται και οι ετικέτες που έχουν αποδώσει οι χρήστες στην συγκεκριμένη δημοσίευση.

### 5.3.4 Αναζήτηση με βάση τα περιγραφικά χαρακτηριστικά

Ο χρήστης δύναται να αναζητήσει δημοσιεύσεις σύμφωνα με τα περιγραφικά τους χαρακτηριστικά κατευθυνόμενος στην σελίδα explorePubs.jsp.

**Εικόνα 5.3.15 Εξερεύνηση δημοσιεύσεων**

Στη σελίδα αυτή ο χρήστης μπορεί να είτε να εξερευνήσει δημοσιεύσεις σύμφωνα με την δημοφιλία τους (αριστερή στήλη) είτε να αναζητήσει συγκεκριμένη δημοσίευση σύμφωνα με τα ποιοτικά χαρακτηριστικά της (δεξιά στήλη). Στην αριστερή στήλη η εξερεύνηση μπορεί να γίνει σύμφωνα με τον τύπο, τον συγγραφέα, το έτος έκδοσης, τον εκδότη/περιοδικό, την ερευνητική περιοχή και τις ετικέτες που έχουν αποδώσει οι χρήστες. Στη δεξιά στήλη ο χρήστης μπορεί να αναζητήσει μια συγκεκριμένη δημοσίευση συμπληρώνοντας τα κατάλληλα πεδία τύπου, συγγραφέα, περιοδικού ή έτους. Ακόμη δίνεται η δυνατότητα να αναζητήσει μέσα στο κείμενο των δημοσιεύσεων. Σε τέτοια περίπτωση θα κατευθυνθεί στην σελίδα των αποτελεσμάτων:



**Search results for text retrieval**

Total 35 occurrences in 4 documents

Συνολικές εμφανίσεις και αριθμός εγγράφων

**Image Indexing and Retrieval using Expressive Fuzzy Description Logics**

Research area: Artificial intelligence Degree: 0.5

Author: N. Simou Th. Athanasiadis G. Stoilos S. Kollias

Contributed by: lucas

Tags: indexing(3) Stoilos(2) Springer(2) Simou(2) image(2) 2008(2) video(1) retrieval(1) DL(1) Desc\_logics(1) Dls(1) fuzzy(1) Kollias(1) AI(1) Art\_Intel(1)

View 27 occurrences View as html

Τίτλος δημοσίευσης

Ερευνητικές περιοχές

**Two Algorithms For Fast Incremental Transitive Closure Of Sparse Fuzzy Binary Relations**

Research area: Knowledge Representation Formalisms and Methods Degree: 0.5  
Artificial intelligence Degree: 0.5

Author: M. Wallace S. Kollias

Contributed by: Mitsos

Tags: fuzzy(2) AI(2) 2007(1) incremental(1) Wallace(1)

Επικέτες

View 4 occurrences View as html

Δυνατότητα εμφάνισης του αρχείου σε μορφή Html

Συνολικές εμφανίσεις λέξης κλειδί

**Reasoning with Very Expressive Fuzzy Description Logics**

Research area: Knowledge Representation Formalisms and Methods Degree: 0.96  
Artificial intelligence Degree: 0.95  
Deduction and Theorem Proving Degree: 0.39

Author: G. Stoilos G. Stamou J.Z. Pan V. Tzouvaras I. Horrocks

Contributed by: lucas

Tags: fuzzy(3) AI(3) DL(2) 2007(2) reasoning(1) DLs(1) desc\_logics(1)

View 3 occurrences View as html

**A Fuzzy Video Content Representation for Video Summarization and Content-Based Retrieval (Erratum)**

Research area: Information Search and Retrieval Degree: 0.97  
Content Analysis and Indexing Degree: 0.96  
Artificial intelligence Degree: 0.8

Author: A. Doulamis N. Doulamis Y. Avrithis S. Kollias

Contributed by: JimmyPage

Tags: CBIR(2) fuzzy(1) retrieval(1) AI(1) 2000(1) video(1)

View 1 occurrences View as html

**Εικόνα 5.3.16 Αποτελέσματα αναζήτησης με λέξη κλειδί**

Στη σελίδα των αποτελεσμάτων παρουσιάζεται το σύνολο των δημοσιεύσεων που πληρούν τα κριτήρια της αναζήτησης (έχουν μέσα στο κείμενο τους την λέξη κλειδί). Η κάθε τέτοια δημοσίευση παρουσιάζεται με σύντομα περιγραφικά χαρακτηριστικά (τίτλος, συγγραφέας, χρήστης που την χαρακτήρισε και ερευνητικές περιοχές) και δίνεται η δυνατότητα στον χρήστη να δει τις εμφανίσεις της λέξης κλειδί μέσα στο κείμενο (view occurrences) ή να δει ολόκληρο το κείμενο σε μορφή html (χωρίς να χρειαστεί να το κατεβάσει). Το κάθε

