



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

**Ανάπτυξη Διαδικτυακής Εφαρμογής Σημασιολογικής  
Πλοήγησης σε Σχεσιακές Βάσεις Δεδομένων**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**ΑΠΟΣΤΟΛΟΥ Α. ΔΟΥΚΑ**

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

Αθήνα, Οκτώβριος 2014





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

## Ανάπτυξη Διαδικτυακής Εφαρμογής Σημασιολογικής Πλοήγησης σε Σχεσιακές Βάσεις Δεδομένων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

**ΑΠΟΣΤΟΛΟΥ Α. ΔΟΥΚΑ**

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

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 30 Οκτωβρίου 2014.

.....  
Γιώργος Στάμου  
Επίκουρος Καθηγητής Ε.Μ.Π.

.....  
Στέφανος Κόλλιας  
Καθηγητής Ε.Μ.Π.

.....  
Ανδρέας – Γεώργιος  
Σταφυλοπάτης  
Καθηγητής Ε.Μ.Π.

Αθήνα, Οκτώβριος 2014

.....  
Απόστολος Λ. Δούκας

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

Copyright © Απόστολος Λ. Δούκας, 2014

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

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

## **Ευχαριστίες**

Θα ήθελα να εκφράσω τις θερμότερες ευχαριστίες μου στον Επίκουρο Καθηγητή κ. Γιώργο Στάμου που μου έδωσε την ευκαιρία να πραγματοποιήσω αυτή τη διπλωματική εργασία και μου συμπαραστάθηκε σε κάθε βήμα, από την έναρξη μέχρι και την ολοκλήρωσή της. Επιπλέον, ευχαριστώ τον ερευνητή Δρ. Αλέξανδρο Χορταρά για τον κώδικα του συστήματος επανεγγραφής που μου παρείχε, καθώς και για τις πολύτιμες ιδέες και συμβουλές του.

Τέλος, ευχαριστώ τη μητέρα μου, η οποία στάθηκε δίπλα μου χωρίς δισταγμό σε κάθε δύσκολη στιγμή.



## Περίληψη

Ένα βασικό πεδίο έρευνας της επιστήμης των υπολογιστών είναι η αναπαράσταση γνώσης, καθώς και η γρήγορη και αποδοτική συλλογιστική σε αυτήν. Η οικογένεια των γλωσσών Περιγραφικής Λογικής παρέχει αυτές τις δυνατότητες, ενώ συγκεκριμένες υλοποιήσεις της, όπως η DL-Lite, επιτρέπουν την ενοποίηση δεδομένων με υφιστάμενα συστήματα σχεσιακών βάσεων δεδομένων, όπου το μεγαλύτερο μέρος της πληροφορίας βρίσκεται ήδη αποθηκευμένο.

Αντικείμενο της παρούσας διπλωματικής εργασίας είναι η ανάπτυξη μίας διαδικτυακής εφαρμογής πλοήγησης σε πραγματικά στοιχεία δανειακών λογαριασμών. Στα πλαίσια της υλοποίησης δημιουργήθηκε μία οντολογία, η οποία περιγράφει τις ιεραρχίες των εννοιών που χαρακτηρίζουν τα εξεταζόμενα στοιχεία καθώς και τις σχέσεις που τις διέπουν, βασισμένη στην Περιγραφική Λογική DL-Lite. Τα αποτελέσματα των αναζητήσεων που πραγματοποιούνται από την εφαρμογή περιορίζονται μέσω επιλογής φίλτρων από το χρήστη, που εμφανίζονται με τη μορφή facets και αντιστοιχούν σε στοιχεία της οντολογίας. Οι επιλογές των φίλτρων μετατρέπονται βάσει της οντολογίας σε ένα συζευκτικό ερώτημα που εισάγεται σε ένα διαθέσιμο σύστημα επανεγγραφής και μετασχηματίζεται σε μία ένωση SQL ερωτημάτων. Τα τελικά ερωτήματα εφαρμόζονται σε μία κλασική σχεσιακή βάση δεδομένων και τα συνοπτικά αποτελέσματα προβάλλονται στις οθόνες της εφαρμογής. Ο χρήστης έχει τη δυνατότητα να πλοηγηθεί στις σελίδες των συνοπτικών αποτελεσμάτων, να επιλέξει νέα ή διαφορετικά φίλτρα και να προβάλει τα αναλυτικά στοιχεία συγκεκριμένων λογαριασμών σε νέο παράθυρο.

**Λέξεις Κλειδιά:** Περιγραφικές Λογικές, DL-Lite, Οντολογία, Faceted Αναζήτηση, Συζευκτικά Ερωτήματα, Επανεγγραφή Ερωτημάτων, Ενοποίηση Δεδομένων





## **Abstract**

A key research area in computer science is knowledge representation, as well as the efficient reasoning over this knowledge. Description Logic languages provide these capabilities, while specific implementations, such as DL-Lite, allow data integration with preexisting database management systems, where the main volume of the information is already stored.

The objective of this diploma thesis is the development of a web application that allows the navigation to information related to bank loan accounts. For this purpose, we created an ontology that describes the class hierarchies considering the available data, and the relationships between them, based on the Description Logic DL-Lite. The returned search results are limited through the selection of faceted filters, which correspond to elements of the ontology. The filter selections are translated to a conjunctive query. This query is fed into an available query rewriting system and is transformed into a union of SQL queries. The final queries are applied to a relational database, and the summary results are displayed as snippets to the main window of the application. The user of the application is able to navigate to the various pages that contain the summary results, select new or different filters from the faceted menu and view the details of a specific account to a new window.

**Keywords:** Description Logics, DL-Lite, Ontology, Faceted Search, Conjunctive Queries, Query Rewriting, Data Integration



## Πίνακας Περιεχομένων

### Περιεχόμενα

1	Εισαγωγή .....	13
1.1	Γενικά .....	13
1.2	Συνοπτική Περιγραφή Υλοποίησης .....	14
1.3	Περιγραφή Περιεχομένων .....	15
2	Θεωρητικό Πλαίσιο .....	17
2.1	Αναπαράσταση Γνώσης .....	17
2.2	Οντολογίες.....	18
2.3	Περιγραφικές Λογικές.....	19
2.3.1	Βασικά Συστατικά Οντολογιών των Περιγραφικών Λογικών .....	19
2.3.2	Η Οικογένεια Περιγραφικών Λογικών DL – Lite.....	23
2.4	Η Γλώσσα OWL.....	27
3	Πολύπλευρη Αναζήτηση (Faceted Search) .....	29
3.1	Γενικά για τα Facets και το Διαδίκτυο .....	29
3.2	Ταξονομίες, Faceted Ταξινόμηση και Οντολογίες.....	31
3.2.1	Ταξονομίες.....	31
3.2.2	Faceted Ταξινόμηση και Οντολογίες .....	33
3.3	Faceted Αναζήτηση σε Βάσεις Γνώσης – Φορμαλιστική Περιγραφή.....	35
4	Περιγραφή Μεθοδολογίας και Οντολογιών .....	39
4.1	Κίνητρο Ανάπτυξης.....	39
4.2	Περιγραφή Χαρακτηριστικών – Έννοιες και Σχέσεις.....	40
4.2.1	Η Έννοια του Δανείου.....	41
4.2.2	Η Έννοια του Δανειολήπτη .....	42
4.2.3	Η έννοια του Ακινήτου.....	43
4.2.4	Η έννοια της Γεωγραφικής Περιοχής.....	43
4.3	Ανάπτυξη Οντολογίας .....	44
4.4	Συζευκτικά Ερωτήματα .....	48
4.5	Εξαγωγή Συμπερασμάτων.....	50
5	Ανάπτυξη και Υλοποίηση .....	53
5.1	Γενική Περιγραφή της Εφαρμογής.....	53
5.2	Χρησιμοποιούμενες Τεχνολογίες .....	55
5.3	Αρχιτεκτονική της Εφαρμογής.....	57
5.3.1	Το Πρότυπο MVC .....	57
5.3.2	Οθόνες .....	59

5.3.3 Έλεγχος Ροής των Εργασιών.....	60
5.3.4 Διαχείριση Δεδομένων και Ερωτημάτων .....	62
5.4 Κατασκευή και Βελτιστοποίηση Ερωτημάτων .....	65
5.4.1 Το Συζευκτικό Ερώτημα .....	65
5.4.2 Επανεγγραφή του Ερωτήματος .....	66
5.4.3 Βελτιστοποίηση του Ερωτήματος .....	67
6 Οθόνες Εφαρμογής – Αποτελεσμάτων.....	69
6.1 Κεντρική Οθόνη .....	69
6.2 Φίλτρα και Αποτελέσματα .....	70
6.3 Προβολή Αναλυτικών Στοιχείων .....	72
7 Επίλογος .....	75
7.1 Σύνοψη .....	75
7.2 Μελλοντικές Επεκτάσεις.....	76
8 Βιβλιογραφία .....	77

# 1

## *Εισαγωγή*

### *1.1 Γενικά*

Στο επιστημονικό πεδίο της τεχνητής νοημοσύνης, η αναπαράσταση και η διασύνδεση της γνώσης, καθώς και οι μηχανισμοί συλλογιστικής πάνω σε αυτήν, αποτελούν σημαντικά αντικείμενα έρευνας τα τελευταία χρόνια. Η ραγδαία εξάπλωση του παγκόσμιου ιστού (world wide web – internet), καθώς και η αφθονία της πληροφορίας που κατακλύζει τους διάφορους ιδιωτικούς και δημόσιους οργανισμούς, έχει φέρει στο προσκήνιο την ανάγκη για νέες τεχνικές αναπαράστασης και ταξινόμησής, με σκοπό να καταστήσει ευκολότερη την πλοήγηση και την αναζήτηση πάνω στην πληροφορία αυτή.

Οι σχεσιακές βάσεις δεδομένων αποτελούν το πιο διαδεδομένο μέσο αποθήκευσης πληροφοριών για κάθε είδους εταιρία και οργανισμό παγκοσμίως. Τα διάφορα εμπορικά συστήματα διαχείρισης βάσεων δεδομένων (Relational Database Management Systems – RDBMS) που έχουν αναπτυχθεί και εξελίσσονται διαρκώς εδώ και αρκετά χρόνια, προσφέρουν σημαντικές δυνατότητες αποθήκευσης και άντλησης δεδομένων, καθώς παρέχουν εξελιγμένες και γρήγορες μεθόδους υλοποίησης ερωτημάτων πάνω σε πολύ μεγάλους όγκους πληροφορίας.

Οι τεχνολογίες γνώσεις μπορούν να συμπληρώσουν τα παραπάνω, προς όφελος των τελικών χρηστών στους οποίους παρουσιάζεται η ανάγκη να πλοηγηθούν σε περίπλοκα

και εκτεταμένα στοιχεία. Η χρήση οντολογιών, μέσω των οποίων αναπαρίστανται σημαντικές έννοιες και σχέσεις μεταξύ των εννοιών, καθώς και οι μηχανισμοί συλλογιστικής που παρέχονται μέσω διαφόρων εργαλείων όπως οι περιγραφικές λογικές, είναι σε θέση να βελτιώσουν και να απλοποιήσουν την αναζήτηση, καθώς και να βοηθήσουν στην οργάνωση και την αναπαράσταση της πληροφορίας, μέσω της δημιουργίας μίας ευέλικτης και λειτουργικής διεπαφής (interface)

## **1.2 Συνοπτική Περιγραφή Υλοποίησης**

Στην εργασία αυτή παρουσιάζεται η ανάπτυξη μίας εφαρμογής πλοήγησης σε δανειακούς λογαριασμούς. Για τους σκοπούς της υλοποίησης χρησιμοποιήθηκαν πραγματικά αποπροσωποποιημένα στοιχεία λογαριασμών και δανειοληπτών, τα οποία παρασχέθηκαν από μία ελληνική εμπορική τράπεζα.

Στην πρώτη φάση της υλοποίησης δημιουργήθηκε μία οντολογία στην οποία περιγράφονται βασικές έννοιες, ιεραρχίες εννοιών και ρόλοι. Οι κύριες έννοιες της οντολογίας βασίζονται σε λογικές έννοιες και κατηγοριοποιήσεις που αναπαρίστανται στα αντίστοιχα διαθέσιμα στοιχεία της τράπεζας, όπως “*Δάνειο*” και “*Δανειολήπτης*”, οι οποίες αναλύονται σε μία πληθώρα υποεννοιών, όπως για παράδειγμα η υποέννοια “*Στεγαστικό Δάνειο*”. Οι ρόλοι ορίζουν σχέσεις μεταξύ των εννοιών, όπως για παράδειγμα ο ατομικός ρόλος “*ανήκειΣε*”, ο οποίος συνδέει αντικείμενα από την έννοια “*Δάνεια*” με αντίστοιχα στιγμιότυπα της έννοιας “*Δανειολήπτης*”. Τέλος, στην οντολογία ορίζονται αξιώματα τα οποία συσχετίζουν τις διάφορες έννοιες και ορίζουν νέες.

Με βάση την παραπάνω οντολογία αναπτύχθηκε μία δικτυακή (web) εφαρμογή η οποία παρέχει στο χρήστη τη δυνατότητα πολύπλευρης (*faceted*) αναζήτησης και πλοήγησης στα δανειακά στοιχεία. Το μενού της εφαρμογής χτίζεται με τη μορφή δέντρου, βασισμένο στις ιεραρχίες της οντολογίας, ενώ οι σχέσεις μεταξύ των βασικών εννοιών χρησιμοποιούνται για την προβολή των διαφορετικών facets. Σύμφωνα με τις επιλογές του χρήστη δημιουργείται ένα συζευκτικό ερώτημα βάσει των εννοιών και των σχέσεων που επελέγησαν. Το συζευκτικό ερώτημα και η οντολογία τροφοδοτούνται σε ένα σύστημα επανεγγραφής ερωτημάτων (query rewriter), το οποίο έχει αναπτυχθεί από την ομάδα του εργαστηρίου Επεξεργασίας Εικόνας, Βίντεο και Πολυμέσων του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Ε.Μ.Π. Μέσα από αυτή τη διαδικασία προκύπτει τελικά ένα SQL ερώτημα που εφαρμόζεται στη βάση δεδομένων με τα τραπεζικά στοιχεία. Τα τελικά αποτελέσματα σε επίπεδο δανειακού λογαριασμού, προβάλλονται στην οθόνη της εφαρμογής, όπου είναι διαθέσιμη και η δυνατότητα προβολής αναλυτικότερων πληροφοριών σε ξεχωριστό παράθυρο.

### 1.3 Περιγραφή Περιεχομένων

Ακολουθεί μία σύντομη περιγραφή των κεφαλαίων από τα οποία συντίθεται η παρούσα διπλωματική εργασία, πέρα από το τρέχον κεφάλαιο που αποτελεί την εισαγωγή:

- **Θεωρητικό Πλαίσιο:** Σε αυτό το κεφάλαιο παρουσιάζονται και αναλύονται τα θεωρητικές έννοιες και εργαλεία που χρησιμοποιήθηκαν στα πλαίσια της ανάπτυξης της εφαρμογής πλοήγησης, όπως οι γλώσσες αναπαράστασης γνώσης DL-Lite, που αποτελούν υποσύνολο της οικογένειας των γλωσσών Περιγραφικής Λογικής, και οι μηχανισμοί συλλογιστικής και εξαγωγής συμπερασμάτων. Επιπλέον γίνεται συνοπτική αναφορά στην έννοια της οντολογίας και στη γλώσσα αναπαράστασης οντολογιών OWL.
- **Faceted Αναζήτηση και Σχετικές Υλοποιήσεις:** Παρουσίαση της έννοιας του faceted search και περιγραφή αντίστοιχων συστημάτων, στα οποία υλοποιήθηκε ενοποίηση δεδομένων (data integration) με ταυτόχρονη εφαρμογή της.
- **Περιγραφή Μεθοδολογίας, Δεδομένων, Οντολογίας και Ερωτημάτων:** Ανάλυση της μεθοδολογίας που ακολουθήθηκε για την ανάπτυξη της εφαρμογής, των διαθέσιμων δεδομένων και των σχέσεων που τα διέπουν και παρουσίαση της οντολογίας που υλοποιήθηκε.
- **Ανάπτυξη και Υλοποίηση:** Αναλυτική περιγραφή των βημάτων που ακολουθήθηκαν, καθώς και των διάφορων εργαλείων και τεχνολογιών που χρησιμοποιήθηκαν, στα πλαίσια της ανάπτυξης της εφαρμογής
- **Οθόνες Εφαρμογής και Αποτελεσμάτων:** Παρουσίαση χρήσης της εφαρμογής, μέσω συγκεκριμένων παραδειγμάτων αναζήτησης και προβολής των σχετικών οθονών με τα αποτελέσματα





# 2

## *Θεωρητικό Πλαίσιο*

### *2.1 Αναπαράσταση Γνώσης*

Ένας ορισμός της τεχνητής νοημοσύνης είναι η εμφάνιση ευφυούς συμπεριφοράς από μία μηχανή, η οποία επιτυγχάνεται με υπολογιστικά μέσα. Η αναπαράσταση γνώσης και η συλλογιστική αποτελούν έναν κλάδο της τεχνητής νοημοσύνης που μελετά τον τρόπο με τον οποίο ένα πρόγραμμα που ονομάζεται «πράκτορας» χρησιμοποιεί αυτά που γνωρίζει έτσι ώστε να αποφασίσει τι να πράξει. Με άλλα λόγια, πρόκειται για την αναγωγή της σκέψης – συλλογιστικής σε υπολογιστική διαδικασία.

Το πρόβλημα της καταγραφής και της τυποποίησης της ανθρώπινης γνώσης έχει απασχολήσει του επιστήμονες και τους φιλόσοφους αιώνες πριν την εποχή των ηλεκτρονικών υπολογιστών. Από την κλασσική αρχαιότητα εντοπίζονται οι πρώτες προσπάθειες εύρεσης μίας τυπικής (formal) μεθόδου καταγραφής της γνώσης. Οι προσπάθειες αυτές εστιάζονταν κυρίως στην τυποποίηση μαθηματικών προβλημάτων, με αποτέλεσμα οι διάφορες μέθοδοι που αναπτύχθηκαν να μην αντιστοιχούν απόλυτα στις ανάγκες αναπαράστασης μη μαθηματικής γνώσης, κάτι που είναι η κύρια περίπτωση στις εφαρμογές της τεχνητής νοημοσύνης στην πληροφορική. Παρ' όλ' αυτά, τα διάφορα συστήματα που αναπτύχθηκαν και χρησιμοποιήθηκαν κατά το παρελθόν από φιλόσοφους, μαθηματικούς και επιστήμονες της λογικής αποτέλεσαν τη βάση πάνω στην οποία στηρίχθηκαν οι σύγχρονες σχετικές μέθοδοι. Τα συστήματα αναπαράστασης γνώσης προσομοιάζουν στις φυσικές γλώσσες που μιλούν οι άνθρωποι, γι αυτό και

αποκαλούνται και γλώσσες. Έχουν συντακτικό και σημασιολογία, όπως και οι φυσικές γλώσσες, διαθέτουν όμως επιπλέον το χαρακτηριστικό του μηχανισμού εξαγωγής συμπερασμάτων ή συλλογιστικής (reasoning), μέσω του οποίου δίνεται η δυνατότητα εξαγωγής νέας γνώσης και συμπερασμάτων, με σημείο αναφοράς την αρχική γνώση.

Χαρακτηριστικά παραδείγματα γλωσσών αναπαράστασης γνώσης είναι η προτασιακή λογική και η κατηγορηματική λογική. Η δεύτερη αποτελεί ένα γενικό όρο που χαρακτηρίζει διάφορα συμβολικά τυπικά συστήματα (formal systems) όπως η κατηγορηματική λογική πρώτης τάξης, η λογική δεύτερης τάξης, η λογική πολλών ειδών (many – sorted logic) και άλλες. Η πιο σημαντική από αυτές είναι η λογική πρώτης τάξης (first – order logic), πάνω στην οποία βασίστηκαν νεότερες γλώσσες αναπαράστασης γνώσης που σχεδιάστηκαν από επιστήμονες πληροφορικούς για να καλύψουν ιδιαίτερες ανάγκες, όπως περιγράφηκε παραπάνω. Περιπτώσεις τέτοιων γλωσσών είναι τα σημασιολογικά δίκτυα και οι περιγραφικές λογικές, οι οποίες παρουσιάζονται αναλυτικότερα στη συνέχεια.

## **2.2 Οντολογίες**

Ο όρος οντολογία χρησιμοποιήθηκε για πρώτη φορά στη φιλοσοφία, όπως συμβαίνει αντίστοιχα με πάρα πολλές έννοιες που καταλήγουν να χρησιμοποιούνται καθημερινά στις επιστήμες. Πρόκειται για μία αναφορά στη μελέτη της φύσης και της ουσίας των όντων, τη φιλοσοφική αναζήτηση που εξετάζει τις αρχές της ύπαρξης και συγκρότησης του όντος. Ως “Ον” ορίζεται αυτό που πραγματικά υπάρχει, κάθε τι που διαθέτει υπόσταση. Η ιστορική τους προέλευση προέρχεται από τον κλάδο της φιλοσοφίας που ονομάζεται μεταφυσική, ο οποίος πραγματεύεται τη φύση της πραγματικότητας και της ύπαρξης.

Η πρώτη συνάντηση της έννοιας της οντολογίας με την πληροφορική πραγματοποιήθηκε στο πεδίο της τεχνητής νοημοσύνης, όπου από τα μέσα της δεκαετίας του 1970 αναγνωρίστηκε η ανάγκη σύλληψης και αναπαράστασης της γνώσης για την ανάπτυξη ισχυρών ΑΙ συστημάτων. Ένας πρώτος δημοφιλής και συνοπτικός ορισμός της οντολογίας σε αυτό το περιεχόμενο δόθηκε από τον Tom Gruber στις αρχές της δεκαετίας του 1990 και θα μπορούσε να αποδοθεί στα ελληνικά ως εξής: “Μία οντολογία είναι η προδιαγραφή μιας σύλληψης” (στα αγγλικά: “An ontology is a specification of a conceptualization”).

Στην επιστήμη της πληροφορικής μία οντολογία αναπαριστά φορμαλιστικά τη γνώση ως μία ιεραρχία εννοιών ενός πεδίου ενδιαφέροντος (domain of interest), χρησιμοποιώντας

ένα κοινό λεξιλόγιο με το οποίο δηλώνονται τύποι, ιδιότητες και σχέσεις μεταξύ των εννοιών. Οι οντολογίες αποτελούν δομημένα πλαίσια για την οργάνωση της πληροφορίας και πέρα από την τεχνητή νοημοσύνη εμφανίζονται σε εφαρμογές σημασιολογικού ιστού (semantic web), ανάπτυξης λογισμικού (software engineering), βιοπληροφορικής, βιβλιοθηκονομίας και διάφορες άλλες, ως μία μορφή αναπαράστασης γνώσης σχετικά με τον κόσμο ή κάποιο τμήμα αυτού.

## **2.3 Περιγραφικές Λογικές**

Οι Περιγραφικές Λογικές αποτελούν μία οικογένεια τυπικών γλωσσών αναπαράστασης γνώσης. Οι περισσότερες από αυτές είναι αποφασίσιμα υποσύνολα (decidable fragment) της λογικής πρώτης τάξης. Ανήκουν στις γλώσσες που τελικά χρησιμοποιήθηκαν ως βάση για να καλύψουν τις διάφορες ελλείψεις και προβλήματα των αντίστοιχων μαθηματικών λογικών, παρέχοντας τη δυνατότητα χρήσης τους σε ευρύτερα πεδία εφαρμογών. Διαφέρουν από προγενέστερες γλώσσες παρόμοιου σκοπού, όπως τα σημασιολογικά δίκτυα, καθώς είναι εξοπλισμένες με μία τυπική, βασισμένη στη γνώση σημασιολογία και χρησιμοποιούν ένα δομημένο και αυστηρά ορισμένο τρόπο αναπαράστασης της γνώσης.