περιγραφικό χαρακτηριστικό αποτελεί και σύνδεσμο για να εξερευνήσει ο χρήστης και άλλες δημοσιεύσεις με αυτά τα χαρακτηριστικά, π.χ. πατώντας πάνω σε μια ετικέτα θα περιηγηθεί σε όλες τις δημοσιεύσεις του συστήματος στις οποίες έχει αποδοθεί η συγκεκριμένη ετικέτα. Πατώντας πάνω στον σύνδεσμο για να δει τις εμφανίσεις της λέξης κλειδί θα κατευθυνθεί στην σελίδα showOccurrences.jsp:

**Showing occurrences of *retrieval* in document**

---

<p><b>Two Algorithms For Fast Incremental Transitive Closure Of Sparse Fuzzy Binary Relations</b></p> <p>Author: <a href="#">M. Wallace</a> <a href="#">S. Kollias</a></p> <p>Contributed by: <a href="#">Mitsos</a></p>	<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; border-bottom: 1px solid #ccc;">Research area</th> <th style="text-align: right; border-bottom: 1px solid #ccc;">Degree</th> </tr> </thead> <tbody> <tr> <td style="border-bottom: 1px solid #ccc;">Knowledge Representation Formalisms and Methods</td> <td style="text-align: right; border-bottom: 1px solid #ccc;">0.5</td> </tr> <tr> <td>Artificial intelligence</td> <td style="text-align: right;">0.5</td> </tr> </tbody> </table>	Research area	Degree	Knowledge Representation Formalisms and Methods	0.5	Artificial intelligence	0.5
Research area	Degree						
Knowledge Representation Formalisms and Methods	0.5						
Artificial intelligence	0.5						

Tags: [fuzzy\(2\)](#) [AI\(2\)](#) [2007\(1\)](#) [incremental\(1\)](#) [Wallace\(1\)](#)

cannot be calculated on-line; the relation needs to be available in an already transitive form if it is to be used by a knowledge system that aims to respond to user requests in a timely manner. Moreover, in parallel to the developments in the field of ontologies, the utilization of fuzzy binary relations for the representation of knowledge, towards more intelligent and semantics based information and multimedia **retrieval** [Avrithis et. al (2001)], has also started to be investigated. More information on the properties of these relations and the algorithms that utilize them can be acquired from [Akrivas et. al (2002)][Wallace et. al (2003a)][Wallace et. al (2003b)][Wallace et. al (2002)] [Avrithis et. al (2003)]. Similar arguments to those of the previous case lead to the conclusion the

close to  $n^2$ ; we may overcome this problem with the utilization of a more compact representation format, since utilized relations are typically very sparse; this has had a small expense of computational time required to access a specific element. Still, the two following issues, related to the theory and practice of transitive closure of fuzzy ordering binary relations will still have to be tackled: ♦ Most **retrieval** and document analysis algorithms that utilize relational knowledge assume that the fuzzy binary relation they accept as input is transitive in some form. Thus, the transitive closure of the  $n \times n$  relation has to be computed before the system is made operable for the first time. ♦ Relations that represent human knowledge are typically constructed