Οι Περιγραφικές Λογικές είναι περισσότερο εκφραστικές από τον προτασιακό λογισμό, αλλά παράλληλα παρέχουν μεγαλύτερες δυνατότητες «αποφασισιμότητας» (decidability) συγκριτικά με τη λογική πρώτης τάξης, χαρακτηριστικά τα οποία έχουν συμβάλει στην ευρεία διάδοσή τους. Η ικανότητα εξαγωγής συμπερασμάτων αυξάνει σημαντικά τις δυνατότητες μοντελοποίησης, όμως καθιστά απαραίτητη την ύπαρξη κατάλληλων υπολογιστικών εργαλείων για αυτή την εργασία. Η διαδικασία αυτή ονομάζεται συλλογιστική (reasoning) και βασικός σκοπός των γλωσσών Περιγραφικής Λογικής είναι η εξασφάλιση ύπαρξης σχετικών αλγορίθμων με καλές, ή τουλάχιστον αποδεκτές χρονικές επιδόσεις. Αυτός είναι και ο κύριος λόγος ύπαρξης πολλών Περιγραφικών Λογικών, έτσι ώστε να επιτυγχάνεται ισορροπία μεταξύ εκφραστικότητας και δυνατότητας συλλογιστικής, ανάλογα με τις ανάγκες της εκάστοτε εφαρμογής τους.

### **2.3.1 Βασικά Συστατικά Οντολογιών των Περιγραφικών Λογικών**

Οι Περιγραφικές Λογικές παρέχουν τη δυνατότητα αναπαράστασης σχέσεων μεταξύ οντοτήτων στα πλαίσια ενός πεδίου εφαρμογής και περιλαμβάνουν τα ακόλουθα είδη οντοτήτων:

- Ατομικές Έννοιες (Atomic Concepts)

- Ατομικούς Ρόλους (Atomic Roles) ή Σχέσεις (Relations)
- Άτομα (Individuals)

Οι αντίστοιχες δομές στη λογική πρώτης τάξης είναι τα μοναδιαία κατηγορήματα (unary predicates), τα δυαδικά κατηγορήματα (binary predicates) και ο σταθερές (constants). Σε αντίθεση με τις πιο πολλές Περιγραφικές Λογικές, στη λογική πρώτης τάξης παρέχεται η δυνατότητα κατηγορημάτων με περισσότερα ορίσματα. Οι ατομικές έννοιες αναπαριστούν σύνολα από άτομα, ενώ οι ατομικοί ρόλοι δυικές σχέσεις μεταξύ των ατόμων. Τα άτομα χαρακτηρίζονται από ονόματα (individual names). Σε ένα παράδειγμα οντολογίας που περιγράφει του υπαλλήλους μιας εταιρίας και τις σχέσεις ιεραρχίας μεταξύ τους, θα μπορούσε να οριστεί η έννοια “*Προϊστάμενος*” που συμπεριλαμβάνει το σύνολο των προϊσταμένων στην εταιρία και ο ρόλος “*προϊσταταιΤου*” για να αναπαραστήσει τη δυική σχέση μεταξύ προϊσταμένων της εταιρίας και υφιστάμενών τους. Απλά ατομικά ονόματα όπως “*κόστας*” και “*γιάννης*” μπορούν να χρησιμοποιηθούν για να αναπαραστήσουν τα άτομα.

Σε αντίθεση με τις βάσεις δεδομένων, μία Οντολογία Περιγραφικής Λογικής είναι μία πιο αφηρημένη έννοια που δεν περιγράφει πλήρως την “κατάσταση του κόσμου” αναφοράς. Αντιθέτως, αποτελείται από ένα σύνολο δηλώσεων, τα λεγόμενα αξιώματα, τα οποία πρέπει να ισχύουν στην εκάστοτε περιγραφόμενη κατάσταση. Τα αξιώματα αυτά συνήθως περιλαμβάνουν μόνο ένα μέρος της γνώσης που απεικονίζει η οντολογία, ενώ μπορούν παράλληλα να υπάρχουν και άλλες καταστάσεις οι οποίες είναι συνεπείς στην οντολογία. Αν και από την οπτική γωνία της λογικής δεν υπάρχει ουσιαστική διαφορά μεταξύ διαφορετικών τύπων αξιωμάτων, στις Περιγραφικές Λογικές είναι σύνηθες να διαχωρίζονται στις ακόλουθες κατηγορίες:

### **Αξιώματα Ισχυρισμών (Assertion Axioms)**

Τα αξιώματα ισχυρισμών αναπαριστούν γνώση που αφορά συγκεκριμένα άτομα, τα οποία αναφέρονται με τα ατομικά τους ονόματα. Κατηγοριοποιούνται σε ισχυρισμούς εννοιών και ισχυρισμούς ρόλων. Αν θεωρήσουμε ως  $C$  μία έννοια,  $R$  έναν ρόλο και  $a, b$  δύο άτομα, η τυπική σύνταξη του ισχυρισμού έννοιας είναι  $C(a)$  και του ισχυρισμού ρόλου  $R(a, b)$ . Ένα συγκεκριμένο παράδειγμα ABox αξιώματος, με βάση την οντολογία της εταιρίας που αναφέρθηκε παραπάνω, είναι ο ισχυρισμός ατομικής έννοιας:

Υπάλληλος(γιάννης)

Ο παραπάνω ισχυρισμός υποδηλώνει ότι το άτομο με το όνομα “γιάννης” αποτελεί στιγμιότυπο της έννοιας “Υπάλληλος”.

Ένα αντίστοιχο παράδειγμα ατομικού ρόλου είναι το ακόλουθο:

*προϊσταται*Σε(γιάννης,κώστας)

το οποίο δηλώνει ότι ο Γιάννης προϊσταται του Κώστα ή, πιο φορμαλιστικά, ότι το άτομο με το όνομα “γιάννης” συμμετέχει στη σχέση που αναπαρίσταται από το “*προϊσταται*Σε” με το άτομο με το όνομα “κώστας”.

Επειδή οι Περιγραφικές Λογικές δεν λαμβάνουν την υπόθεση μοναδικού ονόματος (unique name assumption), δύο διαφορετικά ονόματα μπορούν να αναφέρονται στο ίδιο άτομο. Γι αυτό το λόγο υπάρχει ακόμα ο ισχυρισμός ανισότητας ατόμων (individual inequality assumption). Για παράδειγμα:

*γιάννης*  $\neq$  *κώστας*

Σε πλήρη αντιστοιχία με το παραπάνω υπάρχει και ο ισχυρισμός ισότητας ατόμων (individual equality assumption), ο οποίος δηλώνει ότι δύο διαφορετικά ονόματα αναφέρονται στο ίδιο άτομο:

*γιάννης*  $\approx$  *κώστας*

Το σύνολο των αξιωμάτων ισχυρισμών ονομάζεται σώμα ισχυρισμών (Assertion Box – ABox) και συχνά συμβολίζεται ως  $\mathcal{A}$ .

### **Αξιώματα Ορολογίας (Terminological Axioms)**

Τα αξιώματα ορολογίας περιγράφουν σχέσεις μεταξύ εννοιών. Χωρίζονται σε αξιώματα υπαγωγής, τα οποία δηλώνουν ότι μία έννοια είναι γενικότερη από μία άλλη, και σε αξιώματα ισοδυναμίας, τα οποία υποδεικνύουν ότι δύο έννοιες είναι ταυτόσημες. Για παράδειγμα, το γεγονός ότι όλοι οι προϊστάμενοι σε μια εταιρία είναι και υπάλληλοι, δηλαδή ανήκουν στη γενικότερη κατηγορία του υπαλλήλου, μπορεί να εκφραστεί με το αξίωμα υπαγωγής:

*Προϊστάμενος*  $\sqsubseteq$  *Υπάλληλος*

Αυτού του είδους η γνώση μπορεί να χρησιμοποιηθεί για περαιτέρω εξαγωγή συμπερασμάτων σχετικά με τα άτομα. Για παράδειγμα, το παραπάνω αξίωμα σε συνδυασμό με το αξίωμα ισχυρισμού *Προϊστάμενος*(κώστας) οδηγεί στο συμπέρασμα ότι ο Κώστας είναι υπάλληλος.

Τα αξιώματα ισοδυναμίας εκφράζουν το γεγονός ότι δύο έννοιες διαθέτουν τα ίδια ακριβώς στιγμιότυπα, όπως για παράδειγμα:

$$\text{Εργαζόμενος} \equiv \text{Υπάλληλος}$$

Σε αυτήν την περίπτωση δηλώνεται ότι οι δύο διαφορετικές έννοιες περιλαμβάνουν ακριβώς το ίδιο σύνολο ατόμων. Συχνότερα χρησιμοποιούνται πιο περίπλοκες εκφράσεις οι οποίες συνδυάζουν έννοιες για να ορίσουν μία ισοδυναμία, όπως:

$$\text{Προϊστάμενος} \equiv \text{Διευθυντής} \sqcup \text{Υποδιευθυντής} \sqcup \text{Τμηματάρχης}$$

### ***Βάση Γνώσης και Σημασιολογία***

Η βάση γνώσης (knowledge base) μίας Περιγραφικής Λογικής αναπαριστά το εκάστοτε πεδίο ενδιαφέροντος (domain of interest) μέσω ενός συνόλου αξιωμάτων ισχυρισμών, το οποίο αποκαλείται σώμα ισχυρισμών (Assertional Box – ABox) και ενός συνόλου αξιωμάτων ορολογίας το οποίο ονομάζεται σώμα ορολογίας (Terminological Box – TBox). Μία τέτοια βάση γνώσης συμβολίζεται ως:

$$\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$$

Η σημασιολογία (semantics) μιας Περιγραφικής Λογικής δίνεται μέσω τη ερμηνείας της. Σε μια φυσική γλώσσα οι ερμηνείες συνήθως είναι συγκεκριμένες και αντιληπτές από τους ανθρώπους που την ομιλούν. Αντίθετα, στις Περιγραφικές Λογικές και στις γλώσσες αναπαράστασης γνώσης γενικότερα, όταν για παράδειγμα μία έννοια συμβολίζεται με μία λέξη που είναι κατανοητή στη φυσική γλώσσα, δεν σημαίνει απαραίτητα ότι αυτή η έννοια διαθέτει και στην Περιγραφική Λογική μία ερμηνεία ίδια με αυτήν της φυσικής γλώσσας. Η ερμηνεία της έννοιας σε αυτήν την περίπτωση πρέπει να δοθεί από το σχεδιαστή της βάσης γνώσης. Στις Περιγραφικές Λογικές μία ερμηνεία (interpretation) ορίζεται τυπικά από το ζεύγος:

$$(\Delta_I, \cdot^I)$$

όπου το  $\Delta_I$  είναι ένα μη κενό σύνολο που ονομάζεται χώρος ερμηνείας (domain of interpretation) και περιέχει στοιχεία που ονομάζονται αντικείμενα (objects) και  $\cdot^I$  μία συνάρτηση ερμηνείας (interpretation function) που ερμηνεύει κάθε ατομική έννοια  $A$  ως ένα υποσύνολο  $A^I$  του  $\Delta^I$  ( $A^I \subseteq \Delta^I$ ) και κάθε ατομικό ρόλο  $R$  ως ένα υποσύνολο  $R^I$  του  $\Delta^I \times \Delta^I$  ( $R^I \subseteq \Delta^I \times \Delta^I$ ). Η συνάρτηση ερμηνείας επεκτείνεται έτσι ώστε να αποδώσει

ερμηνεία σε διάφορες περιγραφές εννοιών, ανάλογα με την εκάστοτε Περιγραφική Λογική.

### **2.3.2 Η Οικογένεια Περιγραφικών Λογικών DL – Lite**

Ένα από τα βασικά προβλήματα στις Περιγραφικές Λογικές είναι η ισορρόπηση μεταξύ της εκφραστικής ικανότητας και της υπολογιστικής πολυπλοκότητας της ορθής και πλήρους εξαγωγής συμπερασμάτων (sound and complete reasoning). Όπως έχει φανεί μέχρι τώρα, πολλές Περιγραφικές Λογικές με αποδοτικούς αλγόριθμους συλλογιστικής, συγκεκριμένα πολυωνυμικού χρόνου, υστερούν στην εκφραστική ικανότητα που απαιτείται έτσι ώστε να περιλαμβάνουν εννοιολογικά μοντέλα και βασικές οντολογικές γλώσσες, ενώ οι περισσότερες Περιγραφικές Λογικές με επαρκή δυνατότητα μοντελοποίησης εμφανίζουν εκθετικούς χρόνους συλλογιστικής. Η απαίτηση για πολυωνυμικούς χρόνους γίνεται ύψιστης σημασίας όταν ένα οντολογικό σύστημα καλείται να διαχειριστεί μεγάλους όγκους αντικειμένων, όπως χιλιάδες ή εκατομμύρια στιγμιότυπα, κάτι που ισχύει πλέον συχνά στις σύγχρονες εφαρμογές. Βασικό πλεονέκτημα των DL-Lite, το οποίο και οδήγησε στην επιλογή χρήσης τους για αυτήν την εργασία, είναι ότι έχουν αναπτυχθεί μηχανισμοί οι οποίοι αναπαριστούν τους ισχυρισμούς ABox ως σχέσεις που μπορούν διαχειριστούν σε ένα δευτερεύον σύστημα αποθήκευσης, όπως ένα σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (RDBMS).

Οι γλώσσες DL-Lite, όπως παρουσιάστηκαν πρώτη φορά στο [6], είναι μια οικογένεια Περιγραφικών Λογικών που κατασκευάστηκε με σκοπό να περιλαμβάνει βασικές οντολογικές γλώσσες, διατηρώντας ταυτόχρονα χαμηλή πολυπλοκότητα συλλογιστικής. Ο όρος συλλογιστική σε αυτή την περίπτωση σημαίνει όχι μόνο τον υπολογισμό της υπαγωγής μεταξύ των εννοιών και τον έλεγχο της ικανοποιησιμότητας σε ολόκληρη τη βάση γνώσης, αλλά επίσης και τη δυνατότητα απάντησης σε περίπλοκα ερωτήματα (συγκεκριμένα, σε ενώσεις συζευκτικών ερωτημάτων) σε επίπεδο στιγμιοτύπων της DL βάσης γνώσης.

Όπως ισχύει συνήθως στις Περιγραφικές Λογικές, μία βάση γνώσης της οικογένειας DL – Lite αποτελείται από ένα σώμα ορολογίας TBox και ένα σώμα ισχυρισμών ABox, όπου το πρώτο ορίζει γενικές έννοιες και ρόλους και το δεύτερο στιγμιότυπα εννοιών και ρόλων. Η βασική γλώσσα της οικογένειας ονομάζεται DL-Lite<sub>CORE</sub> και παρέχει δυνατότητα έκφρασης ισχυρισμών του τύπου “ISA” πάνω σε έννοιες, περιγραφή ξένων μεταξύ τους εννοιών, τύπους ρόλων, περιορισμούς συμμετοχής σε ρόλους, όπως για παράδειγμα ισχυρισμούς που δηλώνουν ότι όλα τα στιγμιότυπα μίας έννοιας συμμετέχουν σε έναν συγκεκριμένο ρόλο, και περιορισμούς μη συμμετοχής σε ρόλους. Δύο κύριες και σημαντικές επεκτάσεις της DL-Lite<sub>CORE</sub> είναι οι λογικές DL-Lite<sub>F</sub> και

DL-Lite<sub>R</sub>. Η πρώτη προσθέτει στην DL-Lite<sub>CORE</sub> τη δυνατότητα έκφρασης συναρτησιακών περιορισμών σε ρόλους, ενώ η δεύτερη προσθέτει ισχυρισμούς “ISA” και “disjointness” μεταξύ των ρόλων. Η Περιγραφική Λογική πάνω στην οποία βασίστηκε η εφαρμογή που αναπτύχθηκε είναι η DL-Lite<sub>R</sub>, με κάποιους επιπλέον περιορισμούς που θέτει το διαθέσιμο σύστημα επανεγγραφής ερωτημάτων.

Για την παραπάνω οικογένεια Περιγραφικών Λογικών υπάρχουν υλοποιημένες τεχνικές για τις συνήθεις εργασίες συλλογιστικής, όπως η υπαγωγή εννοιών και ρόλων, η ικανοποιησιμότητα της βάση γνώσης και ο έλεγχος στιγμιοτύπων, οι οποίες είναι υπολογιστικά προβλέψιμες (computationally tractable). Συγκεκριμένα, όλες οι παραπάνω εργασίες εμφανίζουν πολυωνυμική πολυπλοκότητα χρόνου σε σχέση με τη βάση γνώσης, ενώ η ικανοποιησιμότητα και ο έλεγχος στιγμιοτύπων παρουσιάζουν πολυπλοκότητα δεδομένων (data complexity) LOGSPACE σε σχέση με το μέγεθος του σώματος ορολογίας ABox.

Οι έννοιες και ο ρόλοι των DL-Lite Περιγραφικών Λογικών σχηματίζονται σύμφωνα με την ακόλουθη σύνταξη:

$$\begin{array}{ll} B \rightarrow A \mid \exists R & R \rightarrow P \mid P^- \\ C \rightarrow B \mid \neg B & E \rightarrow R \mid \neg R \end{array}$$

Στις παραπάνω σχέσεις το  $A$  υποδηλώνει ατομική έννοια, το  $P$  ατομικό ρόλο και το  $P^-$  τον αντίστροφο του ατομικού ρόλου  $P$ . Το  $B$  αναπαριστά μία *βασική έννοια*, δηλαδή μπορεί να είναι είτε μία ατομική έννοια ή μία έννοια της μορφής  $\exists R$ , όπου το  $R$  αναπαριστά ένα *βασικό ρόλο* ο οποίος μπορεί να είναι είτε ένας ατομικός ρόλος ή ο αντίστροφος ενός ατομικού ρόλου. Τέλος, το  $C$  υποδηλώνει μία *γενική έννοια*, η οποία μπορεί να είναι μία *βασική έννοια* ή η άρνησή της και το  $E$  έναν *γενικό ρόλο*, ο οποίος αντίστοιχα μπορεί να είναι ένας *βασικός ρόλος* ή η *άρνησή του*.

Το TBox της DL-Lite<sub>CORE</sub> αποτελείται από ένα πεπερασμένο σύνολο αξιωμάτων υπαγωγής της μορφής:

$$B \sqsubseteq C$$

Άρα, σύμφωνα με τους προηγούμενους ορισμούς, στο δεξί μέρος της σχέσης επιτρέπονται μόνο γενικές έννοιες, ενώ στα αριστερό μέρος μπορούν να εμφανιστούν αποκλειστικά βασικές έννοιες. Κάθε τέτοιο αξίωμα δηλώνει πως όλα τα στιγμιότυπα της



έννοιας  $B$  αποτελούν επίσης και στιγμιότυπα της έννοιας  $C$ . Επιπλέον, στο αριστερό μέρος του αξιώματος υπαγωγής μπορούν να εμφανιστούν κατασκευαστές εννοιών της μορφής  $B_1 \sqcup B_2$ , όπου το σύμβολο  $\sqcup$  αναπαριστά την ένωση, ενώ αντίστοιχα στο δεξί μέρος επιτρέπονται κατασκευαστές της μορφής  $C_1 \sqcap C_2$ , όπου το  $\sqcap$  αναπαριστά την τομή. Αντίστοιχα, μπορεί να τοποθετηθεί η κενή έννοια (bottom concept)  $\perp$  στους κατασκευαστές του αριστερού μέρους του αξιώματος υπαγωγής και η καθολική έννοιας (universal concept)  $\top$  στους κατασκευαστές του δεξιού.

Το  $\text{ABox}$  της  $\text{DL-Lite}_{\text{CORE}}$  σχηματίζεται από ένα πεπερασμένο σύνολο αξιωμάτων ισχυρισμών πάνω σε ατομικές έννοιες και ατομικούς ρόλους, της μορφής:

$$A(a) \qquad P(a, b)$$

τα οποία δηλώνουν ότι το αντικείμενο που συμβολίζεται με τη σταθερά  $a$  είναι ένα στιγμιότυπο της ατομικής έννοιας  $A$  και το ζεύγος αντικειμένων που συμβολίζεται από το ζεύγος σταθερών  $(a, b)$  είναι ένα στιγμιότυπο του ατομικού ρόλου  $P$ .

Στη συνέχεια περιγράφεται η επέκταση της ερμηνείας  $I = (\Delta^I, \cdot^I)$  για τους κατασκευαστές της  $\text{DL-Lite}_{\text{CORE}}$ :

$$A^I \subseteq \Delta^I$$

$$P^I \subseteq \Delta^I \times \Delta^I$$

$$(P^-)^I = \{(x_2, x_1) \mid (x_1, x_2) \in P^I\}$$

$$(\exists R)^I = \{x \mid \exists x'. (x, x') \in R^I\}$$

$$(\neg B)^I = \Delta^I \setminus B^I$$

$$(\neg R)^I = \Delta^I \times \Delta^I \setminus R$$

Μία ερμηνεία  $I$  αποτελεί μοντέλο ενός αξιώματος υπαγωγής  $B \sqsubseteq C$ , αν ισχύει  $B^I \subseteq C^I$ . Η έννοια του μοντέλου μπορεί να επεκταθεί και σε αξιώματα υπαγωγής γενικότερης μορφής, με βάση αυτά που επιτρέπει η  $\text{DL-Lite}_{\text{CORE}}$ . Για παράδειγμα, μία ερμηνεία  $I$  αποτελεί μοντέλο του αξιώματος υπαγωγής  $C_1 \sqsubseteq C_2$ , όπου τα  $C_1, C_2$  αποτελούν γενικές έννοιες, όπως περιγράφηκε παραπάνω, αν ισχύει  $C_1^I \subseteq C_2^I$ . Όμοια ισχύει και στην περίπτωση δύο γενικών ρόλων.

Για τον προσδιορισμό της σημασιολογίας στα αξιώματα ισχυρισμών, η συνάρτηση ερμηνείας επεκτείνεται στα σταθερά ονόματα, αποδίδοντας σε κάθε σταθερά  $a$  ένα μοναδικό αντικείμενο  $a^I \in \Delta^I$ . Αυτό σημαίνει ότι επιβάλλεται η υπόθεση μοναδικού ονόματος στις σταθερές. Μία ερμηνεία  $I$  αποτελεί μοντέλο ενός ισχυρισμού έννοιας  $A(a)$  αν ισχύει  $a^I \in A^I$ . Αντίστοιχα, μία ερμηνεία  $I$  αποτελεί μοντέλο ενός ισχυρισμού ρόλου  $P(a, b)$  αν ισχύει  $(a^I, b^I) \in P^I$ .

Για έναν ισχυρισμό  $\alpha$  και μία ερμηνεία  $I$  η οποία αποτελεί μοντέλο του  $\alpha$ , συμβολίζουμε  $I \models \alpha$ . Για ένα πεπερασμένο σύνολο ισχυρισμών  $\kappa$ , με  $I \models \kappa$  υποδηλώνεται ότι το  $I$  αποτελεί ερμηνεία για κάθε ισχυρισμό στο  $\kappa$ . Μία ερμηνεία  $I$  ονομάζεται μοντέλο μίας βάσης γνώσης  $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$  αν ικανοποιεί τα  $I \models \mathcal{T}$  και  $I \models \mathcal{A}$ . Μία βάση γνώσης λέγεται ικανοποιήσιμη αν διαθέτει τουλάχιστον ένα μοντέλο. Επιπλέον, μία βάση γνώσης  $\mathcal{K}$  συνεπάγεται έναν ισχυρισμό  $\alpha$ , αν όλα τα μοντέλα του  $\mathcal{K}$  είναι επίσης και μοντέλα του  $\alpha$ . Αυτή η περίπτωση συμβολίζεται με  $\mathcal{K} \models \alpha$  ή  $\langle \mathcal{T}, \mathcal{A} \rangle \models \alpha$ . Όμοια, ένα σώμα ορολογίας  $\mathcal{T}$  συνεπάγεται έναν ισχυρισμό  $\alpha$  ( $\mathcal{T} \models \alpha$ ) αν όλα τα μοντέλα του  $\mathcal{T}$  είναι επίσης και μοντέλα του  $\alpha$ .