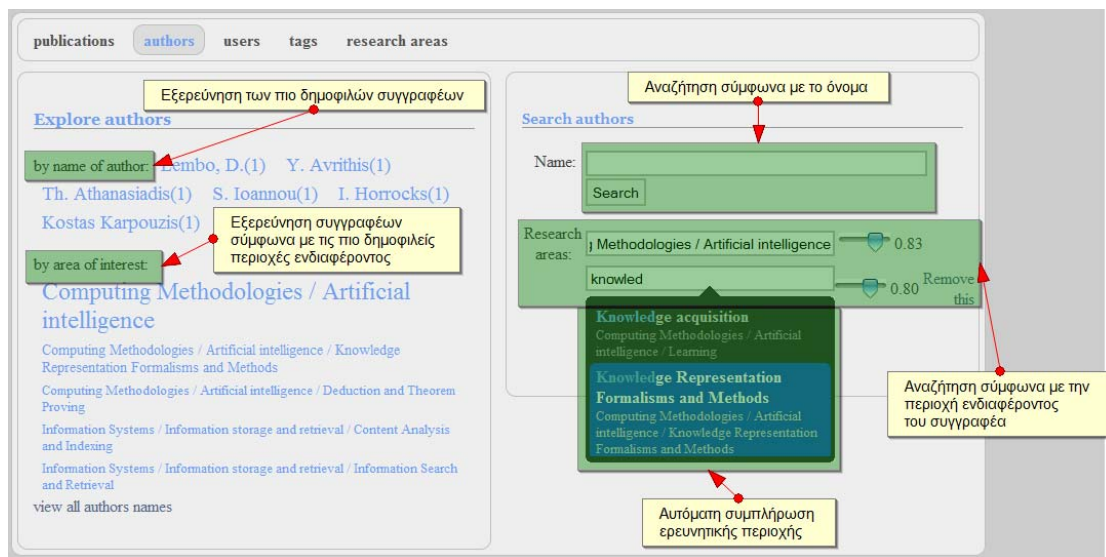
a computational complexity that is below  $O(n^2)$ . Finally, we have implemented the traditional and proposed methodologies and we have tested them on a real life fuzzy binary relation. The difference in execution time was immense, proving the efficiency of the proposed algorithm. The results of this work can be directly applied in the fields of knowledge based systems and intelligent information and multimedia **retrieval**, as they allow for the efficient handling of large sparse relations. Other fields and applications that require the transitive closure of binary relations, fuzzy or not, may benefit from this work if the relations they deal with conform to the assumption of sparseness. Acknowledgement The authors wish to acknowledge the contribution of Dr. Yannis Avrithis,

G. and Stamou, G., \Automatic Thematic Categorization of Docu- October 21, 2006 10:7 WSPC/INSTRUCTION FILE "wallace - transitive closure" 14 Manolis Wallace and Stefanos Kollias presents Using a Fuzzy Taxonomy and Fuzzy Hierarchical Clustering", FUZZ-IEEE, St. Louis, MO, USA, 2003. Wallace, M., Akrivas, G., Stamou, G. and Kollias, S., \Representation of user preferences and adaptation to context in multimedia content { based **retrieval** ", Workshop on Multimedia Semantics, SOFSEM, Milovy, Czech Republic, 2002. Wallace, M., Kollias, S., \Computationally efficient incremental transitive closure of sparse fuzzy binary relations", submitted to the IEEE International Conference of Fuzzy Systems (FUZZ-IEEE) in January 2004. Warren, H.S., \A modification of Warshall's algorithm for the