Η επέκταση της DL-Lite<sub>CORE</sub>, βάσει της οποίας έχει αναπτυχθεί στο σύστημα επανεγγραφής ερωτημάτων στο οποίο βασίστηκε η ανάπτυξη της εφαρμογής που θα παρουσιαστεί στη συνέχεια του κειμένου, είναι η DL-Lite<sub>R</sub>. Αυτή η Περιγραφική Λογική παρέχει τη δυνατότητα δημιουργίας αξιωμάτων υπαγωγής της μορφής:

$$R \sqsubseteq E$$

όπου  $R$  και  $E$  είναι ρόλοι που έχουν οριστεί παραπάνω. Η DL-Lite<sub>R</sub> μπορεί επίσης να θεωρηθεί ως μία επέκταση του τμήματος της οντολογικής γλώσσας RDFS, το οποίο προσομοιάζει στις Περιγραφικές Λογικές, παρέχοντας επιπλέον τη δυνατότητα ορισμού ξένων μεταξύ τους εννοιών και ρόλων.

### **Ερωτήματα στις DL-Lite**

Το είδος των ερωτημάτων που μπορεί να εφαρμοστεί στις DL-Lite είναι συζευκτικά ερωτήματα (conjunctive queries) και η ένωση αυτών. Ένα συζευκτικό ερώτημα  $q$  έχει την παρακάτω μορφή:

$$\{\vec{x} | \exists \vec{y}. conj(\vec{x}, \vec{y})\}$$

Όπου  $conj(\vec{x}, \vec{y})$  είναι μία σύζευξη από άτομα και ισότητες, με ελεύθερες μεταβλητές τα  $\vec{x}$  και  $\vec{y}$ . Μία ένωση από συζευκτικά ερωτήματα έχει τη μορφή:

$$\left\{ \vec{x} \mid \bigvee_{i=1, \dots, n} \exists \vec{y}_i. conj_i(\vec{x}, \vec{y}_i) \right\}$$

όπου  $conj_i(\vec{x}, \vec{y}_i)$  όπως και πριν. Σε συνέχεια των παραπάνω, ένα συζευκτικό ερώτημα, έστω  $q$ , πάνω σε μία βάση γνώσης  $\mathcal{K}$  περιέχει άτομα της μορφής  $A(z)$  και  $P(z_1, z_2)$ , όπου τα  $A$  και  $P$  αντιστοιχούν σε μία ατομική έννοια και έναν ατομικό ρόλο της  $\mathcal{K}$ , ενώ τα  $z, z_1, z_2$  μπορούν να είναι είτε σταθερές της  $\mathcal{K}$  ή μεταβλητές. Η απάντηση του  $q$  με βάση την  $\mathcal{K}$  είναι το σύνολο  $answer(q, \mathcal{K})$  των πλειάδων (tuples)  $\vec{a}$  σταθερών που εμφανίζονται στη βάση γνώσης  $\mathcal{K}$  και για τις οποίες ισχύει  $\vec{a}^M \in q^M$ , για κάθε μοντέλο  $M$  του  $\mathcal{K}$ . Προφανώς το σύνολο  $answer(q, \mathcal{K})$  είναι πεπερασμένο, δεδομένου ότι το ίδιο ισχύει και για την  $\mathcal{K}$ .

Όπως έχει ήδη αναφερθεί στην αρχή της παραγράφου, τα προβλήματα της ικανοποιήσιμότητας και της λογικής συνεπαγωγής των ισχυρισμών της βάσης γνώσης εμφανίζουν αποδεδειγμένα πολύ καλή πολυπλοκότητα χρόνου και δεδομένων. Όσον αφορά την πολυπλοκότητα απάντησης συζευκτικών ερωτημάτων, κάθε τέτοιο ερώτημα μπορεί να μετασχηματιστεί σε μία ένωση απλών ερωτημάτων τα οποία αποτιμώνται με βάση το ABox. Η αναδόμηση του ερωτήματος γίνεται με βάση τα αξιώματα του TBox, το οποίο στη συνέχεια παύει να είναι χρήσιμο. Εάν πράγματι το ABox αναπαρίσταται από μία βάση δεδομένων, η πολυπλοκότητα χρόνου είναι η ίδια με αυτήν της εκτέλεσης του ερωτήματος στο σχετικό RDBMS. Όσο για την πολυπλοκότητα των δεδομένων, στην περίπτωση συζευκτικού ερωτήματος, τούτο μετασχηματίζεται σε ένωση αντίστοιχων ερωτημάτων, άρα η εν λόγω πολυπλοκότητα είναι πολυωνυμική.

## 2.4 Η Γλώσσα OWL

Η γλώσσα OWL (Web Ontology Language) αποτελεί ένα σημαντικό εργαλείο για την ανάπτυξη εφαρμογών του σημασιολογικού ιστού, μίας νέας μορφής του παγκόσμιου ιστού (World Wide Web – WWW) όπου η πληροφορία οργανώνεται με τρόπο κατανοητό από τους υπολογιστές, παρέχοντας τη δυνατότητα σε προγράμματα που ονομάζονται πράκτορες (agents) να διεκπεραιώνουν πολύπλοκες, σημασιολογικά ορισμένες εργασίες. Πρόκειται για μία γλώσσα ανάπτυξης οντολογιών που χρησιμοποιεί ως λογικό υπόβαθρο τις Περιγραφικές Λογικές.

Όπως ισχύει και στις Περιγραφικές Λογικές, η OWL στην πραγματικότητα αποτελεί μία οικογένεια γλωσσών με διαφορετική εκφραστική ικανότητα και πολυπλοκότητα εξαγωγής συμπερασμάτων. Στις αρχικές προδιαγραφές της περιελάμβανε τις τρεις ακόλουθες υπογλώσσες:

- Την OWL Lite, η οποία προσφέρει αρκετές δυνατότητες όπως τη δημιουργία ταξινομήσεων και περιορισμών, αλλά δεν υποστηρίζει το σύνθετο σύνολο μίας γλώσσας οντολογίας
- Την OWL DL (Description Logic), η οποία προσφέρει τη δυνατότητα πολύπλοκων σχέσεων και εκφράσεων, εξασφαλίζοντας παράλληλα ότι η αυτόματη εφαρμογή κανόνων λογικής (reasoning) τερματίζει σε πεπερασμένο χρόνο. Λόγω αυτών των ιδιοτήτων αποτέλεσε την πιο δημοφιλή από τις τρεις γλώσσες
- Την OWL Full, η οποία είναι βασισμένη σε διαφορετική σημασιολογία από τις δύο προηγούμενες, επιτρέποντας το σχηματισμό πιο πολύπλοκων σχέσεων και συνθέσεων μεταξύ κλάσεων και ιδιοτήτων, δίνοντας τη δυνατότητα να δημιουργηθούν περιορισμοί κάθε είδους. Η γλώσσα αυτή είναι μη αποφασίσιμη, με αποτέλεσμα να μην υπάρχει δυνατότητα δημιουργίας λογισμικού που να μπορεί να εκτελέσει αυτόματο reasoning

Το World Wide Web Consortium (W3C) τον Οκτώβριο του 2009 ανακοίνωσε την νέα έκδοση της OWL, την OWL 2, όπου περιλαμβάνονται και πάλι τρεις υπογλώσσες, τα επονομαζόμενα προφίλ (profiles): Η OWL 2 EL που χαρακτηρίζεται από πολυωνυμικό χρόνο συλλογιστικής, η OWL QL που σχεδιάστηκε έτσι ώστε να παρέχει ευκολότερη πρόσβαση για την εφαρμογή ερωτημάτων σε βάσεις δεδομένων, και η OWL RL που αποτελεί ένα υποσύνολο κανόνων της OWL 2.

Σημειώνεται ότι η DL-Lite<sub>F</sub> αποτελεί ένα αυστηρό υποσύνολο της γλώσσας οντολογιών OWL, και πιο συγκεκριμένα της OWL Lite, ενώ η DL-Lite<sub>R</sub> αποτελεί το λογικό υπόβαθρο της γλώσσας OWL 2 QL, η οποία είναι ένα γνωστό βατό προφίλ της OWL 2

# 3

## *Πολύπλευρη Αναζήτηση (Faceted Search)*

### *3.1 Γενικά για τα Facets και το Διαδίκτυο*

Η λέξη “facet” στα αγγλικά χρησιμοποιείται σε κυριολεκτικούς όρους για να περιγράψει τις μικρές επίπεδες επιφάνειες που σχηματίζονται κατά την επεξεργασία των πολύτιμων λίθων. Ένας άλλος πιο ταιριαστός στην περίπτωσή μας ορισμός που εμφανίζεται στα λεξικά της αγγλικής γλώσσας είναι οι διαφορετικές όψεις ή πλευρές που συνθέτουν κάποιο θέμα.

Στην ακαδημαϊκή κοινότητα ο όρος “faceted search” αρχικά εμφανίστηκε στην επιστήμη της βιβλιοθηκονομίας, η οποία έχει σαν σκοπό την οργάνωση της πληροφορίας και της γνώσης, έτσι ώστε να γίνεται πιο εύκολη και ευέλικτη η πρόσβαση σε αυτή από τους χρήστες. Πρόκειται για μία τεχνική που επιτρέπει την πλοήγηση σε μία συλλογή πληροφοριών μέσω της εφαρμογής πολλαπλών φίλτρων. Η πληροφορία είναι οργανωμένη σύμφωνα με ένα πολύπλευρο (faceted) σύστημα ταξινόμησης, το οποίο κατατάσσει τα ποικίλα στοιχεία της σε διάφορες σαφείς και διακριτές διαστάσεις, τα facets. Με αυτόν τον τρόπο οι ταξινομήσεις καθίστανται προσβάσιμες με πολλούς διαφορετικούς τρόπους αντί για μία προκαθορισμένη διάταξη.

Όπως ήταν αναμενόμενο, η faceted αναζήτηση πολύ γρήγορα επεκτάθηκε και στην επιστήμη της πληροφορικής. Μέσω αυτής δόθηκε η δυνατότητα στο χρήστη μίας εφαρμογής αναζήτησης να περιορίζει τα επιστρεφόμενα αποτελέσματα, με τον

καθορισμό ενός πλήθους διακριτών χαρακτηριστικών, τα λεγόμενα facets. Κάθε facet αντικατοπτρίζει μία συγκεκριμένη προοπτική στα περιεχόμενα των αποτελεσμάτων, η οποία συνήθως είναι ξεκάθαρα διαχωρισμένη από τις υπόλοιπες. Οι επιλογές ενός facet μπορεί να αποτελούνται από μία απλή επίπεδη λίστα όπου μόνο μία επιλογή επιτρέπεται, ή από μία ιεραρχική λίστα στην οποία ο χρήστης μπορεί να πλοηγηθεί σε διάφορα επίπεδα κάνοντας drill down.

Η κύρια δύναμη της faceted αναζήτησης εντοπίζεται στη δυνατότητα που παρέχει στο χρήστη να καθορίζει μόνος του τις παραμέτρους πλοήγησης, συνδυάζοντας τις διάφορες οπτικές που τον ενδιαφέρουν. Γι αυτό το λόγο η faceted αναζήτηση είναι εδώ και αρκετά χρόνια ιδιαίτερα δημοφιλής στο διαδίκτυο και εμφανίζεται σε πληθώρα εμπορικών και όχι μόνο δικτυακών εφαρμογών. Χαρακτηριστικά παραδείγματα αποτελούν ιστοσελίδες πολύ μεγάλης επισκεψιμότητας όπως το booking.com, όπου ο χρήστης έχει τη δυνατότητα να περιορίσει (narrow down) τα αποτελέσματα για την αναζήτηση καταλύματος, εφαρμόζοντας φίλτρα σε διάφορες κατηγορίες όπως την κατηγορία του ξενοδοχείου και τη βαθμολογία που αυτό έχει λάβει από άλλους χρήστες. Επίσης η faceted αναζήτηση εμφανίζεται σε πολλές ιστοσελίδες πώλησης προϊόντων, καθώς και σε αντίστοιχες πύλες (portals) που συγκεντρώνουν σχετικά στοιχεία από πολλά διαφορετικά sites, όπως το πολύ γνωστό και δημοφιλές στους Έλληνες χρήστες skroutz.gr, όπου ο χρήστης ανάλογα με το προϊόν που αναζητά μπορεί να περιορίσει τα εμφανιζόμενα αποτελέσματα σύμφωνα με το εύρος τιμών, τον τύπο και διάφορα άλλα σχετιζόμενα με το είδος χαρακτηριστικά.

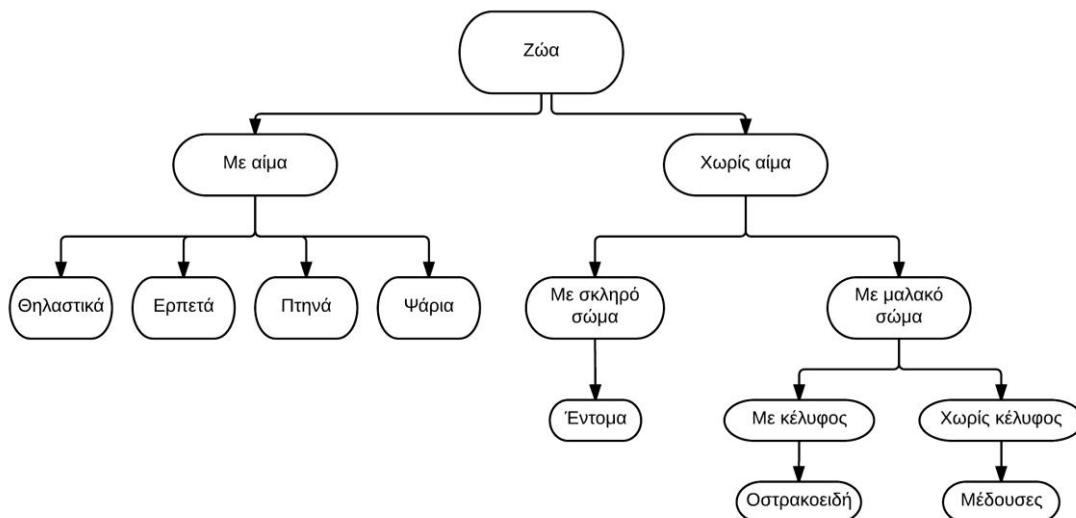
Η faceted αναζήτηση δεν περιορίζεται μόνο σε εμπορικές εφαρμογές. Τα τελευταία χρόνια έχει κάνει την εμφάνισή της και σε ιστοσελίδες που περιέχουν μεγάλο όγκο πληροφορίας, όπως σελίδες εφημερίδων και μέσω μαζικής ενημέρωσης (π.χ. The Financial Times), βιβλιοθήκες (NSCU Libraries) και μη κερδοσκοπικούς οργανισμούς (Urban Land Institute). Σε πολλές από αυτές τις περιπτώσεις συνδυάζεται με πιο κλασικές μεθόδους αναζήτησης, όπως η αναζήτηση κειμένου. Ο χρήστης εισάγει αρχικά κάποια λέξη κλειδί και στη συνέχεια περιορίζει τα αποτελέσματα βάσει των επιλογών που του παρέχονται από τα facets. Δύο γνωστές πλατφόρμες για τη δημιουργία μηχανών αναζήτησης που υποστηρίζουν faceted πλοήγηση είναι η *Apache Solr* και η *Elasticsearch*. Και οι δύο αυτές πλατφόρμες έχουν αναπτυχθεί στη γλώσσα προγραμματισμού Java και βασίζονται στο ανοιχτού κώδικα λογισμικό ανάκτησης πληροφοριών *Apache Lucene*.

## 3.2 Ταξονομίες, Faceted Ταξινόμηση και Οντολογίες

Ο Αριστοτέλης έχει περιγραφεί κατά το παρελθόν ως “ίσως ο τελευταίος άνθρωπος που γνώριζε οτιδήποτε μπορεί να γνωρίζει κάποιος στον καιρό του”. Δημιούργησε το πρώτο πλήρες και ολοκληρωμένο φιλοσοφικό σύστημα, συμπεριλαμβάνοντας διάφορες επί μέρους κοινωνικές εκφράσεις και ανθρώπινα ενδιαφέροντα όπως η ηθική, η αισθητική, η λογική, οι φυσικές επιστήμες και η πολιτική. Επιπλέον, ήταν ίσως ο πρώτος φιλόσοφος που καθιέρωσε ένα πλαίσιο για αυτή τη συλλογική γνώση του ανθρώπινου είδους. Όπως αναφέρει ο άγγλος μαθηματικός και φιλόσοφος Alfred North Whitehead, “αν όλη η φιλοσοφία είναι μία αλληλουχία υποσημειώσεων και αναφορών στον Πλάτωνα, τότε το σύνολο της θεωρίας και της πρακτικής της αναπαράστασης γνώσης είναι μία σειρά αναφορών στον Αριστοτέλη”.

### 3.2.1 Ταξονομίες

Ο Αριστοτέλης ήταν ο πρώτος επιστήμονας ταξινομίας της ιστορίας και αυτός ο ρόλος του είναι γνωστός σε όλους τους σύγχρονους ταξινομείς που ασχολούνται επαγγελματικά με την ιεραρχική οργάνωση και αναπαράσταση της γνώσης. Το σύστημα για την ταξινόμηση των έμβιων όντων που ανέπτυξε διαχωρίζει τους οργανισμούς σε δύο κύριες κατηγορίες, τα φυτά και τα ζώα. Τα ζώα διαχωρίζονται περαιτέρω σε αυτά που κυλάει αίμα μέσα στο σώμα τους και σε όσα δεν έχουν αίμα. Η πρώτη κατηγορία διαχωρίζεται σε αυτά που γεννιούνται απ’ ευθείας από τον μητρικό οργανισμό και σε όσα έρχονται στη ζωή μέσω αυγού κ.ο.κ. Στο παρακάτω σχήμα αναπαρίσταται ένα υποσύνολο του εν λόγω συστήματος κατηγοριοποίησης:



Η έννοια της ταξονομίας αρχικά χρησιμοποιήθηκε για την κατηγοριοποίηση έμβιων όντων, όπως αυτή που περιγράφηκε παραπάνω. Στις ημέρες μας ο όρος χρησιμοποιείται ευρύτερα, για την περιγραφή κάθε ιεραρχικά ταξινομημένου σχήματος, όπως και για τις αρχές που διέπουν τέτοιου είδους σχήματα. Βάσει της σύγχρονης χρήσης της, ταξονομία αποκαλείται κάθε οργάνωση αντικειμένων ή αφηρημένων εννοιών σε μία ιεραρχία ή δενδρική δομή. Κάνοντας χρήση της ορολογίας των χαρακτηριστικών ενός δένδρου, στην πράξη ανεστραμμένου, μία ταξονομία αποτελείται από έναν κόμβο – ρίζα στο πάνω μέρος της, κόμβους – φύλλα στο κάτω μέρος, και διάφορα κλαδιά που συνδέουν κάθε κόμβο – γονέα που δεν είναι φύλλο με άλλους κόμβους που αποτελούν τα παιδιά του. Κάθε κόμβος μπορεί να διαθέτει περισσότερα από ένα παιδιά αλλά έχει μόνο έναν γονέα, με εξαίρεση τη ρίζα που δεν έχει γονέα.

Ο κόμβος – ρίζα μίας ταξονομίας αναπαριστά μία κατάταξη που περιλαμβάνει το πλήρες σύνολο των αντικειμένων του περιγραφόμενου κόσμου. Για παράδειγμα, στην παραπάνω αριστοτέλεια ταξονομία, ο κόμβος – ρίζα αντιστοιχεί στο σύνολο όλων των έμβιων όντων. Τα παιδιά του αναπαριστούν έναν διαχωρισμό πρώτου επιπέδου αυτού του συνόλου, τα παιδιά τους υποδιαιρέσεις αυτών κ.ο.κ., καταλήγοντας στους κόμβους – φύλλα οι οποίοι αναπαριστούν τα ίδια τα αντικείμενα.

Μία πολύ σημαντική ιδιότητα των ταξονομιών είναι ότι για κάθε αντικείμενο ή σύνολο αντικειμένων που αντιστοιχεί σε έναν κόμβο, υπάρχει ένα και μοναδικό μονοπάτι από τη ρίζα προς τον κόμβο αυτό. Έτσι, η ταξονομία επιβάλλει μία αυστηρή λογική αλληλουχία στη γνώση που αναπαριστά, η οποία σε κάποιες περιπτώσεις μπορεί να θεωρηθεί προβληματικά άκαμπτη. Όσο η γνώση μεγαλώνει σε όγκο και πολυπλοκότητα, η απαίτηση για ένα μοναδικό μονοπάτι από τη ρίζα προς κάθε κόμβο αποτελεί έναν ολοένα και σκληρότερο περιορισμό, οδηγώντας αναγκαστικά σε σολομώντειες λύσεις τους διάφορους σχεδιαστές ταξονομιών, καθώς αναγκάζονται να επιλέξουν μόνο μία κλάση – γονέα για κάθε υποκλάση κατά το σχεδιασμό της ιεραρχίας, σε περιπτώσεις όπου στις έννοιες του περιγραφόμενου κόσμου πιθανόν να αρμόζει κάτι πιο περίπλοκο.

Μία λύση στο παραπάνω πρόβλημα μπορούν να προσφέρουν οι πολλαπλές ιεραρχίες. Πολλαπλή ιεραρχία ονομάζεται μία αναπαράσταση αντίστοιχη της ταξονομίας, στην οποία κάποιος κόμβος μπορεί να διαθέτει περισσότερους από έναν γονείς. Δυστυχώς οι πολλαπλές ιεραρχίες προκαλούν πιθανόν περισσότερα προβλήματα από όσα επιλύουν, κυρίως σε αυτούς που έχουν το καθήκον να τις περιγράψουν, να τις επεκτείνουν και να τις συντηρούν. Η διαρκής συντήρηση και επέκταση μίας ταξονομίας αποτελεί ήδη μία ιδιαίτερα επίπονη εργασία, ενώ στην περίπτωση των πολλαπλών ιεραρχιών η δυσκολία αυξάνεται εκθετικά. Γι αυτό το λόγο, στις περιπτώσεις που παρουσιάζεται η ανάγκη για



περιγραφή σύνθετων εννοιών έχουν αναπτυχθεί πιο αποδοτικές μέθοδοι όπως η πολύπλευρη (faceted) ταξινόμηση και οι οντολογίες που περιγράφονται στη συνέχεια.

### **3.2.2 Faceted Ταξινόμηση και Οντολογίες**

Όπως έχει ήδη αναφερθεί, η ιδέα των facets έχει γεννηθεί από την επιστήμη της βιβλιοθηκονομίας. Το πρώτο εξελιγμένο σύστημα κατηγοριοποίησης των καταλόγων των βιβλιοθηκών βασίστηκε στη δημιουργία μίας ταξονομίας, αναπτύχθηκε από τον Melvil Dewey και ονομάζεται Dewey Decimal Classification (DDC). Πρόκειται για ένα δημοφιλές σύστημα που χρησιμοποιείται από πολλές βιβλιοθήκες ακόμα και σήμερα. Κάθε κόμβος της ταξονομίας αποτελείται από μία σειρά δεκαδικών ψηφίων, όπου το κάθε διαδοχικό ψηφίο αντιστοιχεί μέσω κάποιου κλαδιού σε επόμενο κόμβο του δέντρου της ιεραρχίας. Όπως γίνεται εύκολα αντιληπτό, σε μία μεγάλη βιβλιοθήκη είναι πρακτικά σίγουρο ότι κάποιες θεματικές κατηγορίες θα μπορούν βάσει λογικής να καταταχθούν κάτω από διαφορετικά κλαδιά της ταξονομίας. Για παράδειγμα, σε μία ιεραρχία όπου τα βιβλία ταξινομούνται τόσο βάσει της γεωγραφικής περιοχής στην οποία αναφέρονται, όσο και σύμφωνα με το σχετιζόμενο επιστημονικό αντικείμενο, ένα βιβλίο με θέμα την ελληνική ιστορία θα μπορούσε να καταταχθεί κάτω και από τους δύο σχετικούς κόμβους. Όμως ο περιορισμός του μοναδικού μονοπατιού επιβάλλει μία και μόνη κατάταξη, κάτι που στο συγκεκριμένο παράδειγμα αποτελεί καίριο πρόβλημα.