**Εικόνα 5.3.17 Εμφανίσεις λέξης κλειδί σε κείμενο δημοσίευσης**

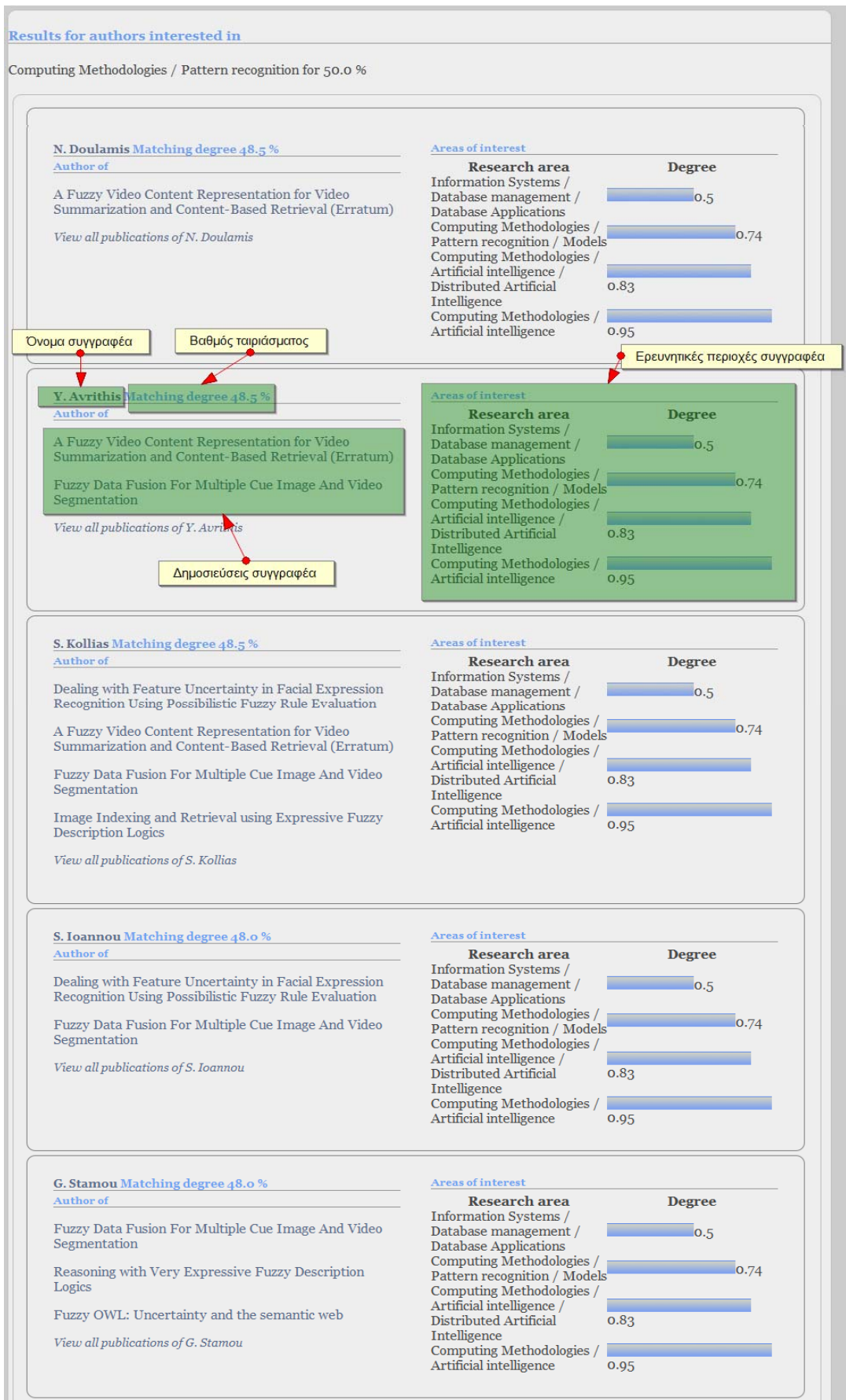
Στη σελίδα αυτή παρουσιάζεται επισημασμένη η λέξη κλειδί για κάθε εμφάνιση της στο κείμενο της δημοσίευσης.

Ο χρήστης μπορεί να αναζητήσει συγγραφείς κατευθυνόμενος στην σελίδα exploreAuthors.jsp.



**Εικόνα 5.3.18 Εξερεύνηση συγγραφέων**

Όπως και στην αντίστοιχη σελίδα για δημοσιεύσεις στην αριστερή στήλη μπορεί να εξερευνηθεί συγγραφείς σύμφωνα με την δημοφιλία τους ενώ στην δεξιά να αναζητήσει ένα συγκεκριμένο συγγραφέα σύμφωνα με το όνομα ή με την ερευνητική περιοχή που ασχολείται. Συγκεκριμένα στην αριστερή στήλη παρουσιάζονται τα ονόματα των πιο δημοφιλών συγγραφέων (σύμφωνα με το πλήθος των δημοσιεύσεων τους που υπάρχουν στο σύστημα) και οι πλέον δημοφιλείς ερευνητικές περιοχές των συγγραφέων. Πατώντας πάνω σε μια τέτοια ερευνητική περιοχή θα κατευθυνθεί σε σελίδα που θα παρουσιάζει όλους τους συγγραφείς που ασχολούνται με την συγκεκριμένη ερευνητική περιοχή. Στην δεξιά στήλη μπορεί να αναζητήσει συγκεκριμένο συγγραφέα σύμφωνα με το όνομα ή να περιορίσει την έρευνα του σύμφωνα με τις ερευνητικές περιοχές εισάγοντας όσες ερευνητικές περιοχές επιθυμεί και τον βαθμό. Σε τέτοια περίπτωση θα κατευθυνθεί στη σελίδα:

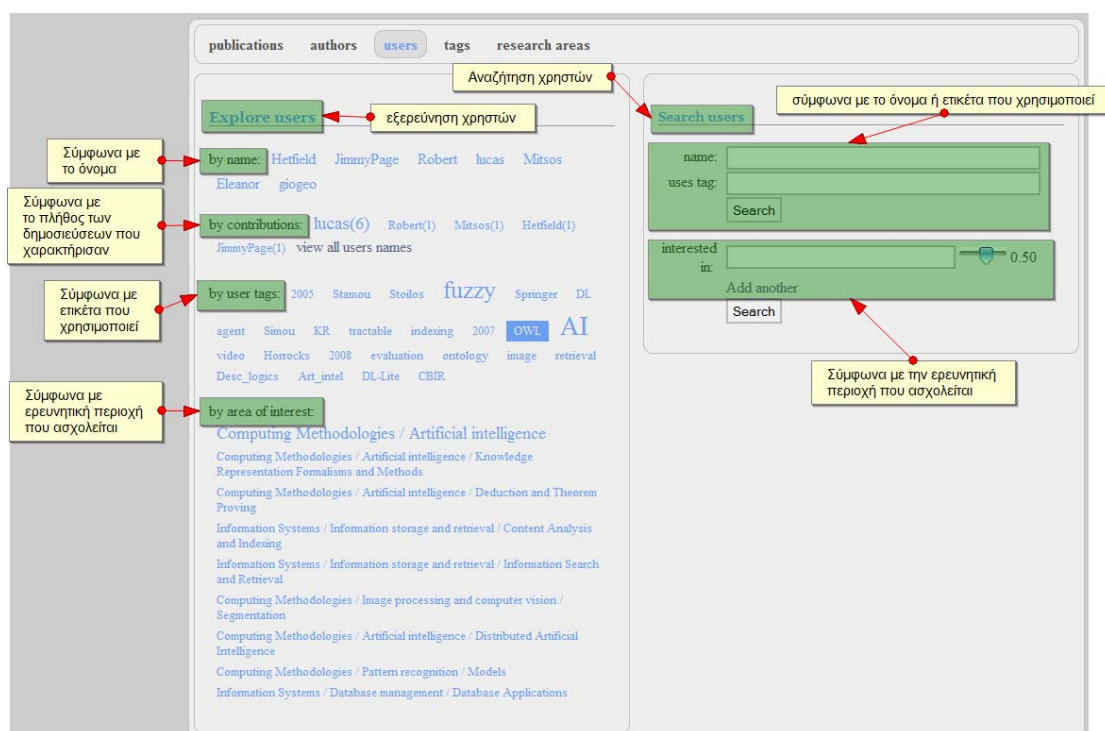


Εικόνα 5.3.19 Αποτελέσματα αναζήτησης συγγραφέων



Όπου εμφανίζεται λίστα με το σύνολο των συγγραφέων που πληρούν τα κριτήρια αναζήτησης. Κάθε τέτοιο αποτέλεσμα παρουσιάζει το όνομα του συγγραφέα, το βαθμό ταιριάσματος (matching degree) του συγκεκριμένου συγγραφέα σύμφωνα με τα κριτήρια της αναζήτησης, μερικές (ή όλες ανάλογα με το πλήθος) δημοσιεύσεις του συγκεκριμένου συγγραφέα και οι ερευνητικές περιοχές που τον ενδιαφέρουν περισσότερο.

Ο χρήστης μπορεί να αναζητήσει άλλους χρήστες του συστήματος κατευθυνόμενος στην σελίδα exploreUsers.jsp.




**Εικόνα 5.3.20 Εξερεύνηση χρηστών συστήματος**

Όπως και στις άλλες αντίστοιχες σελίδες στην αριστερή στήλη δίνεται η δυνατότητα να εξερευνήσει τους χρήστες σύμφωνα με την δημοφιλία τους και στην δεξιά να αναζητήσει ένα συγκεκριμένο χρήστη. Συγκεκριμένα στην αριστερά στήλη ο χρήστης μπορεί να ανακτήσει άλλους χρήστες σύμφωνα με το όνομα, σύμφωνα με το πλήθος των δημοσιεύσεων που χαρακτήρισαν (οι πιο ενεργοί χρήστες), σύμφωνα με ετικέτα που χρησιμοποιεί ή σύμφωνα με την ερευνητική περιοχή που τον ενδιαφέρει. Στην δεξιά στήλη μπορεί να αναζητήσει ένα χρήστη σύμφωνα με το όνομα, μια ετικέτα που χρησιμοποιεί ή ερευνητικές περιοχές που τον ενδιαφέρουν. Σε κάθε περίπτωση ο χρήστης όταν επιλέξει ένα τέτοιο σύνδεσμο (ή εκτελέσει αναζήτηση) θα κατευθυνθεί στη σελίδα:

Results for users interested in  
Computing Methodologies / Artificial intelligence / Deduction and Theorem Proving