Μία λύση στο παραπάνω πρόβλημα προτάθηκε από τον ινδό μαθηματικό και βιβλιοθηκονόμο Shiyali Ramamrita Ranganathan, ο οποίος εισήγαγε έναν νέο συμβολισμό για τις θεματικές κατηγοριοποιήσεις που έδινε τη δυνατότητα αναπαράστασης μίας γενικής κατηγορίας μέσω της σύνθεσης διαφόρων εννοιών, κάτι που στην ιεραρχική ταξονομία δεν ήταν εφικτό λόγω του προαναφερθέντος περιορισμού του μοναδικού μονοπατιού. Πρόκειται για ένα σχήμα ταξινόμησης με χρήση του συμβόλου της άνω κάτω τελείας (colon), μέσω της οποίας διαχωρίζονται ξεχωριστές θεματικές έννοιες, οι οποίες τελικά συνδυάζονται για να σχηματίσουν την τελική περιγραφή κάποιας πολύπλοκης έννοιας. Μέσω του συμβόλου της άνω κάτω τελείας ουσιαστικά διαχωρίζονται τα επί μέρους facets, καθένα από τα οποία αποτελεί ξεχωριστή ιεραρχία μίας επί μέρους ομάδας χαρακτηριστικών. Τα facets συνδυαζόμενα οδηγούν στην περιγραφή των σύνθετων εννοιών. Ο ίδιος ο Ranganathan δίνει ένα παράδειγμα για την περιγραφή του θέματος “της στατιστικής μελέτης της θεραπείας του καρκίνου της μαλακής υπερώας με χρήση ραδίου”. Τα 4 ιεραρχικά facets που συνθέτουν το παραπάνω θέμα είναι τα ακόλουθα:

*Ιατρική (L) → Πεπτικό σύστημα (L2) → Στόμα (L22) → Υπερώα (L215)  
→ Μαλακή υπερώα (L2153)*

*Ασθένεια (4) → Είδος ασθένειας (47) → Όγκος (472) → Καρκίνος (4725)*

*Θεραπεία (6) → Χημειοθεραπεία (63) → Θεραπεία με χημικό στοιχείο (631)  
→ Θεραπεία με χημικό στοιχείο της ομάδας 2 (6312)  
→ Θεραπεία με ράδιο (63129)*

*Μαθηματική μελέτη (B) → Αλγεβρική μελέτη (B2) → Στατιστική μελέτη (B28)*

Η αναπαράσταση για αυτό το σύνθετο θέμα είναι η:

*L2153:4725:63129:B28*

Η παραπάνω faceted αναπαράσταση αποτελεί την έμπνευση για διάφορες σύγχρονες προσεγγίσεις στη βιβλιοθηκονομία και την κατασκευή λεξικών – θησαυρών, το κυριότερο όμως πεδίο εφαρμογής αυτής της ιδέας βρίσκεται στην επιστήμη της πληροφορικής, αφού οδήγησε σε πληθώρα εφαρμογών που σχετίζονται με τη faceted αναζήτηση και την αναπαράσταση γνώσης, μερικές εκ των οποίων περιγράφηκαν σε προηγούμενες ενότητες.

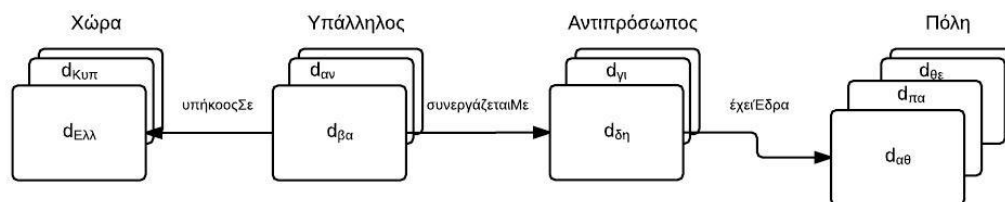
Οι οντολογίες έρχονται να συμπληρώσουν διάφορες εκφραστικές αδυναμίες που εμφάνιζε η faceted ταξινόμηση. Στην πράξη, μία ταξινόμηση αποτελεί ειδική περίπτωση οντολογίας, όπου οι μόνες επιτρεπτές σχέσεις είναι αυτές του είδους “is – a” που συνδέουν τους κόμβους παιδιά με τους γονείς τους. Όπως αναφέρθηκε παραπάνω, ένα faceted σύστημα ταξινόμησης περιλαμβάνει μία συλλογή από facets, καθένα από τα οποία είναι μία ταξινόμηση. Το συγκριτικό πλεονέκτημα των οντολογιών είναι ότι επιτρέπουν περισσότερους και πιο σύνθετους τύπους σχέσεων. Για παράδειγμα, μπορεί κάποιος μέσω μίας οντολογίας να πει ότι *η Αθήνα είναι πρωτεύουσα της Ελλάδας*, όπου οι έννοιες “Αθήνα” και “Ελλάδα” συνδέονται μέσω της σχέσης “είναι–πρωτεύουσα”. Στις οντολογίες, μία σχέση είναι απλά μία πλειάδα που περιλαμβάνει τα σχετιζόμενα αντικείμενα και το είδος της σχέσης. Μία οντολογία μπορεί να περιλαμβάνει πλειάδες που συνδυάζουν πάνω από δύο αντικείμενα και περισσότερα είδη σχέσεων. Στην πράξη όμως, λόγω της περιττής πολυπλοκότητας που μπορεί να προκύψει, οι περισσότερες υλοποιήσεις οντολογιών χρησιμοποιούν τριάδες (δύο αντικείμενα και μία σχέση) και με συνδυασμό αυτών αναπαριστούν τις πιο περίπλοκες έννοιες.

### 3.3 Faceted Αναζήτηση σε Βάσεις Γνώσης – Φορμαλιστική Περιγραφή

Όπως έχει ήδη αναφερθεί, η faceted αναζήτηση είναι μία ιδιαίτερα επιτυχημένη προσέγγιση αναζήτησης σε διάφορες συλλογές εγγράφων και δεδομένων, παρέχοντας τη δυνατότητα στους τελικούς χρήστες να περιορίζουν τα επιστρεφόμενα αποτελέσματα μέσω της σταδιακής εφαρμογής φίλτρων, τα επονομαζόμενα facets, στα διάφορα χαρακτηριστικά των δεδομένων. Η faceted αναζήτηση είναι μία εξαιρετικά δημοφιλής τεχνολογία, η οποία βρίσκει εφαρμογή όλο και πιο συχνά σε ιστοσελίδες ηλεκτρονικού εμπορίου. Η συνήθης πρακτική βασίζεται ουσιαστικά στη διαδοχική εφαρμογή αυτών των φίλτρων (facets) πάνω στις τιμές κάποιων πεδίων μίας κλασικής σχεσιακής βάσης δεδομένων, μέσω ενός SQL ερωτήματος, περιορίζοντας και εξειδικεύοντας κατ' αυτόν τον τρόπο τα αποτελέσματα της αναζήτησης.

Η faceted αναζήτηση έχει προταθεί ως ένα κατάλληλο μοντέλο για την πραγματοποίηση ερωτημάτων σε συλλογές εγγράφων των οποίων τα χαρακτηριστικά απεικονίζονται μέσω της γλώσσας RDF (Resource Description Framework), ενώ έχουν αναπτυχθεί και αρκετά σχετικά συστήματα αναζήτησης. Μία ακόμα πιο πρόσφατη εξέλιξη [8] είναι η μελέτη εφαρμογής τέτοιων ερωτημάτων σε οντολογίες που περιγράφονται μέσω της γλώσσας OWL 2. Στη συνέχεια ακολουθεί μία φορμαλιστική περιγραφή της faceted διεπαφής (interface) και των σχετικών ερωτημάτων που μπορούν να εφαρμοστούν σε αυτήν, στα πλαίσια της faceted αναζήτησης.

Για τις ανάγκες της περιγραφής θεωρούμε μία υποθετική βάση γνώσης που αναπαριστά τους υπαλλήλους και τους συνεργάτες μίας εταιρίας που δραστηριοποιείται σε διάφορες περιοχές της Ελλάδας, μέρη της οποίας απεικονίζονται συνοπτικά παρακάτω:



Στο παραπάνω σχήμα, τα  $d_{\alpha\nu}$  και  $d_{\beta\alpha}$  συμβολίζουν τα άτομα Αντρέα και Βασίλη, οι οποίοι αποτελούν αντικείμενα της έννοιας “Υπάλληλος”. Επιπλέον, στην οντολογία περιλαμβάνεται η έννοια “Αντιπρόσωπος”, στιγμιότυπα της οποίας είναι τα άτομα Γιώργος και Δημήτρης που απεικονίζονται ως  $d_{\gamma\iota}$  και  $d_{\delta\eta}$ . Ο υπάλληλος Αντρέας συνεργάζεται με τον αντιπρόσωπο Γιώργο και ο υπάλληλος Βασίλης με τον αντιπρόσωπο

Δημήτρη. Τα αντικείμενα Αθήνα ( $d_a$ ), Θεσσαλονίκη ( $d_\theta$ ) και Πάτρα ( $d_\pi$ ) ανήκουν στην έννοια “Πόλη”, ενώ η Ελλάδα ( $d_{Ell}$ ) και η Κύπρος ( $d_{Kυπ}$ ) στην έννοια “Χώρα”. Στο παράδειγμα που ακολουθεί σκοπός είναι να εντοπιστούν οι υπάλληλοι της εταιρίας που είναι έλληνες ή κύπριοι υπήκοοι και συνεργάζονται με κάποιον αντιπρόσωπο που έχει έδρα στη Θεσσαλονίκη.

Τα facets αναπαρίστανται ως ζεύγη που αποτελούνται από ένα όρισμα – το όνομα του facet – και ένα σύνολο τιμών, οι οποίες είναι τα ονόματα των αντικειμένων που συνήθως συμβολίζονται μέσω ενός URI. Στην εξεταζόμενη περίπτωση, παραδείγματα ονομάτων facets είναι οι δυαδικές σχέσεις “συνεργάζεταιΜε” και “έχειΈδρα”, ενώ αντίστοιχα παραδείγματα τιμών γι αυτά τα facets αποτελούν συγκεκριμένα άτομα όπως το “ $d_a$ ” (Αθήνα). Η επιλογή πολλαπλών τιμών μέσα σε ένα facet μπορεί να ερμηνευθεί συζευκτικά ή διαζευκτικά, γι αυτό και ορίζονται δύο διακριτές κατηγορίες facets. Επιπλέον, ορίζεται ένας ειδικός τύπος facet, του οποίου οι τιμές αποτελούνται από έννοιες (μοναδιαία κατηγορήματα) αντί για συγκεκριμένα άτομα. Τέλος, γίνεται χρήση μίας ειδικής τιμής η οποία υποδηλώνει το σύνολο όλων των δυνατών τιμών που είναι συμβατές με το όρισμα του εκάστοτε facets.

Υποθέτουμε ότι τα  $C, UP$  και  $BP$  είναι μεταξύ τους ξένα μη πεπερασμένα σύνολα που αποτελούνται από σταθερές, μοναδιαία κατηγορήματα και δυαδικά κατηγορήματα αντίστοιχα.

**Ορισμός 1:** Έστω τα σύμβολα  $type$  και  $any$  τα οποία δεν υπάρχουν στο σύνολο  $C \cup UP \cup BP$ . Ως facet ορίζεται το ζεύγος  $(X, \circ \Gamma)$ , όπου  $\circ \in \{\wedge, \vee\}$ ,  $\Gamma$  ένα μη κενό σύνολο και ισχύει ένα από τα ακόλουθα:

$$X = type \text{ και } \Gamma \subseteq UP, \text{ ή}$$

$$X \in BP, any \in \Gamma \text{ και είτε } \Gamma \subseteq C \cup \{any\} \text{ ή } \Gamma \subseteq UP \cup \{any\}$$

Ένα facet της μορφής  $(X, \wedge \Gamma)$  ονομάζεται συζευκτικό, ενώ ένα facet της μορφής  $(X, \vee \Gamma)$  διαζευκτικό. Για ένα facet  $F = (X, \circ \Gamma)$ , το  $X$  είναι το όνομά του, το οποίο συμβολίζεται ως  $F|_1$ , και το  $\Gamma$  περιέχει τις τιμές του και συμβολίζεται ως  $F|_2$ . Τα ακόλουθα facets μπορούν να χρησιμοποιηθούν για αναζητήσεις στο συγκεκριμένο παράδειγμα:

$$F_1 = (type, \vee \{Υπάλληλος, Αντιπρόσωπος, Πόλη, Χώρα\})$$

$$F_2 = (συνεργάζεταιΜε, \vee \{any, d_{γυ}, d_{ση}\})$$

$$F_3 = (\acute{\epsilon}\chi\epsilon\iota\acute{\epsilon}\delta\rho\alpha\Sigma\epsilon, \vee \{d_{\alpha\theta}, d_{\theta\epsilon}, d_{\pi\alpha}\})$$

$$F_4 = (\upsilon\pi\acute{\eta}\kappa\omicron\omicron\varsigma\tau\eta\varsigma, \wedge \{d_{\epsilon\lambda\lambda}, d_{\kappa\upsilon\pi}\})$$

Το  $F_1$  μπορεί να χρησιμοποιηθεί για τον καθορισμό του τύπου των οντοτήτων, το  $F_2$  για να περιορίσει τα αποτελέσματα σε οντότητες που έχουν συνεργάτες, όπου το “any” μπορεί να χρησιμοποιηθεί για να εκφράσει ότι δεν γίνεται αναζήτηση για κάποιον συγκεκριμένο συνεργάτη, το  $F_3$  για την επιλογή της πόλης εργασίας του συνεργάτη και τα  $F_4$  και  $F_5$  για την επιλογή της υπηκοότητας του υπαλλήλου.

Ένα faceted interface αναπαριστά μία αλληλουχία από facets που προβάλλεται στους χρήστες, η οποία μπορεί να περιλάβει τις επιλογές τους πάνω στις σχετικές τιμές. Με αυτόν τον τρόπο μπορεί να κωδικοποιηθεί τόσο το ερώτημα, μέσω της απάντησης του οποίου προκύπτουν τα αποτελέσματα της αναζήτησης, όσο και οι διαθέσιμες προς το χρήστη επιλογές που του δίνουν τη δυνατότητα να περιορίσει τα αποτελέσματα.

**Ορισμός 2:** Ένα basic faceted interface (BFI) είναι το ζεύγος  $(F, \Sigma)$ , όπου  $F$  ένα facet και  $\Sigma \subseteq F|_2$  το σύνολο των επιλεγμένων τιμών. Το σύνολο των faceted interfaces δίνεται από την ακόλουθη γραμματική, όπου  $I_0$  και  $I_1 = (F, \Sigma)$  είναι BFIs και  $F|_1 \in BP$ :

$$I ::= path \mid (path \wedge path) \mid (path \vee path),$$

$$path ::= I_0 \mid (I_1 / I)$$

Ένα BFI κωδικοποιεί τις επιλογές του χρήστη για κάποιο συγκεκριμένο facet. Για παράδειγμα το BFI  $(F_1, \{\Upsilon\acute{\alpha}\lambda\lambda\eta\lambda\omicron\varsigma\})$  επιλέγει τα αντικείμενα που έχουν κατηγοριοποιηθεί ως υπάλληλοι. Τα BFIs τοποθετούνται μαζί σε μονοπάτια (paths), δηλαδή ακολουθίες από εμφωλευμένα (nested) facets που επιτρέπουν τη πλοήγηση στο σύνολο των αντικειμένων. Μέσω του nesting (‘/’) μπορούν να εκτελεσθούν ερωτήματα όπως το ακόλουθο:

“υπάλληλοι που συνεργάζονται με κάποιον αντιπρόσωπο που έχει έδρα στην Πάτρα”

Το παραπάνω ερώτημα μπορεί να σχεδιαστεί με τη χρήση του interface  $(F_2, \{any\}) / (F_3, \{d_{\pi}\})$ , όπου αρχικά επιλέγονται οι υπάλληλοι που έχουν κάποιον (any) αντιπρόσωπο ως συνεργάτη, και σε δεύτερη φάση εφαρμόζεται ο περιορισμός ο αντιπρόσωπος να εργάζεται στην Πάτρα. Δύο τύποι διακλάδωσης μπορούν να εφαρμοστούν: ο  $(path_1 \wedge path_2)$ , ο οποίος υποδεικνύει ότι τα αποτελέσματα της αναζήτησης πρέπει να ικανοποιούν τις συνθήκες στο  $path_1$  και το  $path_2$  ταυτόχρονα,

και ο  $(path_1 \vee path_2)$  όπου πρέπει να ικανοποιούνται οι συνθήκες στο  $path_1$  ή το  $path_2$ . Το ακόλουθο interface  $I_{Example}$  κωδικοποιεί το ερώτημα “οι υπάλληλοι της εταιρίας που είναι έλληνες ή κύπριοι υπήκοοι και συνεργάζονται με κάποιον αντιπρόσωπο που έχει έδρα στη Θεσσαλονίκη”:

$$((F_1, \{Υπάλληλος\}) \wedge (F_4, \{d_{Eλλ}, d_{Κυπ}\})) \wedge ((F_2, \{any\}) / (F_3, \{d_{θε}\}))$$

Τα ερωτήματα που κωδικοποιούνται από τα faceted interfaces μπορούν να αναπαρασταθούν μέσω τύπων στο θετικό υπαρξιακό (positive existential) τμήμα της λογικής πρώτης τάξης με μία ελεύθερη μεταβλητή. Για κάθε διάζευξη  $\varphi_1 \vee \varphi_2$ , οι φόρμουλες  $\varphi_1$  και  $\varphi_2$  έχουν το πολύ μία κοινή μεταβλητή. Επιπλέον, αυτά τα ερωτήματα περιλαμβάνουν κατηγορήματα με το πολύ δύο ορίσματα και έχουν σχήμα δέντρου. Η μεταβλητή εξόδου του faceted ερωτήματος αποτελεί τη μεταβλητή – ρίζα αυτού του δέντρου. Το ερώτημα  $I_{Example}$  του παραπάνω παραδείγματος τελικά επιστρέφει δύο υπαλλήλους: Τον Αντρέα και το Βασίλη.

# 4

## *Περιγραφή Μεθοδολογίας και Οντολογιών*

Στην παρούσα διπλωματική εργασία, για τις ανάγκες της εφαρμογής που υλοποιήθηκε αναπτύχθηκε μία οντολογία που αφορά χαρακτηριστικά δανειακών λογαριασμών. Τα δανειακά στοιχεία που χρησιμοποιήθηκαν είναι πραγματικά και παρασχέθηκαν από μία ελληνική εμπορική τράπεζα. Σε κάθε περίπτωση τα στοιχεία είναι αποπροσωποποιημένα, έτσι ώστε να μην υπάρχει κίνδυνος εντοπισμού προσωπικών δεδομένων οποιουδήποτε δανειολήπτη. Ακολουθεί μία περιγραφή του κινήτρου για την ανάπτυξη μιας τέτοιου είδους εφαρμογής.

### *4.1 Κίνητρο Ανάπτυξης*

Ένα από τα σημαντικότερα προβλήματα που παρουσιάζονται στον τραπεζικό τομέα είναι η διαχείριση της πληροφορίας. Ένας τεράστιος όγκος δεδομένων βρίσκεται αποθηκευμένος στα data warehouses των τραπεζών και διαρκώς παρουσιάζεται η ανάγκη άντλησής του για πολλές διαφορετικές χρήσεις. Ειδικότερα τα τελευταία χρόνια λόγω της παγκόσμιας τραπεζικής κρίσης αυτή η ανάγκη γίνεται ολοένα και πιο αυξημένη. Μία πληθώρα εμπορικών εφαρμογών χρησιμοποιείται από τις τράπεζες για σκοπούς εξόρυξης δεδομένων και υπολογισμού μεγεθών που είναι απαραίτητα για τη λειτουργία τους και τις διάφορες αναφορές που θέλουν ή υποχρεούνται να παράγουν από τους εποπτικούς μηχανισμούς. Οι περισσότερες από τις τραπεζικές αυτές εφαρμογές είναι επικεντρωμένες

κυρίως σε πραγματοποίηση μετρήσεων και ανάπτυξη στατιστικών υποδειγμάτων με σκοπό την προσπάθεια υπολογισμού και εκτίμησης της εξέλιξης διαφόρων οικονομικών μεγεθών.

Ένα ακόμα σημαντικό στοιχείο που απασχολεί τις τράπεζες είναι η δυνατότητα διαρκούς και γρήγορης διοικητικής πληροφόρησης. Καθημερινά παρουσιάζεται η ανάγκη παρακολούθησης μιας πληθώρας δεικτών και στατιστικών στοιχείων που είναι καθοριστικά για την ομαλή λειτουργία τους και το σχεδιασμό της στρατηγικής που θα ακολουθήσουν. Φυσικά τα εμπορικά εργαλεία που αναφέρθηκαν παραπάνω παρέχουν πολλές τέτοιες δυνατότητες.

Μία επιπλέον ανάγκη που παρουσιάζεται συχνά στα τραπεζικά στελέχη είναι αυτή της έξυπνης πλοήγησης στα διάφορα στοιχεία του χαρτοφυλακίου, εν προκειμένω στους δανειακούς λογαριασμούς. Μέσα από το φιλτράρισμα διαφορετικών περιπτώσεων ο τελικός χρήστης μπορεί να εξάγει πολλά χρήσιμα συμπεράσματα για τα χαρακτηριστικά του χαρτοφυλακίου, εφ' όσον του παρέχεται η δυνατότητα να απομονώσει τις ξεχωριστές περιπτώσεις ανάλογα με τις ανάγκες και τις προτιμήσεις του. Με αυτόν τον τρόπο μπορεί να εντοπίσει πρότυπα και συμπεριφορές στις διάφορες κατηγορίες δανειακών λογαριασμών και πελατών. Για να επιτευχθούν τα παραπάνω είναι απαραίτητη η δόμηση και η οργάνωση της διαθέσιμης πληροφορίας, η οποία σχεδόν πάντοτε βρίσκεται αποθηκευμένη σε συστήματα διαχείρισης βάσεων δεδομένων. Αυτά τα συστήματα χαρακτηρίζονται από μία εσωτερική δομή η οποία δεν είναι πάντοτε ταιριαστή στις ανάγκες του χρήστη για τη σχετική παροχή πληροφόρησης.

Σε αυτό το σημείο έρχεται να λειτουργήσει συμπληρωματικά η επιστήμη αναπαράστασης της γνώσης. Μέσω της ανάπτυξης μίας οντολογίας παρέχεται η δυνατότητα μιας διαφορετικής, πιο γενικευμένης και δυναμικής οργάνωσης και αναπαράστασης της πληροφορίας. Η οντολογία αυτή μετατρέπεται σε μία διαπροσωπία (interface) που δίνει τη δυνατότητα εξερεύνησης των δανειακών στοιχείων με χρήση των εννοιών και των σχέσεών της, κάτι που φυσικά ο χρήστης δεν είναι απαραίτητο να γνωρίζει. Αυτό που επιτυγχάνεται τελικά είναι η διασύνδεση της οντολογίας με τις βάσεις δεδομένων και η τελική ενοποίηση των δεδομένων.

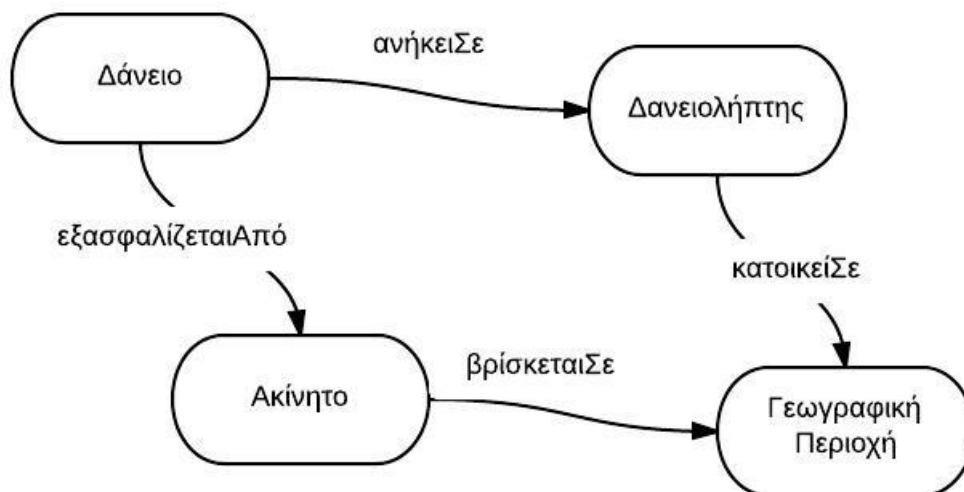
## ***4.2 Περιγραφή Χαρακτηριστικών – Έννοιες και Σχέσεις***

Τα διαθέσιμα στοιχεία αφορούν τα λεγόμενα δάνεια λιανικής τραπεζικής. Συγκεκριμένα, πρόκειται για λογαριασμούς στεγαστικών δανείων, καταναλωτικών δανείων και πιστωτικών καρτών. Τα δάνεια αυτά διαθέτουν μια πληθώρα χαρακτηριστικών, όπως



είναι το τρέχον υπόλοιπο, το αρχικό ποσό, οι δείκτες καθυστέρησης που εμφανίζουν, το επιτόκιο, ο σκοπός χορήγησής τους και πολλά άλλα. Επιπλέον, οι δανειακοί λογαριασμοί σχετίζονται με πελάτες οι οποίοι μπορούν να είναι φυσικά ή νομικά πρόσωπα και επίσης χαρακτηρίζονται από διάφορα στοιχεία όπως η περιοχή κατοικίας και η επαγγελματική δραστηριότητα.

Σύμφωνα με τα παραπάνω σχηματίστηκε μία οντολογία, βάσει των περιορισμών της Περιγραφικής Λογικής DL-Lite<sub>R</sub>. Ο λόγος που χρησιμοποιήθηκε μία γλώσσα από αυτή την DL οικογένεια, όπως έχει ήδη περιγραφεί αναλυτικά στο κεφάλαιο 2, είναι η δυνατότητα μετασχηματισμού των DL ερωτημάτων σε ερωτήματα που μπορούν να εκτελεστούν άμεσα πάνω σε μία βάση δεδομένων, εκμεταλλευόμενα έτσι τις δυνατότητες του εκάστοτε διαθέσιμου RDBMS. Η οντολογία περιλαμβάνει ένα σύνολο βασικών εννοιών, όπως “Δάνειο” και “Δανειολήπτης” οι οποίες διαθέτουν διάφορες υποέννοιες, σχηματίζοντας κάποιες ιεραρχίες. Επιπλέον, οι βασικές έννοιες συνδέονται μεταξύ τους με ατομικούς ρόλους, όπως για παράδειγμα ο “ανήκειΣε” ο οποίος συνδέει το δάνειο με τον δανειολήπτη. Ακολουθεί ένα διάγραμμα που απεικονίζει 4 βασικές έννοιες και τις σχέσεις που τις συνδέουν.



#### 4.2.1 Η Έννοια του Δανείου

Η κεντρική έννοια της οντολογίας που θα παρουσιαστεί στη συνέχεια, πάνω στην οποία παρέχει αποτελέσματα η εφαρμογή πλοήγησης που υλοποιήθηκε, είναι η έννοια του δανείου ή δανειακού λογαριασμού. Το δάνειο αποτελεί το πρώτο και κύριο facet της

εφαρμογής και οι διάφορες κατηγοριοποιήσεις του θα εμφανίζονται ως sub – facets, κάτι που ισχύει αντίστοιχα και για τις υπόλοιπες βασικές έννοιες. Η ύπαρξη των sub – facets οφείλεται στην ανάγκη ομαδοποίησης των διάφορων υποεννοιών, δεδομένου ότι απεικονίζουν διαφορετικές εννοιολογικές κατηγορίες της έννοιας στην οποία υπάγονται. Για παράδειγμα, ένα δάνειο λιανικής μπορεί να είναι δάνειο προκαθορισμένης διάρκειας και συγκεκριμένου αρχικού ποσού ή δάνειο ανακυκλούμενης πίστωσης που διαθέτει κάποιο μέγιστο όριο χρήσης. Το δάνειο προκαθορισμένης διάρκειας μπορεί να αποτελεί στεγαστικό ή καταναλωτικό δάνειο, ενώ το δάνειο ανακυκλούμενης πίστωσης μπορεί να είναι πιστωτική κάρτα ή λογαριασμός ανοιχτού ορίου. Σε όρους Περιγραφικών Λογικών αυτή η ιεραρχία μπορεί να αναπαρασταθεί με τα ακόλουθα αξιώματα:

$$\text{ΔάνειοΠροκαθορισμένουΔοσολογίου} \sqsubseteq \text{Δάνειο}$$

$$\text{ΔάνειοΑνακυκλούμενηςΠίστωσης} \sqsubseteq \text{Δάνειο}$$

$$\text{ΣτεγαστικόΔάνειο} \sqsubseteq \text{ΔάνειοΠροκαθορισμένουΔοσολογίου}$$

$$\text{ΚαταναλωτικόΔάνειο} \sqsubseteq \text{ΔάνειοΠροκαθορισμένουΔοσολογίου}$$

$$\text{ΠιστωτικήΚάρτα} \sqsubseteq \text{ΔάνειοΑνακυκλούμενηςΠίστωσης}$$

$$\text{ΑνοιχτόΌριο} \sqsubseteq \text{ΔάνειοΑνακυκλούμενηςΠίστωσης}$$

Αντίστοιχα, ένα δάνειο κατηγοριοποιείται και βάσει άλλων χαρακτηριστικών, όπως σε δάνειο μικρού, μέσου και μεγάλου πλήθους καθυστερημένων πληρωμών, ανάλογα με το ύψος του υπολοίπου που εμφανίζει, το είδος του επιτοκίου του, το σκοπό για τον οποίο χορηγήθηκε και την παλαιότητα της έναρξής του.

#### **4.2.2 Η Έννοια του Δανειολήπτη**

Το επόμενο facet περιγράφει τις ιδιότητες του δανειολήπτη στον οποίο ανήκει το δάνειο. Η σχέση μεταξύ δανείου και δανειολήπτη μπορεί να συμβολισθεί ως:

$$\text{ανήκειΣε}(\text{Δάνειο}, \text{Δανειολήπτης})$$

Αντίστοιχα με την έννοια του δανείου, ορίζονται διάφορες υποέννοιες που υπάγονται στο δανειολήπτη ανάλογα με το είδος απασχόλησής του (π.χ. υπάλληλος, ελεύθερος επαγγελματίας, συνταξιούχος κλπ) και το επάγγελμά του, όπως για παράδειγμα:

$$\text{Μηχανικός} \sqsubseteq \text{Δανειολήπτης}$$

$$\text{ΗλεκτρολόγοςΜηχανικός} \sqsubseteq \text{Μηχανικός}$$

*ΜηχανολόγοςΜηχανικός*  $\sqsubseteq$  *Μηχανικός*

*ΠολιτικόςΜηχανικός*  $\sqsubseteq$  *Μηχανικός*

Οι σχέσεις που εφαρμόζονται σε μία έννοια προφανώς επεκτείνονται και στις υποέννοιές της. Αν για παράδειγμα υπάρχει μία πιστωτική κάρτα η οποία ανήκει σε έναν ηλεκτρολόγο μηχανικό, δεδομένου ότι ισχύει *ανήκειΣε(Δάνειο, Δανειολήπτης)* προκύπτει και ότι *ανήκειΣε(ΠιστωτικήΚάρτα, ΗλεκτρολόγοςΜηχανικός)*.

### **4.2.3 Η έννοια του Ακινήτου**

Πολλοί από τους δανειακούς λογαριασμούς των στοιχείων της βάσης δεδομένων που χρησιμοποιήθηκε εξασφαλίζονται από ένα ή περισσότερα ακίνητα. Αυτό συμβαίνει κυρίως, αλλά όχι αποκλειστικά, για τα στεγαστικά δάνεια. Μέσω του τρίτου facet της εφαρμογής απεικονίζεται η σχέση εξασφάλισης, όπου αυτή υπάρχει, μεταξύ δανείου και ακινήτου:

*εξασφαλίζεταιΑπό(Δάνειο, Ακίνητο)*

Όπως και στις προηγούμενες περιπτώσεις, διάφορες υποέννοιες υπάγονται στο ακίνητο, ανάλογα με τη χρήση που αυτό εξυπηρετεί και το εύρος της τρέχουσας αξίας του. Για παράδειγμα, ένα ακίνητο μπορεί να αποτελεί κατοικία, επαγγελματικό χώρο ή ακόμα να χρησιμοποιείται ταυτόχρονα και για τους δύο σκοπούς, όπως προκύπτει από τα ακόλουθα αξιώματα:

*Κατοικία*  $\sqsubseteq$  *Ακίνητο*

*ΕπαγγελματικόΑκίνητο*  $\sqsubseteq$  *Ακίνητο*

*ΑκίνητοΜικτήςΧρήσης*  $\sqsubseteq$  *Ακίνητο*

### **4.2.4 Η έννοια της Γεωγραφικής Περιοχής**

Στις βάσεις δεδομένων των τραπεζών τα γεωγραφικά στοιχεία αποτελούν σημαντική πληροφορία που αποθηκεύεται σε διάφορα επίπεδα, έτσι ώστε να παρέχει τη δυνατότητα γνώσης σχετικά με τις περιοχές κατοικίας των πελατών της, τις έδρες που βρίσκονται οι εταιρίες με τις οποίες συνεργάζεται, τη γεωγραφική κατανομή των δανειακών και καταθετικών υπολοίπων βάσει της θέσης των καταστημάτων στα οποία αυτά υπάγονται και πολλά άλλα.

Στο επόμενο facet της εφαρμογής αποτυπώνεται η σχέση διαμονής του δανειολήπτη με κάποια γεωγραφική περιοχή:

*κατοικείΣε(Δανειολήπτης, ΓεωγραφικήΠεριοχή)*

Οι γεωγραφικές περιοχές χωρίζονται ανά την Ελλάδα με διάφορους τρόπους, ανάλογα με το φορέα που πραγματοποιεί το διαχωρισμό και τις παραμέτρους των εκάστοτε προβλημάτων και αναγκών που τον επιβάλουν. Για τις ανάγκες της παρούσας εφαρμογής, στην οντολογία που θα περιγραφεί στη συνέχεια ο διαχωρισμός έγινε βάσει διοικητικών περιφερειών, νομών και δήμων. Έτσι δημιουργείται μία ιεραρχία αξιωμάτων υπαγωγής, όπως η ακόλουθη:

*Ελλάδα  $\sqsubseteq$  ΓεωγραφικήΠεριοχή*

*Κρήτη  $\sqsubseteq$  Ελλάδα*

*Πελοπόννησος  $\sqsubseteq$  Ελλάδα*

*Ήπειρος  $\sqsubseteq$  Ελλάδα*

Η κάθε περιφέρεια χωρίζεται σε νομούς, όπως:

*ΝομόςΧανίων  $\sqsubseteq$  Κρήτη*

*ΝομόςΡεθύμνου  $\sqsubseteq$  Κρήτη*

*ΝομόςΗρακλείου  $\sqsubseteq$  Κρήτη*

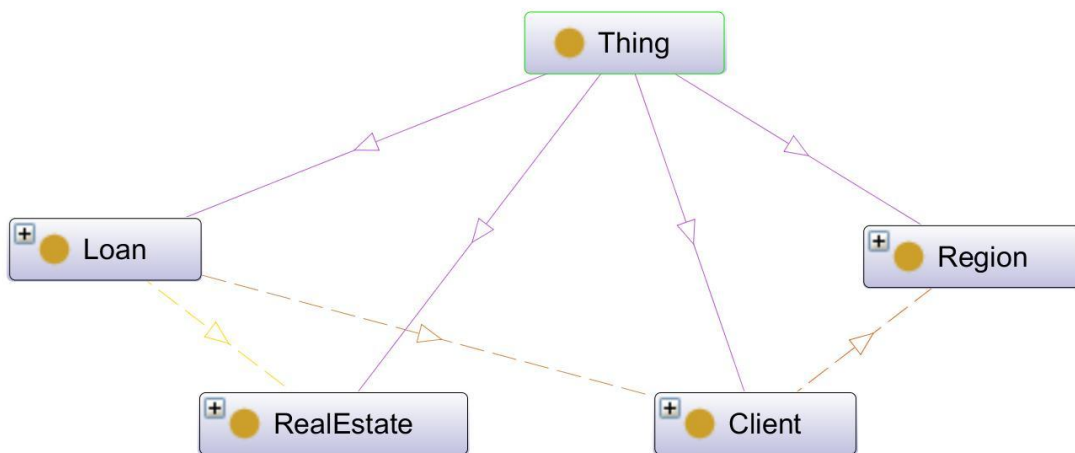
*ΝομόςΛασιθίου  $\sqsubseteq$  Κρήτη*

### **4.3 Ανάπτυξη Οντολογίας**

Η ανάπτυξη της οντολογίας πραγματοποιήθηκε με χρήση της εφαρμογής *protégé*. Το *protégé* είναι μία ελεύθερη, ανοικτού κώδικα (free open source) εφαρμογή σύνταξης οντολογιών, υλοποιημένη στη γλώσσα προγραμματισμού Java. Παράλληλα, προσφέρει τη δυνατότητα διασύνδεσης με άλλες εφαρμογές, οι οποίες λειτουργούν μέσω του *protégé* με τη μορφή plugins και παρέχουν επιπλέον δυνατότητες, όπως μηχανισμούς συλλογιστικής (reasoning) και εξαγωγής συμπερασμάτων και νέων εννοιών και γραφικής αναπαράστασης των στοιχείων της συντασσόμενης οντολογίας. Το *protégé* αποτελεί ένα ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment – IDE)

οντολογιών, ίσως το δημοφιλέστερο αντίστοιχο εργαλείο τη χρονική στιγμή που γράφεται αυτό το κείμενο, παρέχοντας ένα εύχρηστο γραφικό περιβάλλον που επιτρέπει στο χρήστη να κατασκευάζει και να συντηρεί πολύπλοκες οντολογίες γρήγορα και ευέλικτα, χωρίς να είναι αναγκασμένος να εμπλακεί άμεσα με τη σύνταξη και τις ιδιαιτερότητες της γλώσσα OWL.

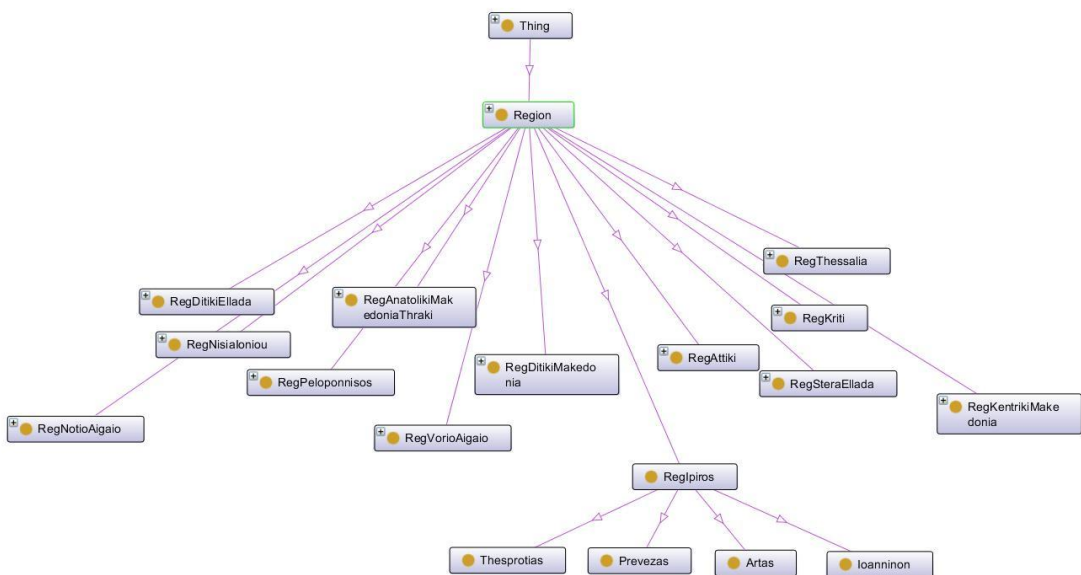
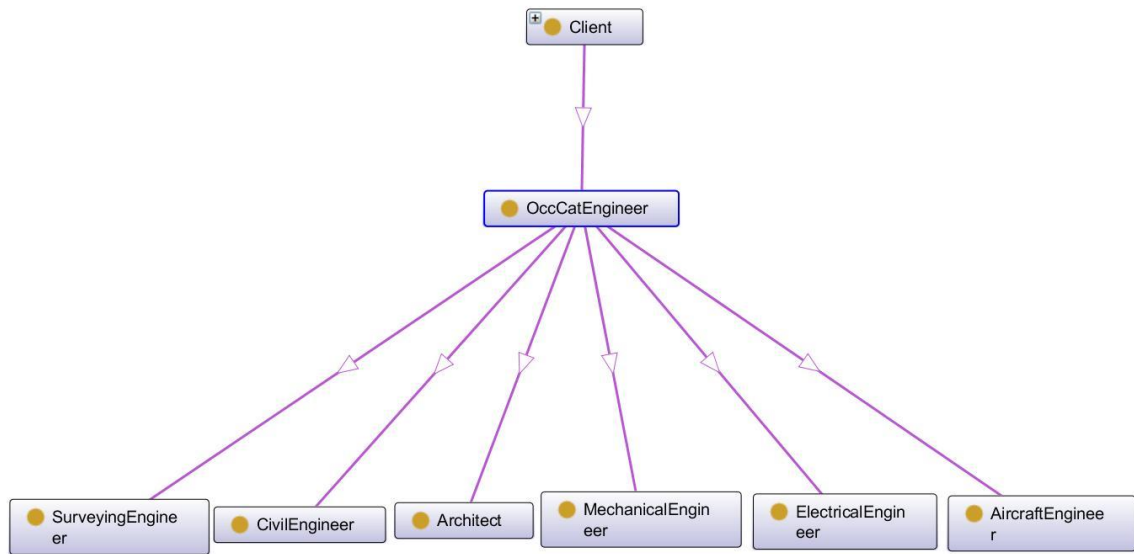
Οι τέσσερις βασικές έννοιες και οι σχέσεις που τις διέπουν ορίστηκαν βάσει της γενικής περιγραφής που δόθηκε στην προηγούμενη παράγραφο. Όπως σε κάθε OWL οντολογία, όλες οι έννοιες υπάγονται σε μία γενική έννοια με το όνομα “*Thing*”. Στο ακόλουθο σχήμα απεικονίζεται η παραπάνω ιεραρχία. Αυτός ο γράφος, καθώς και τα ακόλουθα σχήματα της παραγράφου, δημιουργήθηκε με τα εργαλεία OWLViz και OntoGraf του *protégé*. Με τις συνεχείς γραμμές συμβολίζονται τα αξιώματα υπαγωγής (is – a) και με τις διακεκομμένες τα αξιώματα σχέσεων μεταξύ των εννοιών:

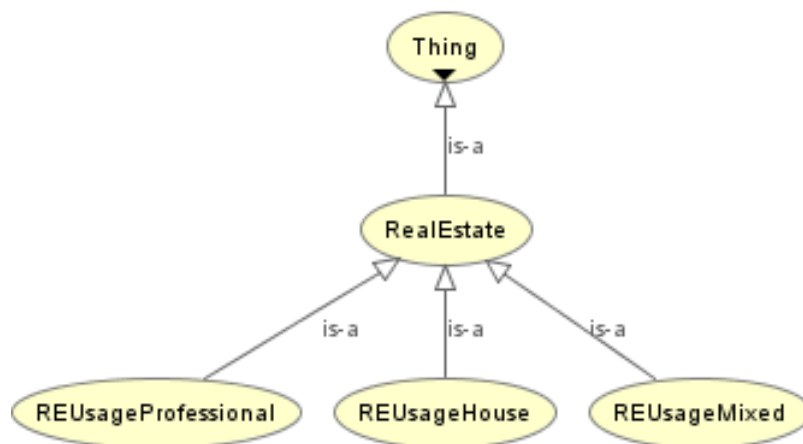
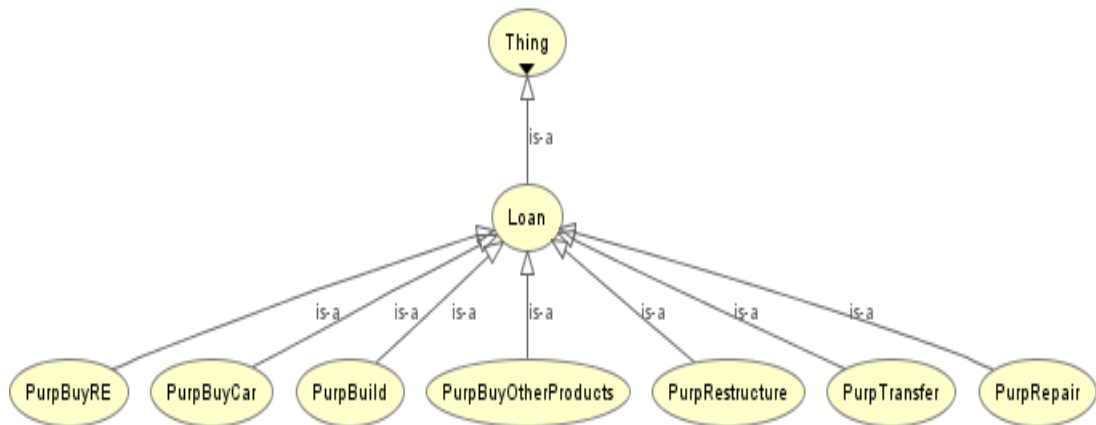
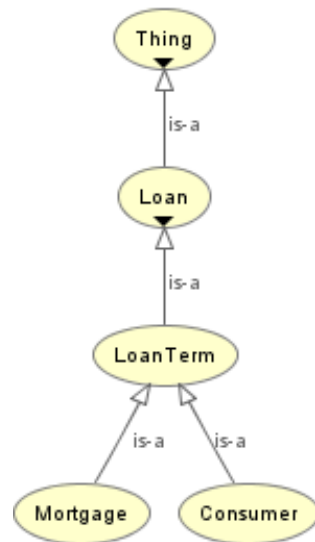
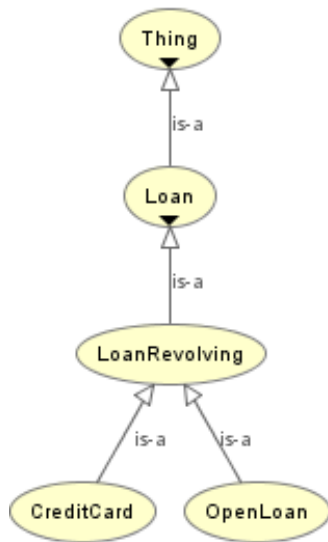


Μέσω των παραπάνω ορίζονται τα κεντρικά facets της εφαρμογής. Τα ονόματα των σχέσεων στην οντολογία είναι τα ακόλουθα:

- ownedBy(Loan, Client)
- residesIn(Client, Region)
- hasCollateral(Loan, RealEstate)

Οι ιεραρχίες των εννοιών επεκτείνονται και σε επιπλέον επίπεδα, όπως φαίνεται στα ακόλουθα παραδείγματα:





Το τελικό παραγόμενο αποτέλεσμα που προκύπτει από την κατασκευή της οντολογίας στο protégé είναι ένα αρχείο κειμένου με κατάληξη “.owl”, το οποίο περιέχει την περιγραφή της οντολογίας στη συγκεκριμένη γλώσσα αναπαράστασης γνώσης.