**Hetfield** ← Όνομα χρήστη



Real name: James Hetfield  
email: James.Hetfield@metallica.com


see complete profile → Σύνδεσμος προβολής πλήρους προφίλ

Σύννεφο ετικετών που χρησιμοποιεί

Tag cloud

pat\_recogn video image **AI**  
Springer evaluation DL Art\_intel  
2005 **fuzzy** 2007 indexing  
agent

**lucas**



Real name: Lucas Anastasiou  
email: anastluc@gmail.com  
see complete profile

Tag cloud

KR **fuzzy** CBIR image reasoning ontology virtual  
indexing OWL DL-Lite Dls agents Desc\_logics retrieval  
Art\_intel DL 2002 fusion Simou Stoilos **AI** 2007  
stoilos prediction\_theory 2008

**Εικόνα 5.3.21 Αποτελέσματα αναζήτησης χρηστών**

Όπου εμφανίζεται λίστα με όλους τους χρήστες που πληρούν τα κριτήρια της αναζήτησης. Κάθε τέτοιο αποτέλεσμα παρουσιάζει το όνομα του χρήστη, το πραγματικό του όνομα, την ηλεκτρονική του διεύθυνση (email), σύνδεσμο για προβολή του πλήρους προφίλ του χρήστη και ένα σύννεφο από ετικέτες που χρησιμοποιεί προσωπικά ο συγκεκριμένος χρήστης για να οργανώσει τις ετικέτες που τον ενδιαφέρουν. Αν ο χρήστης επιλέξει μια ετικέτα ενός χρήστη θα κατευθυνθεί σε σελίδα που παρουσιάζονται όλες οι δημοσιεύσεις που ο συγκεκριμένος χρήστης έχει αποδώσει την συγκεκριμένη ετικέτα.

Publications tagged with fuzzy

Χρήστης: lucas  
Ετικέτα: fuzzy  
Αριθμός αποτελεσμάτων: Results 1-4 of 4 total results

**Image Indexing and Retrieval using Expressive Fuzzy Description Logics**  
 Author: N. Simou Th. Athanasiadis G. Stoilos S. Kollias  
 Tagged: indexing Stoilos Springer Simou image 2008 video retrieval DL Desc\_logics Dls  
 fuzzy Kollias AI Art\_Intel  
 Rated:9.0 /10

**Answering vague queries in fuzzy DL-Lite**  
 Author: Straccia, U.  
 Tagged: DL-Lite myDissertation AI DL fuzzy  
 Rated:10.0 /10

**Reasoning with Very Expressive Fuzzy Description Logics**  
 Author: G. Stoilos G. Stamou J.Z. Pan V. Tzouvaras I. Horrocks  
 Tagged: fuzzy AI DL 2007 reasoning DLs desc\_logics  
 Rated:9.0 /10

**Fuzzy OWL: Uncertainty and the semantic web**  
 Author: G. Stoilos G. Stamou V. Tzouvaras Pan, J.Z. Horrocks, I.  
 Tagged: fuzzy AI DL OWL KR representation Stamou art\_intel 2005 Horrocks stoilos  
 Rated:9.0 /10

Εικόνα 5.3.2 Αποτελέσματα αναζήτησης με βάση την ετικέτα

Αντίστοιχα μπορεί να κατευθυνθεί στη σελίδα που παρουσιάζεται το πλήρες προφίλ του χρήστη:

lucas profile

Real name: Lucas Anastasiou

Σύννεφο από ετικέτες του χρήστη

Λίστα δημοσιεύσεων που συνείσφερε ο χρήστης

**User contributions**

- contributed Fuzzy OWL: Uncertainty and the semantic web on 9/6 19:25  
tagged: Horrocks 2005 fuzzy art\_intel Stamou stoilos KR OWL DL representation AI
- contributed Answering vague queries in fuzzy DL-Lite on 8/6 10:21  
tagged: DL-Lite myDissertation fuzzy DL AI
- contributed DL-Lite: Tractable description logics for ontologies on 8/6 10:13  
tagged: DL-Lite ontology KR DL AI tractable
- contributed Reasoning with Very Expressive Fuzzy Description Logics on 29/5 18:11  
tagged: fuzzy DLs desc\_logics DL reasoning 2007 AI

**User comments**

- Fuzzy OWL: Uncertainty and the semantic web  
Paper that suggests a fuzzy extension of OWL  
on 9/6 19:25
- Answering vague queries in fuzzy DL-Lite  
fuzzy DL-Lite  
on 8/6 10:21
- DL-Lite: Tractable description logics for ontologies  
the paper that introduces the crisp DL-Lite  
on 8/6 10:12
- Image Indexing and Retrieval using Expressive Fuzzy Description Logics  
Σχόλιο για να δοκιμάσω τα ελληνικά...  
on 29/5 19:48

Λίστα με σχόλια του χρήστη

Εικόνα 5.3.23 Προφίλ χρήστη

Στη σελίδα αυτή πέραν του σύννεφου από ετικέτες που παρουσιάζεται μπορεί να δει τη πλήρη λίστα από δημοσιεύσεις που συνείσφερε (χαρακτήρισε) στο σύστημα ο συγκεκριμένος χρήστης καθώς και τη πλήρη λίστα από τα σχόλια που έκανε ο χρήστης σε δημοσιεύσεις.



# 6

## *Επίλογος*

Ακολουθεί μια σύνοψη των αποτελεσμάτων της διπλωματικής καθώς και κατευθύνσεις για μελλοντικές επεκτάσεις του συστήματος.

### **6.1 Σύνοψη και συμπεράσματα**

Η παρούσα διπλωματική ασχολήθηκε με την υλοποίηση συστήματος οργάνωσης βιβλιογραφικών δημοσιεύσεων (reference management software) όπου ενθυλακώνονται κοινωνικά χαρακτηριστικά στα πλαίσια της φιλοσοφίας του Web 2.0 ενώ ταυτόχρονα γίνεται χρήση τεχνολογιών σημασιολογικού ιστού. Δίνεται η δυνατότητα για ασαφή χαρακτηρισμό των βιβλιογραφικών αναφορών με τη βοήθεια σχετικής ασαφούς Οντολογίας ενώ ταυτόχρονα επιτρέπεται η χρήση αυθαίρετου λεξιλογίου μέσω ετικετών. Επιτεύχθηκε η διασύνδεση του συστήματος με μηχανή συλλογιστικής που διαθέτει δυνατότητα αναζήτησης με χρήση συνδετικών ερωτημάτων ασαφούς συνάθροισης. Ως περιγραφική λογική της Οντολογίας που χρησιμοποιήθηκε επιλέχθηκε η ασαφής έκδοση της γλώσσας DL-Lite εκμεταλλευόμενοι έτσι τα μοναδικά χαρακτηριστικά που παρουσιάζει η γλώσσα αυτή: πλούσια εκφραστικότητα και βατή πολυπλοκότητα, κάτι που επιτρέπει την κλιμάκωση (scaling) της εφαρμογής χωρίς ιδιαίτερες απαιτήσεις σε λειτουργικούς πόρους.

Το σύστημα που υλοποιήθηκε αποτελεί μια υβριδική μέθοδο για τη λύση του προβλήματος της ανάκτησης πληροφορίας (information retrieval) καθώς καταφέρνει να συνδυάσει δυο δημοφιλείς μεθόδους χαρακτηρισμού και αναζήτησης πληροφορίας: τη μέθοδο της ετικετοποίησης και ανάκτηση με χρήση ενδιάμεσου στρώματος Οντολογίας. Η χρήση ετικετών επιτρέπει την από κοινού διαμόρφωση των μεταδεδομένων του κάθε πόρου ενώ η χρήση οντολογίας επιτρέπει τη χρήση μεθόδων συλλογιστικής στην αναζήτηση και ανάκτηση πληροφορίας.

## 6.2 Μελλοντικές επεκτάσεις

Πιθανές μελλοντικές επεκτάσεις του συστήματος μπορεί να είναι:

Υλοποίηση υπηρεσίας ιστού (web service) για εκμετάλλευση των δυνατοτήτων του συστήματος από τρίτες εφαρμογές και ανάπτυξης εργαλείων που προσφέρουν στην ευχρηστία του συστήματος. Πιθανά προγράμματα που μπορούν να γραφτούν ώστε να κάνουν χρήση της υπηρεσίας είναι πρόγραμμα μαζικού χαρακτηρισμού ενός συνόλου από δημοσιεύσεις που διαθέτουν τα ίδια χαρακτηριστικά

Ενσωμάτωση ετικέτας (tag) στο εννοιολογικό στρώμα της Οντολογίας. Αυτό θα επιτρέψει τη δημιουργία εξαγωγής συσχετίσεων της σημασιολογίας κάθε ετικέτας με συγκεκριμένους πόρους και με ανάπτυξη αλγορίθμων να χρησιμοποιηθεί ως μηχανή εισηγήσεων (recommendation engine) ή για άλλες εφαρμογές.