## 4.4 Συζευκτικά Ερωτήματα

Στις εφαρμογές που πραγματοποιούν συνεχή και εντατική διαχείριση δεδομένων, τα ερωτήματα στις βάσεις γνώσεις διαδραματίζουν κεντρικό ρόλο. Σε ορισμένες περιπτώσεις η ανάκτηση στιγμιοτύπων (instance retrieval) αποτελεί μία μάλλον αδύναμη μορφή ερωτημάτων. Ως ερωτήματα χρησιμοποιούνται έννοιες, οι οποίες μπορούν να εκφράσουν μόνο δενδρικές δομές. Επιπρόσθετα, δεν είναι πάντοτε άμεσα δυνατή η ανάκτηση πλειάδων από στιγμιότυπα που να πληρούν κάποιες συγκεκριμένες συνθήκες. Τα συζευκτικά ερωτήματα (Conjunctive Queries – CQs) αποτελούν μία εκφραστική γλώσσα ερωτημάτων που παρέχει επιπλέον σημαντικές δυνατότητες σε σχέση με την κλασική ανάκτηση στιγμιοτύπων.

Συζευκτικό ερώτημα είναι μία σύζευξη ατομικών εννοιών της μορφής  $C(x)$  και ατομικών ρόλων της μορφής  $R(x, y)$ , όπου  $C$  είναι μία έννοια,  $R$  ένας ρόλος και τα  $x, y$  μπορούν να είναι μεταβλητές ή ονόματα ατόμων. Με βάση την οντολογία που περιγράφηκε παραπάνω, ένα συζευκτικό ερώτημα που θα επιστρέψει ζεύγη δανείων – δανειοληπτών είναι το ακόλουθο:

$$q(x, y) \leftarrow Loan(x) \wedge ownedBy(x, y) \wedge Client(y)$$

Στην αριστερή πλευρά του βέλους τα  $(x, y)$  ορίζουν ποιες μεταβλητές θα επιστρέψουν τιμές με την απάντηση του ερωτήματος. Στις περιπτώσεις όπως αυτή του παραδείγματος, όπου όλες οι μεταβλητές που εμφανίζονται στο ερώτημα επιστρέφουν στην απάντηση, τα αποτελέσματα του ερωτήματος θα αποτελούνται από ζεύγη ονομάτων ατόμων (αριθμός δανείου, κωδικός δανειολήπτη) από τη βάση γνώσης, τέτοια ώστε μετά την αντικατάσταση των μεταβλητών από τα αντίστοιχα ονόματα του κάθε ζεύγους που θα προκύψει από την απάντηση του ερωτήματος, όλα τα ορίσματα της σύζευξης (conjunctions) να είναι αληθή για κάθε μοντέλο της βάσης γνώσης. Με αυτόν τον τρόπο δομούνται τα συζευκτικά ερωτήματα στην εφαρμογή που αναπτύχθηκε, αφού τα αποτελέσματα τελικά θα προκύψουν από μία ένωση SQL ερωτημάτων στη σχεσιακή βάση δεδομένων, όπου οι πιθανές τιμές που μπορούν να λάβουν όλες οι ελεύθερες μεταβλητές είναι απαραίτητες για την εξαγωγή επιπλέον στοιχείων που σχετίζονται με κάθε κατηγορία και θα παρουσιαστούν στο χρήστη.

Η απάντηση ερωτημάτων σε μία βάση γνώσης διαφέρει από την αντίστοιχη διαδικασία σε μία κλασική βάση δεδομένων. Στη δεύτερη περίπτωση ισχύει η υπόθεση κλειστού κόσμου (Closed World Assumption – CWA), κάτι που σημαίνει πως οτιδήποτε δεν δηλώνεται ρητά θεωρείται ως ψευδές. Αντιθέτως, στις Περιγραφικές Λογικές, όπως και γενικότερα στη Λογική Πρώτης Τάξης, ισχύει συνήθως η υπόθεση ανοιχτού κόσμου



(Open World Assumption – OWA). Μία ακόμα διαφορά είναι ότι στις βάσεις δεδομένων ισχύει η υπόθεση μοναδικού ονόματος (Unique Name Assumption – UNA), ενώ στα περισσότερα συστήματα Περιγραφικών Λογικών διαφορετικά ονόματα ατόμων μπορούν να αντιστοιχούν στο ίδιο στοιχείο ενός μοντέλου.

Στη συνέχεια παρουσιάζονται μερικά παραδείγματα συζευκτικών ερωτημάτων με βάση την οντολογία που αναπτύχθηκε. Αρχικά εμφανίζονται απλά ερωτήματα που αντιστοιχούν στην επιλογή ενός μόνο φίλτρου της εφαρμογής, τα οποία γίνονται σταδιακά όλο και πιο σύνθετα μέσω της επιλογής πολλαπλών περιορισμών στα διάφορα facets, με σκοπό να περιοριστούν και να συγκεκριμενοποιηθούν τα αποτελέσματα της αναζήτησης.

### **Παράδειγμα 1:**

Ένα πρώτο απλό ερώτημα είναι η εύρεση των δανειακών λογαριασμών που αποτελούν δάνεια προκαθορισμένης διάρκειας:

$$q(x) \leftarrow \text{LoanTerm}(x)$$

### **Παράδειγμα 2:**

Οι λογαριασμοί που αποτελούν στεγαστικά δάνεια και χορηγήθηκαν με σκοπό την επισκευή μίας κατοικίας:

$$q(x) \leftarrow \text{Mortgage}(x) \wedge \text{PurpRepair}(x)$$

### **Παράδειγμα 3:**

Οι λογαριασμοί που αποτελούν ανακυκλούμενες πιστώσεις και ο κάτοχός τους είναι μηχανικός:

$$q(x, y) \leftarrow \text{LoanRevolving}(x) \wedge \text{ownedBy}(x, y) \wedge \text{OccCatEngineer}(y)$$

### **Παράδειγμα 4:**

Οι πιστωτικές κάρτες των οποίων οι κάτοχοι είναι γιατροί που κατοικούν στην Κρήτη:

$$q(x, y, z) \leftarrow \text{CreditCard}(x) \wedge \text{ownedBy}(x, y) \wedge \text{Doctor}(y) \wedge \text{residesIn}(y, z) \\ \wedge \text{RegKriti}(z)$$

### **Παράδειγμα 5:**

Τα καταναλωτικά δάνεια που εμφανίζουν μεγάλη καθυστέρηση, οι κάτοχοί τους έχουν υψηλό μορφωτικό επίπεδο και κατοικούν στην Ήπειρο:

$$q(x, y, z) \leftarrow \text{Consumer}(x) \wedge \text{DelHigh}(x) \wedge \text{ownedBy}(x, y) \wedge \text{EduLevelHigh}(y) \\ \wedge \text{residesIn}(y, z) \wedge \text{RegIpiros}(z)$$

#### **Παράδειγμα 6:**

Τα δάνεια που ανήκουν σε φυσικά πρόσωπα και εξασφαλίζονται από επαγγελματικό ακίνητο:

$$q(x, y, z) \leftarrow \text{Loan}(x) \wedge \text{ownedBy}(x, y) \wedge \text{TypeIndividual}(y) \wedge \text{hasCollateral}(x, z) \\ \wedge \text{REUsageProfessional}(z)$$

#### **Παράδειγμα 7:**

Τα δάνεια προκαθορισμένης διάρκειας και κυμαινόμενου επιτοκίου που χορηγήθηκαν με σκοπό την αγορά αυτοκινήτου, ο κάτοχός τους είναι ελεύθερος επαγγελματίας μέσης εκπαίδευσης και κατοικεί στην Αττική:

$$q(x, y, z) \leftarrow \text{TermLoan}(x) \wedge \text{RateVariable}(x) \wedge \text{PurpBuyCar}(x) \wedge \text{ownedBy}(x, y) \\ \wedge \text{OccTypeFreelancer}(y) \wedge \text{EduLevelMid}(y) \wedge \text{residesIn}(y, z) \\ \wedge \text{RegAttiki}(z)$$

#### **Παράδειγμα 8:**

Τα στεγαστικά δάνεια σταθερού επιτοκίου χωρίς καθυστέρηση που εξασφαλίζονται από ακίνητο μικτής χρήσης, ο κάτοχός τους είναι ηλεκτρολόγος μηχανικός, εργάζεται ως υπάλληλος και κατοικεί στην Αττική:

$$q(x, y, w, z) \leftarrow \text{Mortgage}(x) \wedge \text{RateFixed}(x) \wedge \text{DelZero}(x) \wedge \text{hasColateral}(x, w) \\ \wedge \text{REUsageMixed}(w) \wedge \text{ownedBy}(x, y) \wedge \text{ElectricalEngineer}(y) \\ \wedge \text{OccTypeEmployee}(y) \wedge \text{residesIn}(y, z) \wedge \text{RegAttiki}(z)$$

## **4.5 Εξαγωγή Συμπερασμάτων**

Η οντολογία που αναπτύχθηκε στο protégé, πέρα από την περιγραφή των ατομικών εννοιών και των ατομικών ρόλων, αποτελεί ουσιαστικά μόνο το σώμα ορολογίας (TBox) της απαιτούμενης βάσης γνώσης, αφού το σώμα ισχυρισμών (ABox) θα προκύψει μέσω των στοιχείων που βρίσκονται αποθηκευμένα στη βάση δεδομένων. Όπως περιγράφεται αναλυτικότερα σε επόμενο κεφάλαιο, το συζευκτικό ερώτημα που αρχικά δημιουργείται μέσω της επιλογής των φίλτρων στα facets και της χρήσης της ιεραρχίας και των σχέσεων που περιγράφονται στην οντολογία, μετατρέπεται με τη διαδικασία της

επανεγγραφής (rewriting) σε ένα SQL ερώτημα, ακριβέστερα στην ένωση (union) πολλών SQL ερωτημάτων, που εφαρμόζονται στη βάση δεδομένων για να προκύψουν τα τελικά αποτελέσματα.

Στα πλαίσια αυτής της διαδικασίας χρησιμοποιείται ένα αρχείο απεικονίσεων. Πρόκειται για ένα αρχείο κειμένου που αντιστοιχίζει τις έννοιες και τις σχέσεις της οντολογίας με απλά SQL ερωτήματα. Στην περίπτωση των εννοιών περιλαμβάνει SQL selects με ένα επιστρεφόμενο πεδίο αντίστοιχο με το εκάστοτε επιλεγμένο φίλτρο, όπου ανάλογα με το λογικό περιεχόμενο της έννοιας εφαρμόζονται where clauses στα σχετικά πεδία της βάσης για να προκύψουν τα επιθυμητά αποτελέσματα. Όπως είναι αναμενόμενο, στη βάση δεδομένων δεν απεικονίζονται επακριβώς μέσω κάποιου συγκεκριμένου πεδίου όλες οι έννοιες που αναπαρίστανται στην βάση γνώσης. Σε αυτές τις περιπτώσεις το σύστημα επανεγγραφής, τροφοδοτούμενο από την οντολογία και χρησιμοποιώντας τα SQL ερωτήματα που αντιστοιχούν σε έννοιες που απεικονίζονται στη βάση δεδομένων, δημιουργεί σύνθετα SQL ερωτήματα που επιστρέφουν τα αποτελέσματα που πράγματι αναλογούν στις οριζόμενες έννοιες.

Για παράδειγμα, στα διαθέσιμα τραπεζικά στοιχεία των δανειοληπτών περιλαμβάνεται το ακριβές επάγγελμα του κάθε φυσικού προσώπου, αλλά όχι η γενικότερη επαγγελματική κατηγορία. Στην οντολογία ορίζονται ευρύτερες επαγγελματικές κατηγορίες, ως γενικότερες έννοιες των βασικών επαγγελμάτων, ενώ στο αρχείο απεικονίσεων υπάρχουν αντιστοιχίσεις μόνο για τα επαγγέλματα. Έτσι, εάν ο χρήστης επιθυμεί να δει στα αποτελέσματα της εφαρμογής τα δάνεια των πελατών που είναι μηχανικοί γενικά, θα δημιουργηθεί μία ένωση ερωτημάτων από την οποία θα επιστραφούν οι λογαριασμοί των δανειοληπτών με όλα τα σχετικά επαγγέλματα, όπως για παράδειγμα αυτά του ηλεκτρολόγου μηχανικού και του αρχιτέκτονα, που υπάγονται στην έννοια του μηχανικού, δεδομένου ότι στο σώμα ορολογίας της οντολογίας έχουν οριστεί τα:

*ElectricalEngineer*  $\sqsubseteq$  *OccCatEngineer*

και

*Architect*  $\sqsubseteq$  *OccCatEngineer*



# 5

## *Ανάπτυξη και Υλοποίηση*

### *5.1 Γενική Περιγραφή της Εφαρμογής*

Σκοπός της εφαρμογής που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας είναι η πλοήγηση σε στοιχεία δανειακών λογαριασμών που παρασχέθηκαν από μία ελληνική εμπορική τράπεζα. Η εφαρμογή παρέχει τη δυνατότητα στο χρήστη να εντοπίζει δανειακούς λογαριασμούς που πληρούν τις προϋποθέσεις που τον ενδιαφέρουν και να εξετάζει τα λεπτομερή χαρακτηριστικά τους. Ο περιορισμός των αποτελεσμάτων υλοποιείται μέσω της επιλογής φίλτρων από το χρήστη, τα οποία είναι οργανωμένα με τη μορφή facets που αντιστοιχούν στις διάφορες εννοιολογικές κατηγορίες που χαρακτηρίζουν τα δάνεια, προσφέροντας μία ιδιαίτερος απλή στη χρήση διεπαφή (user interface).

Η οργάνωση και η ιεραρχία των προβαλλόμενων faceted φίλτρων αντιστοιχεί πλήρως σε μία οντολογία που έχει περιγραφεί στη γλώσσα OWL, ενώ τα πρωτογενή δεδομένα βρίσκονται αποθηκευμένα σε μία κλασσική σχεσιακή βάση δεδομένων (Relational Database). Τα παραπάνω δεν απασχολούν σε καμία περίπτωση τον τελικό χρήστη, όμως ο τρόπος δόμησης της πληροφορίας μέσω της οντολογίας επιτρέπει στο μηχανικό που τη συντηρεί να προβαίνει με ευέλικτο τρόπο σε τροποποιήσεις, σύμφωνα με τις επιθυμίες και τις ανάγκες των χρηστών, χωρίς να είναι απαραίτητο να επεμβαίνει στις βασικές

δομές της εφαρμογής, δηλαδή στον κώδικα που δημιουργεί τα ερωτήματα και επιστρέφει τα αποτελέσματα.

Στη συνέχεια ακολουθεί μία συνοπτική περιγραφή των βημάτων που μεσολαβούν από τη στιγμή που ενεργοποιείται η εφαρμογή μέχρι την προβολή των τελικών επιστρεφόμενων αποτελεσμάτων:

- Με το άνοιγμα της εφαρμογής προβάλλονται τα φίλτρα με τη μορφή facets στο αριστερό μέρος της οθόνης. Το κάθε facet είναι ένα expandable menu, στο οποίο κάνοντας click εμφανίζονται οι διάφορες σχετικές κατηγορίες επιλογών με τη μορφή radio buttons
- Ο χρήστης επιλέγει τα φίλτρα που τον ενδιαφέρουν, έτσι ώστε να περιορίσει τα επιστρεφόμενα αποτελέσματα και πατά το κουμπί “*Εφαρμογή Φίλτρων*”
- Με την υποβολή των φίλτρων ενεργοποιείται ο μηχανισμός σχηματισμού του συζευκτικού ερωτήματος. Για τη διαδικασία αυτή χρησιμοποιείται το αρχείο OWL στο οποίο περιγράφεται η οντολογία. Με αυτόν τον τρόπο συντίθεται το ερώτημα μέσω των εννοιών και των σχέσεων που αντιστοιχούν στα φίλτρα και τα διάφορα facets που συμμετέχουν στις επιλογές του χρήστη
- Το συζευκτικό ερώτημα τροφοδοτείται στο σύστημα επανεγγραφής ερωτημάτων. Μία ακόμα είσοδος σε αυτό το σύστημα είναι το αρχείο απεικονίσεων, το οποίο αντιστοιχίζει έννοιες και ρόλους της οντολογίας σε απλά SQL ερωτήματα στη βάση δεδομένων. Τα where clauses αυτών των ερωτημάτων εκφράζουν σε όρους πεδίων της βάσης τις σχετικές έννοιες και ρόλους, ενώ τα πεδία που επιλέγονται (select) αντιστοιχούν στις ελεύθερες μεταβλητές των conjuncts. Από την επανεγγραφή του ερωτήματος προκύπτει ένα SQL ερώτημα. Τα πεδία που επιλέγονται σε αυτό το ερώτημα αντιστοιχούν στις ελεύθερες μεταβλητές του συζευκτικού ερωτήματος
- Το παραπάνω ερώτημα εμπλουτίζεται με πολλά επιπλέον πεδία στο select τμήμα του, έτσι ώστε να γίνουν διαθέσιμες οι διάφορες λεπτομέρειες των δανειακών λογαριασμών που θα προβληθούν τελικά στο χρήστη. Επιπλέον, τροποποιείται η μορφή του ερωτήματος που προκύπτει αρχικά από τον rewriter, έτσι ώστε να αφαιρεθούν κάποια περιττά joins και η δημιουργία ενδιάμεσων πινάκων, με αποτέλεσμα να βελτιστοποιείται το τελικό SQL ερώτημα και να ελαττώνεται σημαντικά ο χρόνος εκτέλεσής του. Ένα ακόμα πλεονέκτημα αυτής της διαδικασίας είναι η καλύτερη χρήση των δεικτών (indexes) των πινάκων μέσω της νέας μορφής του ερωτήματος, έτσι ώστε να μειώνεται ακόμα περισσότερο ο χρόνος εκτέλεσης

- Το SQL ερώτημα που προκύπτει από το παραπάνω βήμα εφαρμόζεται στη βάση δεδομένων. Τα επιστρεφόμενα αποτελέσματα αποθηκεύονται σε μία κατάλληλη δομή, κάθε αντικείμενο της οποίας αποτελείται από τα πλήρη διαθέσιμα στοιχεία ενός δανειακού λογαριασμού. Η επιστροφή των στοιχείων γίνεται τμηματικά, ανάλογα με το πλήθος των λογαριασμών που τελικά θα προβάλλονται σε κάθε σελίδα, έτσι ώστε ο χρόνος απόκρισης της εφαρμογής να είναι ο ελάχιστος δυνατός
- Η δομή με τα μερικά αποτελέσματα, το συνολικό πλήθος τους και οι μεταβλητές που αντιστοιχούν στα επιλεγμένα φίλτρα τροφοδοτούνται στο μηχανισμό προβολής της εφαρμογής. Στο κεντρικό τμήμα της οθόνης εμφανίζονται κάποια βασικά στοιχεία των δανείων με τη μορφή snippets, ενώ στο πάνω μέρος προβάλλεται το συνολικό πλήθος των αποτελεσμάτων της αναζήτησης. Στο κατώτερο μέρος της οθόνης εμφανίζεται το πλήθος των σελίδων των αποτελεσμάτων, καθώς και κουμπιά μέσω των οποίων ο χρήστης μπορεί να πλοηγηθεί στις διαφορετικές σελίδες των επιστρεφόμενων στοιχείων. Στο αριστερό τμήμα της οθόνης εμφανίζονται expanded τα facets που είχαν χρησιμοποιηθεί για την αναζήτηση, όπως και οι αρχικές επιλογές του χρήστη
- Ο χρήστης σε αυτό το σημείο έχει τη δυνατότητα να πλοηγηθεί στις σελίδες των αποτελεσμάτων, βλέποντας τα βασικά στοιχεία (snippets) των διαφόρων δανείων. Επιπλέον μπορεί να τροποποιήσει τα φίλτρα του προηγούμενου ερωτήματος και να προχωρήσει σε επανυποβολή ή να καθαρίσει όλα τα φίλτρα μέσω της σχετικής διαθέσιμης επιλογής και να προχωρήσει σε νέα αναζήτηση
- Κάνοντας click σε κάποιο συγκεκριμένο δανειακό λογαριασμό από αυτούς που συμμετέχουν στα αποτελέσματα της αναζήτησης, ανοίγει νέο παράθυρο στο οποίο προβάλλονται αναλυτικά στοιχεία για αυτόν το λογαριασμό

Από αυτά τα βήματα, τα δύο πρώτα και τα τρία τελευταία είναι εμφανή στο χρήστη, ενώ όλα τα υπόλοιπα αποτελούν ενδιάμεσα στάδια της εφαρμογής. Στις επόμενες παραγράφους αυτού του κεφαλαίου θα περιγραφούν οι τεχνολογίες που χρησιμοποιήθηκαν καθώς και οι τεχνικές που εφαρμόστηκαν για την υλοποίηση όλων των παραπάνω.

## ***5.2 Χρησιμοποιούμενες Τεχνολογίες***

Για την ανάπτυξη της εφαρμογής πλοήγησης χρησιμοποιήθηκαν συνδυαστικά διάφορες δημοφιλείς και πετυχημένες τεχνολογίες. Μία πρώτη επιλογή που έγινε ήταν αυτή της

δημιουργίας μίας δυναμικής δικτυακής (web) εφαρμογής. Τα πλεονεκτήματα της επιλογής αυτής, συγκριτικά με μία standalone εφαρμογή, είναι η εύκολη, ταυτόχρονη και απομακρυσμένη πρόσβαση από πολλούς διαφορετικούς χρήστες μέσω ενός απλού φυλλομετρητή ιστοσελίδων (web browser) χωρίς την ανάγκη κάποιας εγκατάστασης λογισμικού ή περιορισμούς λόγω του διαθέσιμου λειτουργικού συστήματος στους υπολογιστές από τους οποίους θα χρησιμοποιείται.

### **Διεπαφή**

Η διεπαφή προς τον τελικό χρήστη αποτελείται από δυναμικές ιστοσελίδες. Από τις διάφορες διαθέσιμες τεχνολογίες κατασκευής δυναμικών ιστοσελίδων και δικτυακών εφαρμογών προκρίθηκε η JSP (JavaServer Pages) για δύο βασικούς λόγους. Ο πρώτος είναι ότι εκτελείται στη μεριά του εξυπηρετητή (Server – Side Scripting) παράγοντας τελικά απλό HTML κώδικα που καταλήγει στον browser του χρήστη. Με αυτόν τον τρόπο όλες οι “βαριές” και σημαντικές διαδικασίες, όπως η εφαρμογή των ερωτημάτων στη βάση δεδομένων και η διαχείριση των αποτελεσμάτων, εκτελούνται σε ένα κεντρικό και ασφαλές σημείο, ενώ ο χρήστης έχει πρόσβαση μόνο στα τελικά αποτελέσματα. Ο δεύτερος λόγος είναι η ανάπτυξη της λειτουργικότητας της εφαρμογής στη γλώσσα προγραμματισμού Java. Η επιλογή αυτή έγινε λόγω των διάφορων διαθέσιμων APIs (Application Programming Interfaces) που έχουν υλοποιηθεί σε Java για την ανάγνωση και διαχείριση OWL οντολογιών (OWL API), καθώς και για την επανεγγραφή των συζευκτικών ερωτημάτων.