Ανάπτυξη εργαλείων που επιτρέπουν την απευθείας εισαγωγή αναφορών στο σύστημα από δημοφιλείς πύλες αναζήτησης βιβλιογραφικών αναφορών όπως π.χ. το Google Scholar.

Επεξεργασία φυσικής γλώσσας των κειμένων των δημοσιεύσεων που εισάγονται στο σύστημα. Αυτό θα επιτρέψει μια σειρά από λειτουργίες όπως:

Η πιο απλή εφαρμογή θα είναι η εξαγωγή των κύριων λέξεων κλειδιών (keywords) που περιέχονται στα έγγραφα όπου θα μπορούν να χρησιμοποιηθούν ως προτεινόμενες ετικέτες

Την εξαγωγή των αναφορών που περιέχει η εκάστοτε δημοσίευση κάτι που θα οδηγήσει στο χτίσιμο μιας βιβλιογραφικής βάσης όπου θα μπορούν να εξαχθούν χρήσιμοι βιβλιομετρικοί δείκτες όπως π.χ. citation-index, h-index, impact factor κτλ.

Μια πιο εξειδικευμένη εφαρμογή είναι η απευθείας συσχέτιση του κειμένου με έννοιες μιας καλά ορισμένης οντολογίας. Προς αυτή την κατεύθυνση μπορεί να γίνει χρήση υφιστάμενων υπηρεσιών όπως είναι π.χ. η υπηρεσία openCalais<sup>1</sup>, απαιτείται όμως η διασύνδεση - ευθυγράμμιση της Οντολογίας αυτής με την Οντολογία που χρησιμοποιεί το σύστημα

---

<sup>1</sup> <http://www.opencalais.com/>

# 7

## Βιβλιογραφία

- [1] A. Hotho, R. Jaschke, C. Schmitz, and G. Stumme, "Folkrank: A ranking algorithm for folksonomies," *Proc. FGIR*, vol. 2006, 2006.
- [2] G. Vossen and S. Hagemann, "Unleashing web 2.0: from concepts to creativity," *Ubiquity*, vol. 8, 2007.
- [3] T. O'Reilly, "What is Web 2.0: Design patterns and business models for the next generation of software."
- [4] N. Zakas, J. McPeak, and J. Fawcett, *Professional Ajax*: Wrox, 2007.
- [5] E. Quintarelli, "Folksonomies: power to the people," 2005.
- [6] S. Hayman, "Folksonomies and Tagging: new developments in social bookmarking," vol. 27, p. 2007, 2007.
- [7] A. Mathes, "Folksonomies-cooperative classification and communication through shared metadata," 2004.
- [8] A. Acquisti and R. Gross, "Imagined communities: Awareness, information sharing, and privacy on the facebook," *Lecture notes in computer science*, vol. 4258, pp. 36-58, 2006.
- [9] G. Stoilos, "Εισαγωγή στις περιγραφικές λογικές," 2007.
- [10] F. Baader, D. Calvanese, D. L. McGuinness, D. Nardi, and P. F. Patel-Schneider, *The Description Logic Handbook: Theory, Implementation, and Applications*: Cambridge University Press, 2003.
- [11] M. Schmidt-Schaub and G. Smolka, "Attributive concept descriptions with complements," *Artificial intelligence*, vol. 48, pp. 1-26, 1991.
- [12] D. Calvanese, G. De Giacomo, M. Lenzerini, R. Rosati, and G. Vetere, "DL-Lite: Practical reasoning for rich DLs," 2004.
- [13] G. Stoilos, "Αβεβαιότητα και σημασιολογικός ιστός: Εκφραστικές Ασαφείς Περιγραφικές λογικές," vol. PhD Athens: National Technical University of Athens, 2008, p. 185.
- [14] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati, "DL-Lite: Tractable description logics for ontologies," 2005, p. 602.
- [15] D. Calvanese, G. De Giacomo, D. Lembo, M. Lenzerini, and R. Rosati, "Data complexity of query answering in description logics," *Proc. of KR*, vol. 2006, pp. 260-270, 2006.
- [16] E. Sanchez, *Fuzzy logic and the semantic web*: Elsevier, 2006.
- [17] U. Straccia, "Answering vague queries in fuzzy DL-Lite," 2006, p. 2238?2245.
- [18] A. Silberschatz, H. Korth, and S. Sudershan, *Database system concepts*: McGraw-Hill, Inc. New York, NY, USA, 1998.
- [19] <http://www.ibm.com/developerworks/podcast/dwi/cm-int082206txt.html>