Για την προβολή των αποτελεσμάτων χρησιμοποιήθηκε η βιβλιοθήκη JSTL (JSP Standard Tag Library) η οποία αποτελεί επέκταση της τεχνολογίας JSP. Η δημιουργία των expandable menus έγινε με χρήση της client – side scripting γλώσσας προγραμματισμού JavaScript. Για τη συνολική εμφάνιση και μορφή των τελικών ιστοσελίδων χρησιμοποιήθηκε η γλώσσα CSS (Cascading Style Sheets)

### **Λειτουργικότητα**

Η λειτουργία της εφαρμογής βασίστηκε στη γλώσσα προγραμματισμού Java. Βασικές διεργασίες όπως η διαχείριση των υποβαλλόμενων φίλτρων, η κλήση των διαφόρων κλάσεων, η δημιουργία sessions που αποθηκεύονται οι απαραίτητες δομές και μεταβλητές και η προβολή των JSP σελίδων με τα αποτελέσματα υλοποιήθηκε μέσω των Java Servlets. Τα servlets είναι ειδικού σκοπού κλάσεις της γλώσσας Java που επεκτείνουν τις δυνατότητες ενός εξυπηρετητή.

### **Web Server και Web Container**



Για τη λειτουργία μίας δικτυακής εφαρμογής είναι απαραίτητος ένας web server, ο οποίος διαχειρίζεται τις HTTP αιτήσεις (HTTP requests). Το HTTP (HyperText Transfer Protocol) αποτελεί το κύριο δικτυακό πρωτόκολλο που χρησιμοποιείται για το διαμοιρασμό της πληροφορίας στον παγκόσμιο ιστό. Επιπλέον, για την εκτέλεση ενός servlet είναι απαραίτητη η χρήση ενός web container, ο οποίος διαχειρίζεται τα requests του servlet και των JSP αρχείων. Ο web container δημιουργεί στιγμιότυπα των servlet κλάσεων, φορτώνει τους servlets, δημιουργεί και διαχειρίζεται αντικείμενα αιτήσεων και απαντήσεων (request and respond objects) και εκτελεί διάφορα καθήκοντα διαχείρισης των servlets. Το λογισμικό που χρησιμοποιήθηκε και για τους δύο αυτούς ρόλους είναι ο Apache Tomcat, ο οποίος παρέχει ένα HTTP web server περιβάλλον για την εκτέλεση του Java κώδικα.

### ***Βάση Δεδομένων***

Για την αποθήκευση των διαθέσιμων δεδομένων και την εκτέλεση των SQL ερωτημάτων χρησιμοποιήθηκε το σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System – RDBMS) MySQL Community Server.

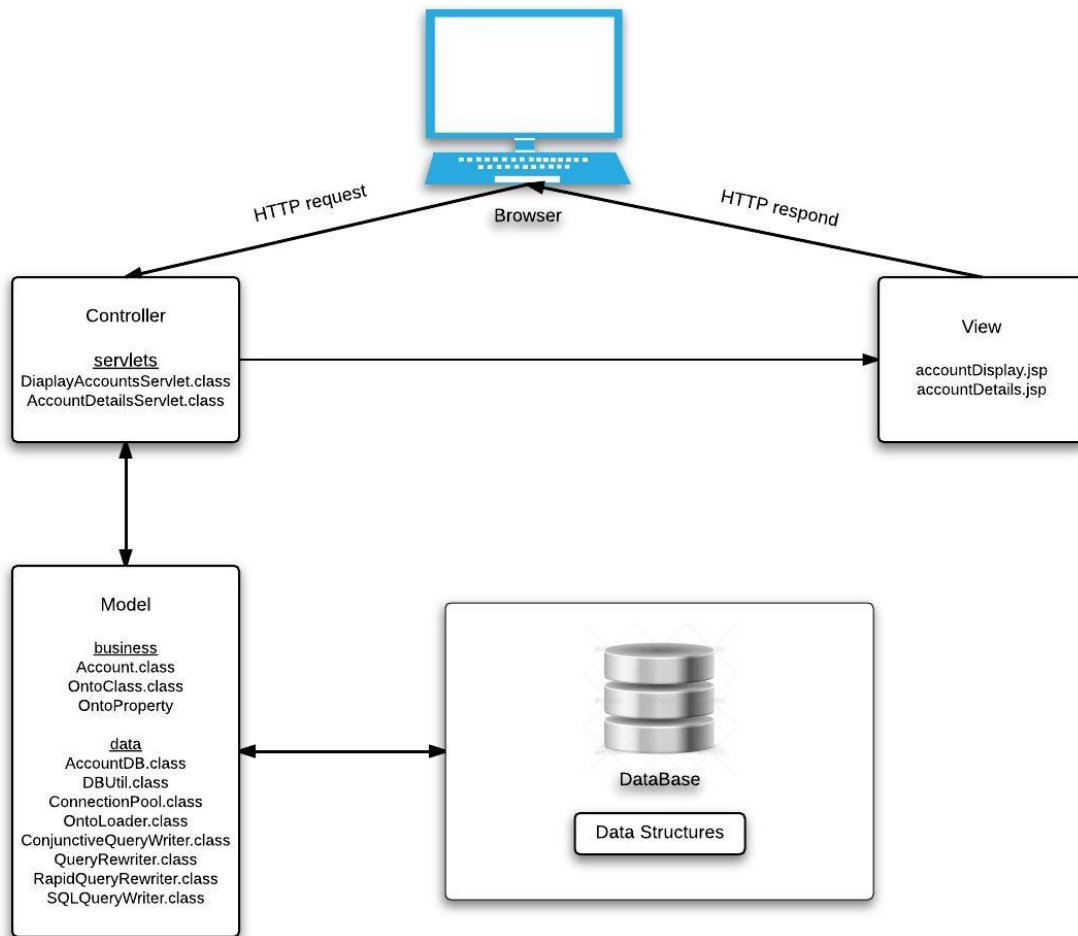
### ***Εργαλεία Ανάπτυξης***

Το περιβάλλον ανάπτυξης που χρησιμοποιήθηκε για τη δημιουργία αυτής της εφαρμογής είναι το IDE NetBeans. Πέρα από την εγγραφή, επεξεργασία και debugging του πηγαίου κώδικα σε Java, μέσω του NetBeans πραγματοποιούνται εύκολα και οι υπόλοιπες απαραίτητες σχετικές εργασίες όπως η διαχείριση του Apache Tomcat και το deployment των αρχείων στον εξυπηρετητή. Η δημιουργία και διαχείριση της βάσης δεδομένων και το φόρτωμα των txt αρχείων στους αντίστοιχους πίνακες πραγματοποιήθηκε μέσω του γραφικού εργαλείου MySQL Workbench.

## ***5.3 Αρχιτεκτονική της Εφαρμογής***

### ***5.3.1 Το Πρότυπο MVC***

Τα πρότυπα σχεδίασης λογισμικού (software design pattern) αποτελούν μία τυποποιημένη προσέγγιση που χρησιμοποιείται από τους προγραμματιστές για την επίλυση κοινών προγραμματιστικών προβλημάτων. Ένα από τα πλέον δημοφιλή τέτοια πρότυπα για την ανάπτυξη δικτυακών εφαρμογών είναι το πρότυπο MVC (Model – View – Controller) που παρουσιάζεται στο ακόλουθο σχήμα:



Όπως υποδηλώνεται και από το όνομά του, το MVC πρότυπο αποτελείται από τρία στρώματα (layers): το μοντέλο (model), την όψη (view) και τον ελεγκτή (controller).

Το μοντέλο ορίζει το επιχειρηματικό στρώμα (business layer) της εφαρμογής. Αυτό το στρώμα συνήθως υλοποιείται μέσω των JavaBeans, τα οποία αποτελούν επαναχρησιμοποιούμενες κλάσεις της Java που ενθυλακώνουν (encapsulate) πολλαπλά στοιχεία σε ένα ενιαίο object και χαρακτηρίζονται από το ότι είναι serializable, διαθέτουν έναν κατασκευαστή που δεν δέχεται ορίσματα και επιτρέπουν την πρόσβαση στα properties της κλάσης μέσω ειδικών μεθόδων που ονομάζονται getters και setters. Αυτές οι κλάσεις ορίζουν τα δεδομένα των business objects και παρέχουν μεθόδους για τη σχετική επεξεργασία.

Η όψη ορίζει το στρώμα της παρουσίασης της εφαρμογής. Επειδή η χρήση των servlets για τη δημιουργία HTML κώδικα και την αποστολή του στο φυλλομετρητή του χρήστη είναι μία αρκετά δυσκίνητη και άκομμη τεχνική, μία εφαρμογή βασισμένη στο MVC

πρότυπο χρησιμοποιεί συνήθως JSPs για την παρουσίαση των αποτελεσμάτων στον browser.

Ο ελεγκτής διαχειρίζεται τη ροή της εφαρμογής, μία εργασία που διεκπεραιώνεται από τους servlets. Αρχικά, ένας servlet διαβάζει και αποθηκεύει τις παραμέτρους που γίνονται διαθέσιμες από ένα request, το οποίο συνήθως προέρχεται από το στρώμα της όψης. Στη συνέχεια, αν είναι απαραίτητο ο servlet ανανεώνει (update) τα περιεχόμενα του μοντέλου και πιθανόν τα αποθηκεύει στη βάση δεδομένων. Τέλος, ανάλογα με τη λογική που κωδικοποιείται στον servlet, προωθεί το μοντέλο σε κάποια από τις JSPs της εφαρμογής για την τελική προβολή τους.

Στις περισσότερες εφαρμογές παρουσιάζεται η ανάγκη να αντιστοιχισθούν τα δεδομένα του μοντέλου σε κάποιες δομές αποθήκευσης, όμως τα JavaBeans συνήθως δεν παρέχουν μεθόδους αποθήκευσής τους. Γι αυτό το λόγο αναπτύσσονται επιπλέον Java κλάσεις που παρέχουν μεθόδους που υλοποιούν τα διάφορα ενδιάμεσα στάδια και δημιουργούν τις σχετικές δομές για την προσωρινή αποθήκευση των στοιχείων μέσω των sessions της εφαρμογής, καθώς τα JavaBeans μπορούν να γίνουν άμεσα προσβάσιμα από μία JSP σελίδα μέσω των JSP tags της JSTL.

Στη διαδικασία ανάπτυξης σύμφωνα με το MVC πρότυπο, σκοπός είναι τα τρία στρώματα να παραμένουν κατά το δυνατό ανεξάρτητα μεταξύ τους. Έτσι καθίσταται ευκολότερη η τροποποίηση της εφαρμογής σε μεταγενέστερο χρόνο. Για παράδειγμα, αν υπάρξει επιθυμία για πλήρη τροποποίηση της μορφής παρουσίασης των αποτελεσμάτων στο χρήστη, θα χρειαστεί να αλλάξουν μόνο τα στοιχεία του στρώματος όψης, χωρίς να είναι αναγκαία καμία περαιτέρω επέμβαση στο μοντέλο και τον ελεγκτή. Στην πράξη, η πλήρης απομόνωση των τριών στρωμάτων είναι αρκετά δύσκολη, όμως ο στόχος είναι πάντοτε η υλοποίηση να κινείται προς αυτή την κατεύθυνση.

### **5.3.2 Οθόνες**

Η εφαρμογή αποτελείται από δύο βασικά JSP αρχεία, που αποτελούν και το στρώμα όψης της:

#### ***accountsDisplay.jsp***

Αποτελεί την κεντρική οθόνη της εφαρμογής, μέσω της οποίας παρέχονται τα φίλτρα / facets στο αριστερό μέρος και προβάλλονται τα αποτελέσματα με τη μορφή snippets στο κύριο τμήμα της οθόνης. Η υποβολή της φόρμας μέσω του κουμπιού στέλνει ένα post request, το οποίο περιλαμβάνει τις παραμέτρους που επιλέχθηκαν μέσω των radio buttons που αποτελούν τα φίλτρα στον servlet *DisplayAccountsServlet.java*. Ο ίδιος

servlet καλείται και στις περιπτώσεις που ο χρήστης επιλέξει την επόμενη ή κάποια συγκεκριμένη σελίδα των αποτελεσμάτων, με τη διαφορά ότι σε αυτή την περίπτωση οι παράμετροι έχουν ήδη αποθηκευτεί στο τρέχον session και το request προωθεί τον αριθμό της σελίδας που θα προβληθεί. Η σελιδοποίηση σε αυτό το σημείο υλοποιείται με κώδικα JSTL εντός της JSP, όπου μέσω του session είναι γνωστό το πλήθος των συνολικών αποτελεσμάτων και η παράμετρος που ορίζει το πόσα θα προβάλλονται ανά σελίδα. Ο παραπάνω servlet τελικά ανακατευθύνει τη ροή της εφαρμογής ξανά σε αυτή τη σελίδα. Η προβολή των αποτελεσμάτων γίνεται και πάλι μέσω της JSTL, όπου τα δεδομένα αντλούνται απ' ευθείας από το JavaBean που έχει αποθηκευτεί στο session. Στην περίπτωση που επιλεγεί κάποιος συγκεκριμένος λογαριασμός, προωθείται ένα request στο servlet *AccountDetailsServlet.java* με το συγκεκριμένο AccountID ως παράμετρο.

### ***accountDetails.jsp***

Αυτή η JSP προβάλλει τα λεπτομερή αποτελέσματα του λογαριασμού στο χρήστη σε νέο παράθυρο. Η προώθηση στη σελίδα γίνεται μέσω του *AccountDetailsServlet.java* που αναφέρθηκε παραπάνω. Όπως και στην κεντρική JSP, τα αναλυτικά στοιχεία του συγκεκριμένου λογαριασμού είναι αποθηκευμένα στο JavaBean, το ID που χρησιμεύει για τον εντοπισμό του παρέχεται από τον servlet και η προβολή στο χρήστη γίνεται μέσω JSP tags.

### **5.3.3 Έλεγχος Ροής των Εργασιών**

Ο ρόλος των servlets είναι ο έλεγχος της ροής των εργασιών σε μία δικτυακή εφαρμογή. Στα πλαίσια της ανάπτυξης που παρουσιάζεται σε αυτό το κεφάλαιο δημιουργήθηκαν οι δύο ακόλουθες Java κλάσεις που περιλαμβάνονται στο package *servlets*:

#### ***DisplayAccountsServlet.java***

Αποτελεί ουσιαστικά την “καρδιά” της εφαρμογής, κάνοντας override τις μεθόδους *doGet* και *doPost* που καλούνται αυτόματα ανάλογα με το είδος του request. Ο μόνος ρόλος αυτών των δύο μεθόδων είναι να καλέσουν τη μέθοδο *ProcessRequest*.

Κατά την εκκίνησή ο έλεγχος περνά άμεσα στον servlet , ο οποίος παρέχει στην κεντρική σελίδα αποτελέσματα για όλους τους δανειακούς λογαριασμούς. Στην περίπτωση που δεχτεί HTTP request μέσω της υποβολής της φόρμας που περιλαμβάνει τα φίλτρα, δημιουργεί ένα αντικείμενο της κλάσης *HashMap* στο οποίο αποθηκεύει τα ονόματα των παραμέτρων και τις τιμές τους στο session. Αν υπάρχουν ήδη κάποιες τιμές από προηγούμενο φιλτράρισμα τις αντικαθιστά με τις νέες. Εάν δεχθεί το request από κάποιο

link της σελιδοποίησης της κεντρικής σελίδας, αποθηκεύει τον αριθμό της σελίδας που περιέχεται στην παράμετρο, διαφορετικά θέτει την τιμή '1'.

Στη συνέχεια καλεί τις κλάσεις που φορτώνουν το OWL αρχείο και επιστρέφουν τις έννοιες και τις σχέσεις της οντολογίας, οι οποίες αποθηκεύονται σε αντικείμενα της κλάσης *ArrayList* και περιλαμβάνουν αντικείμενα κατάλληλου τύπου που θα περιγραφούν στα πλαίσια του στρώματος του μοντέλου. Τα αντικείμενα των *ArrayLists* και του *HashMap*, δηλαδή η οντολογία και τα επιλεγμένα φίλτρα, τροφοδοτούνται στη στατική μέθοδο μίας κλάσης που σχηματίζει το συζευκτικό ερώτημα και επιστρέφει το αντίστοιχο *String*.

Στο επόμενο βήμα, ο servlet περνά το παραπάνω *String* και την οντολογία στην κλάση που είναι υπεύθυνη για την επανεγγραφή του ερωτήματος. Το επιστρεφόμενο αποτέλεσμα περιέχει την ένωση των ερωτημάτων μετά την επανεγγραφή, σε SQL μορφή πλέον, αποθηκευμένο σε ένα αντικείμενο της κλάσης *HashSet*. Το τελικό βελτιωμένο SQL ερώτημα δημιουργείται και επιστρέφεται ως *String* με την κλήση της κατάλληλης μεθόδου που δέχεται ως παράμετρο το αντικείμενο *HashString*.

Στη συνέχεια ο servlet καλεί τη μέθοδο που θα εφαρμόσει το SQL ερώτημα στη βάση δεδομένων, περνώντας ως παραμέτρους, πέρα από το ίδιο το ερώτημα, το αντικείμενο της κλάσης *HttpSession* που περιέχει το τρέχον session και τις τιμές της σελίδας προς προβολή και του πλήθους των αποτελεσμάτων ανά σελίδα. Το αποτέλεσμα είναι μία *ArrayList* από *JavaBeans* που περιέχουν τα αναλυτικά στοιχεία των λογαριασμών, η οποία αποθηκεύεται στο session. Επίσης ο servlet λαμβάνει πληροφόρηση για το συνολικό πλήθος των αποτελεσμάτων του ερωτήματος.

Τέλος, δημιουργεί ένα αντικείμενο της κλάσης *RequestDispatcher* μέσω του οποίου προωθεί τα αντικείμενα request και response στην κεντρική JSP σελίδα.

#### ***AccountDetailsServlet.java***

Αυτός ο servlet δέχεται request από την κεντρική σελίδα στην περίπτωση που ο χρήστης επιλέξει κάποιον συγκεκριμένο λογαριασμό. Η παράμετρος που διαχειρίζεται είναι το ID αυτού του λογαριασμού, το οποίο περνά σε κατάλληλη μέθοδο έτσι ώστε να παραλάβει και να αποθηκεύσει το *JavaBean* με τα αναλυτικά στοιχεία. Στη συνέχεια, όπως και ο προηγούμενος servlet, προωθεί την αίτηση και την απόκριση και κατευθύνει τη ροή της εφαρμογής, αυτή τη φορά στη σελίδα *accountDetails.jsp*.

### 5.3.4 Διαχείριση Δεδομένων και Ερωτημάτων

Για τις ανάγκες της ανάπτυξης δημιουργήθηκαν κάποιες επιπλέον Java classes και οργανώθηκαν σε δύο ακόμα packages: το package *business* και το package *data*.

Οι κλάσεις του package *business* αποτελούν JavaBeans που έχουν σκοπό την αποθήκευση των δεδομένων που αποτελούν τα αποτελέσματα της αναζήτησης και των χαρακτηριστικών της οντολογίας. Ακολουθεί η περιγραφή τους:

#### *Account.java*

Κάθε αντικείμενο αυτής της κλάσης αποτελεί και έναν δανειακό λογαριασμό. Ορίζονται πεδία / μεταβλητές που αντιστοιχούν σε κάθε αναλυτικό στοιχείο του λογαριασμού και των συσχετισμένων με αυτόν χαρακτηριστικών, όπως τα στοιχεία των δανειοληπτών, της γεωγραφικής περιοχής και πιθανόν των εξασφαλισμένων ακινήτων. Δεδομένου ότι πρόκειται για JavaBean, υλοποιεί το interface *serializable*, παρέχει έναν κατασκευαστή που δεν δέχεται ορίσματα και ορίζει getters και setters μεθόδους για κάθε πεδίο. Οι getters μέθοδοι χρησιμοποιούνται από την JSTL για την προβολή των στοιχείων των λογαριασμών, ενώ οι setters από τις μεθόδους τις κλάσεις που πραγματοποιούν τα τελικά SQL ερωτήματα, για την αποθήκευση των τιμών των πεδίων της βάσης δεδομένων.

#### *OntoClass.java*

Τα αντικείμενα της κλάσης περιλαμβάνουν από μία έννοια της οντολογίας και απεικονίζουν διάφορα χαρακτηριστικά που περιέχονται στο OWL αρχείο όπως το όνομα, την έννοια στην οποία υπάγεται και κάποια OWL annotations που χρησιμοποιήθηκαν για να περιγράψουν το όνομα στα ελληνικά και το facet στο οποίο ανήκει.

#### *OntoProperty.java*

Αντίστοιχα με την παραπάνω κλάση, κάθε αντικείμενό της περιλαμβάνει τα στοιχεία μίας σχέσης της οντολογίας, όπως το όνομα, την έννοια που αποτελεί το *domain* και την έννοια που αποτελεί το *range* της σχέσης.

Στο package *data* ορίζονται οι κλάσεις που αναλαμβάνουν τη διαχείριση των εισόδων και των δεδομένων της εφαρμογής, όπως το φόρτωμα της οντολογίας, τη διαμόρφωση και εφαρμογή των διάφορων ερωτημάτων και την αποθήκευση των αποτελεσμάτων.

#### *OntoLoader.java*

Η κλάση αυτή είναι υπεύθυνη για την ανάγνωση της οντολογίας από το OWL αρχείο. Χρησιμοποιεί το Java OWL API για το φόρτωμα του OWL αρχείου και την εξαγωγή των

απαραίτητων στοιχείων. Υλοποιεί τρεις στατικές μεθόδους, η πρώτη από τις οποίες καλείται εσωτερικά από τις άλλες δύο λαμβάνοντας ως όρισμα το path του αρχείου της οντολογίας και επιστρέφοντας ένα αντικείμενο της κλάσης *OWLOntology*. Η δεύτερη κλάση επιστρέφει ένα αντικείμενο τύπου *ArrayList* στο οποίο αναπαριστά την ιεραρχία των εννοιών με τη μορφή δέντρου. Αντίστοιχα, η τρίτη κλάση επιστρέφει σε παρόμοια δομή τα αντικείμενα με τους ρόλους της οντολογίας. Οι δύο τελευταίες κλάσεις καλούνται από τον κεντρικό servlet.

Ακολουθεί η περιγραφή των τεσσάρων κλάσεων που σχηματίζουν τα συζευκτικά και τα SQL ερωτήματα.

### ***ConjunctiveQueryWriter.java***

Οι μέθοδοι της παραπάνω κλάσης τροφοδοτούνται από τον servlet με τις *ArrayLists* που περιέχουν αντικείμενα τύπου *OntoClass* και *OntoProperty*, καθώς και το αντικείμενο *HashMap* με τα επιλεγμένα φίλτρα. Μέσω της επεξεργασίας αυτών δημιουργείται το συζευκτικό ερώτημα που απεικονίζει τις επιλογές του χρήστη, όπως περιγράφηκε στο προηγούμενο κεφάλαιο, και επιστρέφεται με τη μορφή *String* στον servlet.

### ***QueryRewriter.java***

Υλοποιεί την επανεγγραφή του συζευκτικού ερωτήματος, χρησιμοποιώντας το σχετικό Java API που αναπτύχθηκε από το Image, Video and Multimedia Systems Laboratory (IVML) του τμήματος Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Ε.Μ.Π. Για τη διαδικασία αυτή χρησιμοποιούνται το *String* με το συζευκτικό ερώτημα, το αρχείο OWL με την περιγραφή της οντολογίας και το αρχείο απεικονίσεων (mapping file). Μέσω των κλάσεων του API επιτυγχάνεται η εξαγωγή συμπερασμάτων από την οντολογία, έτσι ώστε το τελικό ερώτημα να περιλαμβάνει έννοιες που δεν απεικονίζονται άμεσα στη βάση δεδομένων. Το αποτέλεσμα που προκύπτει είναι πολλά SQL ερωτήματα, των οποίων η ένωση επιστρέφει τις εγγραφές που αντιστοιχούν στην αναζήτηση που διενεργήθηκε μέσω των φίλτρων. Το σύνολο των ερωτημάτων αποθηκεύεται σε ένα αντικείμενο της κλάσης *HashSet* που αποτελείται από *Strings* και επιστρέφεται στον servlet.

### ***RapidQueryRewriter.java***

Καλείται από τις μεθόδους της κλάσης *QueryRewriter* που μόλις περιγράφηκε. Χρησιμοποιούνται επίσης οι διάφορες κλάσεις του API που αναφέρθηκε παραπάνω.

### ***SQLQueryWriter.java***

Οι μέθοδοι της κλάσης χρησιμοποιούν το αποτέλεσμα της *QueryRewriter*. Το σύνολο των περιεχομένων του *HashMap* μετατρέπεται σε ένα *String* που αποτελεί το τελικό ερώτημα που θα εφαρμοστεί στη βάση δεδομένων. Το αποτέλεσμα είναι ένα *union* από *selects*. Επιπλέον, το κάθε *select* τροποποιείται έτσι ώστε να εκτελείται με πιο αποδοτικό τρόπο στη βάση. Αυτό γίνεται μέσω της απαλοιφής κάποιων περιττών *joins* που υπάρχουν στο αρχικό ερώτημα του *rewriter* και με την απλοποίηση του ερωτήματος χωρίς τη δημιουργία ενδιάμεσων πινάκων.

Η διαδικασία που λειτουργούν οι τέσσερις παραπάνω κλάσεις και η περιγραφή της επεξεργασίας που διενεργείται για τη δημιουργία των ερωτημάτων θα περιγραφεί αναλυτικότερα στην επόμενη παράγραφο αυτού του κεφαλαίου.

Οι τρεις επόμενες κλάσεις εφαρμόζουν το τελικό SQL ερώτημα στη βάση δεδομένων και διαχειρίζονται τα επιστρεφόμενα αποτελέσματα.

### ***AccountDB.java***

Χρησιμοποιείται από τους δύο *servlets* *DisplayAccountsServlet* και *AccountDetailsServlet*, εφαρμόζοντας το τελικό SQL ερώτημα στη βάση δεδομένων. Στην πρώτη περίπτωση αποθηκεύει τα αποτελέσματα σε ένα αντικείμενο *ArrayList* που αποτελείται από αντικείμενα της κλάσης *Account*, λαμβάνοντας υπ' όψιν το πλήθος των εγγραφών που πρέπει να επιστραφεί, καθώς και τη σελίδα των αποτελεσμάτων προς παρουσίαση. Γι αυτό το σκοπό χρησιμοποιεί τις σχετικές παραμέτρους και προσθέτει στο ερώτημα ένα *limit clause* με τις αντίστοιχες τιμές του *offset* και του πλήθους. Στη δεύτερη περίπτωση επιστρέφει ένα αντικείμενο *Account* με τα αναλυτικά στοιχεία του συγκεκριμένου λογαριασμού που επιλέχθηκε προς προβολή από το χρήστη. Και στις δύο περιπτώσεις χρησιμοποιεί τις *setters* μεθόδους της κλάσης *Account* για να αναθέσει τις κατάλληλες τιμές στα σχετικά πεδία των αντικειμένων της.

### ***DBUtil.java***

Παρέχει μεθόδους για τον τερματισμό της σύνδεσης με το σύνολο των αποτελεσμάτων. Χρησιμοποιείται αποκλειστικά από τις μεθόδους της κλάσης *AccountDB*.

### ***ConnectionPool.java***

Υλοποιεί την έναρξη και τη λήξη της σύνδεσης με τη βάση δεδομένων, δημιουργώντας το στιγμιότυπο ενός *connection pool* που χρησιμοποιείται από την κλάση *AccountDB*.



## 5.4 Κατασκευή και Βελτιστοποίηση Ερωτημάτων

Σε αυτή την παράγραφο περιγράφεται αναλυτικά η διαδικασία δημιουργίας και επεξεργασίας των ερωτημάτων. Η παρουσίαση θα γίνει με τη χρήση ενός απλού παραδείγματος. Έστω ότι ο χρήστης αναζητά τους λογαριασμούς δανείων προκαθορισμένης διάρκειας, οι οποίοι εξασφαλίζονται από ακίνητο που διαθέτει επαγγελματική χρήση. Η επιλογή αυτή γίνεται μέσω των δύο εκ των τεσσάρων κεντρικών facets της εφαρμογής, επιλέγοντας τα κατάλληλα radio buttons και πατώντας το κουμπί για την εφαρμογή των φίλτρων.

### 5.4.1 Το Συζευκτικό Ερώτημα

Το ερώτημα σχηματίζεται από τη στατική μέθοδο *getConjunctiveQuery* της κλάσης *ConjunctiveQueryWriter*, η οποία δέχεται ως ορίσματα τις δομές *ArrayList* με τα επεξεργασμένα χαρακτηριστικά της οντολογίας και το *HashMap* με τα ονόματα και τις τιμές των επιλεγμένων φίλτρων. Τα ονόματα των παραμέτρων που αντιστοιχούν στα radio buttons είναι όμοια με τα ονόματα των εννοιών της οντολογίας. Με αυτόν τον τρόπο η μέθοδος εντοπίζει τις έννοιες και τις σχέσεις που εμπλέκονται και σχηματίζει ένα συζευκτικό ερώτημα με την ακόλουθη μορφή:

```
Q(?f1, ?f3) <- <http://LoansOnto#LoanTerm>(?f1),  
<http://LoansOnto#REUsageProfessional>( ?f3),  
<http://LoansOnto#hasCollateral>(?f1, ?f3)
```

Αντίστοιχα ερωτήματα δημιουργούνται και στις περιπτώσεις πιο περίπλοκων αναζητήσεων. Αν στο παραπάνω παράδειγμα, τεθούν οι επιπλέον περιορισμοί ο δανειολήπτης να απασχολείται ως ελεύθερος επαγγελματίας, να είναι μηχανικός στο επάγγελμα και να κατοικεί στη Θεσσαλία, το συζευκτικό ερώτημα που προκύπτει είναι το ακόλουθο:

```
Q(?f1, ?f3, ?f2, ?f4) <- <http://LoansOnto#OccTypeFreelancer>(?f2),  
<http://LoansOnto#RegThessalia>( ?f4), <http://LoansOnto#LoanTerm>(?f1),  
<http://LoansOnto#OccCatEngineer>(?f2),  
<http://LoansOnto#REUsageProfessional>(?f3),  
<http://LoansOnto#hasCollateral>(?f1, ?f3), <http://LoansOnto#ownedBy>(?f1, ?f2),  
<http://LoansOnto#leavesIn>(?f2, ?f4)
```

### 5.4.2 Επανεγγραφή του Ερωτήματος

Η επανεγγραφή του συζευκτικού ερωτήματος υλοποιείται από τη μέθοδο *getRewrittenQuery* της κλάσης *QueryRewriter*, με χρήση βοηθητικών μεθόδων της κλάσης *RapidQueryRewriter*. Σε αυτό το σημείο χρησιμοποιείται το σχετικό API που αναπτύχθηκε από το IVML. Η μέθοδος λαμβάνει ως ορίσματα το String που αναπαριστά το συζευκτικό ερώτημα, το αρχείο της οντολογίας και το αρχείο των απεικονίσεων. Το τελευταίο αποτελεί ένα απλό txt αρχείο που αντιστοιχίζει διάφορους όρους της οντολογίας σε ερωτήματα προς τη βάση δεδομένων. Για παράδειγμα, οι έννοιες *Mortgage* και *Consumer* και ο ρόλος *hasCollateral* απεικονίζονται ως εξής:

<code>&lt;http://LoansOnto#Mortgage&gt;&lt;http://LoansOnto/ID@[0]&gt;</code>	<code>select Loans.AccountID from Bank.Loans where LoanType = 'M'</code>
<code>&lt;http://LoansOnto#Consumer&gt;&lt;http://LoansOnto/ID@[0]&gt;</code>	<code>select Loans.AccountID from Bank.Loans where LoanType = 'C'</code>
<code>&lt;http://LoansOnto#hasCollateral&gt;&lt;http://LoansOnto/ID@[0]&gt; &lt;http://LoansOnto/ID@[1]&gt;</code>	<code>select Loans.AccountID, Loans.RealEstateID from Bank.Loans</code>

Δεν υφίσταται άμεση απεικόνιση στη βάση για όλες τις έννοιες της οντολογίας. Στο εξεταζόμενο παράδειγμα, η έννοια του δανείου προκαθορισμένης διάρκειας δεν διακρίνεται από κάποιο συγκεκριμένο πεδίο, όμως έχει οριστεί στην οντολογία μέσω αξιωμάτων υπαγωγής. Πρόκειται για μία γενικότερη έννοια που περιλαμβάνει τα στεγαστικά και καταναλωτικά δάνεια, τα οποία προσδιορίζονται στα διαθέσιμα στοιχεία μέσω της τιμής συγκεκριμένου πεδίου ενός πίνακα της βάσης, του *LoanType*. Σε αυτή την περίπτωση, το σύστημα επανεγγραφής μέσω της οντολογίας εντοπίζει τις υποέννοιες και με τη βοήθεια του αρχείου απεικονίσεων σχηματίζει δύο SQL ερωτήματα, η ένωση των οποίων επιστρέφει πράγματι τα ζητούμενα δάνεια. Τα ερωτήματα που σχηματίζονται για το εξεταζόμενο παράδειγμα είναι τα ακόλουθα:

```
SELECT T2.C0, T2.C1 FROM (select Loans.AccountID as C0 from Bank.Loans where LoanType = 'C') AS  
T0, (select Loans.RealEstateID as C0 from Bank.Loans where RealEstateType = 3) AS T1, (select  
Loans.AccountID as C0, Loans.RealEstateID as C1 from Bank.Loans) AS T2 WHERE T0.C0 = T2.C0 AND  
T1.C0 = T2.C1
```

και

```
SELECT T2.C0, T2.C1 FROM (select Loans.AccountID as C0 from Bank.Loans where LoanType = 'M') AS  
T0, (select Loans.RealEstateID as C0 from Bank.Loans where RealEstateType = 3) AS T1, (select
```

*Loans.AccountID as C0, Loans.RealEstateID as C1 from Bank.Loans) AS T2 WHERE T0.C0 = T2.C0 AND T1.C0 = T2.C1*

Όπως γίνεται κατανοητό, στις περιπτώσεις όπου στο ερώτημα υπάρχουν περισσότερες από μία έννοιες που δεν απεικονίζονται άμεσα αλλά συντίθενται από άλλες υποέννοιες, το σύνολο των ερωτημάτων που σχηματίζουν την ένωση θα αποτελείται από όλους τους δυνατούς συνδυασμούς των εννοιών / πεδίων που καθορίζουν την κάθε γενικότερη έννοια.

### **5.4.3 Βελτιστοποίηση του Ερωτήματος**

Όπως προκύπτει από το παράδειγμα της προηγούμενη παραγράφου, ο *rewriter* σχηματίζει το κάθε ερώτημα της ένωσης μέσω της κατασκευής ενδιάμεσων πινάκων στους οποίους γίνεται *join*. Αυτό συμβαίνει διότι το σύστημα επανεγγραφής αποτελεί μία γενική υλοποίηση, η οποία προφανώς δεν λαμβάνει υπ' όψιν τις ιδιαιτερότητες της συγκεκριμένης βάσης δεδομένων. Σε περιπτώσεις όπως αυτή της εφαρμογής που αναπτύχθηκε, όπου γίνεται διαχείριση εκατοντάδων χιλιάδων εγγραφών από τη βάση δεδομένων, αυτή η μορφή ερωτημάτων αργεί σημαντικά να εκτελεστεί στο RDBMS, καθώς στο *join* των ενδιάμεσων πινάκων δεν είναι δυνατόν να εφαρμοστούν οι δείκτες που επιταχύνουν πάρα πολύ τα *selects* και τα διάφορα *clauses*. Επιπλέον, σε κάποιες περιπτώσεις όπως αυτή του παραδείγματος, τα πεδία που συμμετέχουν στη σχέση της οντολογίας μπορεί να βρίσκονται στον ίδιο πίνακα, οπότε το *join* δεν είναι καν απαραίτητο, αφού το ίδιο ερώτημα μπορεί να πραγματοποιηθεί και με ένα απλό *select*.

Για τη βελτίωση του χρόνου εκτέλεσης υλοποιήθηκαν κατάλληλες μέθοδοι στην κλάση *SQLQueryWriter*, μέσω των οποίων τα ερωτήματα μετασχηματίζονται σε απλούστερη αλλά πλήρως ισοδύναμη μορφή. Η εφαρμογή των τελικών ερωτημάτων στη βάση δεδομένων, μετά το μετασχηματισμό, επιστρέφει αποτελέσματα σε χρόνους που κυμαίνονται μεταξύ μερικών εκατοστών του δευτερολέπτου μέχρι και λόγων δευτερολέπτων. Επιπλέον, σχηματίζεται το SQL *union* και προστίθενται στο *select* τα επιπλέον πεδία που περιέχουν τα λεπτομερή χαρακτηριστικά του κάθε λογαριασμού.

Το ερώτημα του συγκεκριμένου παραδείγματος διαμορφώνεται ως εξής:

```
select Loans.AccountID, Loans.RealEstateID, Customers.CustomerID, Loans.LoanType,  
Loans.RealEstateType ...  
from Loans  
where Loans.LoanType = 'M' and Loans.RealEstateType = 3  
union
```

```
select Loans.AccountID, Loans.RealEstateID, Customers.CustomerID, Loans.LoanType,  
Loans.RealEstateType ...  
from Loans  
where Loans.LoanType = 'C' and Loans.RealEstateType = 3
```

Το επιστρεφόμενο αποτέλεσμα είναι ένα `String` που περιέχει το παραπάνω ερώτημα, το οποίο τροφοδοτείται στις μεθόδους της κλάσης `AccountDB` για την εφαρμογή του στη βάση δεδομένων.

# 6

## Οθόνες Εφαρμογής – Αποτελεσμάτων

Σε αυτό το κεφάλαιο θα παρουσιαστούν διάφορες οθόνες αποτελεσμάτων της εφαρμογής, ο τρόπος λειτουργίας των φίλτρων, η σελιδοποίηση και η παρουσίαση των αναλυτικών στοιχείων για συγκεκριμένες περιπτώσεις. Για τη χρήση της εφαρμογής μπορεί να χρησιμοποιηθεί οποιοσδήποτε σύγχρονος web browser. Οι εικόνες που ακολουθούν στη συνέχεια προέρχονται από τον Chrome.

### 6.1 Κεντρική Οθόνη

Με το άνοιγμα της εφαρμογής εμφανίζεται η κεντρική οθόνη των αποτελεσμάτων. Επειδή δεν έχει εφαρμοστεί ακόμα κάποιο φίλτρο, όλοι οι δανειακοί λογαριασμοί περιλαμβάνονται στα αποτελέσματα.

Αποτελέσματα Λογαριασμών

Πλήθος αποτελεσμάτων: 203 796

Συνοπτικά Στοιχεία

Λογαριασμός	132014740641
Στοιχεία Δανείου: Είδος Χρήσης: Στεγαστικό Δάνειο. Ημερομηνία Ενεργοποίησης: 12 Οκτ 2001. Τρέχον Υπόλοιπο: 650,96 €.— Στοιχεία Δανειολήπτη: Είδος Αποσφάγισης: Ασπί. Επάγγελμα: ΛΟΙΠΑ — Περσική Κατοικίας Δανειολήπτη: Νομός: ΑΤΤΙΚΗΣ — Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: - Τρέχουσα Αξία Ακινήτου: -	
Λογαριασμός	132014760705
Στοιχεία Δανείου: Είδος Χρήσης: Στεγαστικό Δάνειο. Ημερομηνία Ενεργοποίησης: 29 Ιαν 2002. Τρέχον Υπόλοιπο: 53.434,42 €.— Στοιχεία Δανειολήπτη: Είδος Αποσφάγισης: Ελεύθερος Επαγγελματίας. Επάγγελμα: ΛΟΙΠΑ — Περσική Κατοικίας Δανειολήπτη: Νομός: ΣΑΜΟΥ — Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: Μικτή. Τρέχουσα Αξία Ακινήτου: 616.523,68 €	
Λογαριασμός	132014763006
Στοιχεία Δανείου: Είδος Χρήσης: Στεγαστικό Δάνειο. Ημερομηνία Ενεργοποίησης: 19 Οκτ 2001. Τρέχον Υπόλοιπο: 7.588,70 €.— Στοιχεία Δανειολήπτη: Είδος Αποσφάγισης: Υπάλληλος. Επάγγελμα: ΛΟΙΠΑ — Περσική Κατοικίας Δανειολήπτη: Νομός: ΑΤΤΙΚΗΣ — Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: Κατοικία. Τρέχουσα Αξία Ακινήτου: 148.019,15 €	
Λογαριασμός	132014777073
Στοιχεία Δανείου: Είδος Χρήσης: Στεγαστικό Δάνειο. Ημερομηνία Ενεργοποίησης: 12 Οκτ 2001. Τρέχον Υπόλοιπο: 19.854,86 €.— Στοιχεία Δανειολήπτη: Είδος Αποσφάγισης: Υπάλληλος. Επάγγελμα: ΛΟΙΠΑ — Περσική Κατοικίας Δανειολήπτη: Νομός: ΣΕΡΡΩΝ — Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: Κατοικία. Τρέχουσα Αξία Ακινήτου: 127.856,43 €	
Λογαριασμός	132014786808
Στοιχεία Δανείου: Είδος Χρήσης: Στεγαστικό Δάνειο. Ημερομηνία Ενεργοποίησης: 4 Δεκ 2001. Τρέχον Υπόλοιπο: 2.400,69 €.— Στοιχεία Δανειολήπτη: Είδος Αποσφάγισης: Υπάλληλος. Επάγγελμα: ΛΟΙΠΑ — Περσική Κατοικίας Δανειολήπτη: Νομός: ΦΘΙΩΤΙΔΑΣ — Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: Κατοικία. Τρέχουσα Αξία Ακινήτου: 103.709,91 €	
Λογαριασμός	132014856174
Στοιχεία Δανείου: Είδος Χρήσης: Στεγαστικό Δάνειο. Ημερομηνία Ενεργοποίησης: 26 Νοε 2001. Τρέχον Υπόλοιπο: 15.516,83 €.— Στοιχεία Δανειολήπτη: Είδος Αποσφάγισης: Υπάλληλος. Επάγγελμα: ΛΟΙΠΑ — Περσική Κατοικίας Δανειολήπτη: Νομός: ΑΤΤΙΚΗΣ — Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: Κατοικία. Τρέχουσα Αξία Ακινήτου: 69.259,74 €	
Λογαριασμός	132014809341
Στοιχεία Δανείου: Είδος Χρήσης: Στεγαστικό Δάνειο. Ημερομηνία Ενεργοποίησης: 12 Μαρ 2002. Τρέχον Υπόλοιπο: 10.066,30 €.— Στοιχεία Δανειολήπτη: Είδος Αποσφάγισης: Ελεύθερος Επαγγελματίας. Επάγγελμα: ΛΟΙΠΑ — Περσική Κατοικίας Δανειολήπτη: Νομός: ΚΕΦΑΛΛΗΝΙΑΣ — Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: Κατοικία. Τρέχουσα Αξία Ακινήτου: 79.121,36 €	

Στο άνω μέρος του κεντρικού τμήματος εμφανίζεται το πλήθος των αποτελεσμάτων που προκύπτουν από την εφαρμογή των φίλτρων.

Πλήθος αποτελεσμάτων: 318

---

Στο κατώτερο τμήμα προβάλλεται ο μηχανισμός σελιδοποίησης. Ο χρήστης μπορεί να δει τον αριθμό των σελίδων που αντιστοιχούν στο πλήθος των αποτελεσμάτων, καθώς και πλοηγηθεί στην προηγούμενη ή επόμενη σελίδα ή να επιλέξει τη σελίδα προς προβολή.

Σελίδες αποτελεσμάτων: 87



Η σελίδα που προβάλλεται στην οθόνη εμφανίζεται σκιασμένη σε σχέση με τις υπόλοιπες. Με αυτόν τον τρόπο παρέχει την πληροφορία του αριθμού της σελίδας αποτελεσμάτων που βρίσκεται ο χρήστης κατά την πλοήγησή του στα στοιχεία.

## 6.2 Φίλτρα και Αποτελέσματα

Το μενού επιλογής των φίλτρων εμφανίζεται στο αριστερό τμήμα της κεντρικής οθόνης. Κατά την εκκίνηση της εφαρμογής, όπως και στις περιπτώσεις όπου έχει προηγηθεί καθαρισμός των φίλτρων, παρουσιάζονται στο χρήστη μόνο οι επιλογές που σχετίζονται με το κεντρικό facet, δηλαδή τα χαρακτηριστικά / υποέννοιες του δανείου. Για τα υπόλοιπα facets εμφανίζεται μόνο το όνομά τους, το οποίο έχει τη μορφή link. Κάνοντας click σε οποιοδήποτε όνομα facet εμφανίζονται οι σχετιζόμενες με αυτό επιλογές με τη μορφή expandable menu. Επίσης υπάρχει και η δυνατότητα απόκρυψης του πρώτου facet, όπως και των υπολοίπων, με την ίδια διαδικασία.

**Φίλτρα πλοήγησης**

- **Δάνειο**
  - Είδος Δανείου*
    - Προκαθορισμένης Διάρκειας
      - Στεγαστικό Δάνειο
      - Καταναλωτικό Δάνειο
    - Ανακυκλούμενης Πίστωσης
      - Πιστωτική Κάρτα
      - Ανοιχτό Όριο
  - Είδος Επποκίου*
    - Σταθερό
    - Κυμαινόμενο
  - Καθυστερημένες Δόσεις*
    - Το Πολύ Μία
    - Δύο ή Τρεις
    - Πάνω από Τρεις
  - Σκοπός Χρηματοδότησης*
    - Αγορά Ακινήτου
    - Ανέγερση
    - Επισκευή
    - Μεταφορά άλλου Δανείου
    - Ρύθμιση άλλου Δανείου
    - Αγορά Αυτοκινήτου
    - Αγορά Λοπών Προϊόντων
- **Δανειολήπτης**
- **Περιοχή Κατοικίας Δανειολήπτη**
- **Συσχετιζόμενο Ακίνητο**

**Φίλτρα πλοήγησης**

- **Δάνειο**
- **Δανειολήπτης**
- **Περιοχή Κατοικίας Δανειολήπτη**
- **Συσχετιζόμενο Ακίνητο**

*Είδος Απασχόλησης*

- Υπάλληλος
- Ελεύτερος Επαγγελματίας
- Συνταξιούχος
- Εισοδηματίας
- Λοπά

*Κατηγορία Επαγγέλματος*

- Μηχανικός
- Εκπαιδευτικός
- Υπάλληλος Γραφείου
- Ελεύτερο Επάγγελμα
- Νομικό Επάγγελμα
- Ιατρικό Επάγγελμα
- Τεχνικό Επάγγελμα
- Καλλιτεχνικό Επάγγελμα
- Λοπά Επαγγέλματα

- **Περιοχή Κατοικίας Δανειολήπτη**
  - Περιφέρεια*
    - Αττική
    - Ανατολική Μακεδονία - Θράκη
    - Κεντρική Μακεδονία
    - Δυτική Μακεδονία
    - Ήπειρος

Για μία συγκεκριμένη επιλογή φίλτρων παρόμοια με την παραπάνω, τα αποτελέσματα παρουσιάζονται στην επόμενη εικόνα. Σημειώνεται ότι τα facets που έχουν χρησιμοποιηθεί κατά την αναζήτηση από το χρήστη παραμένουν ανοιχτά και μετά την επιστροφή των αποτελεσμάτων.

- Υψηλό
- Είδος Απασχόλησης**
- Υπάλληλος
- Ελεύθερος Επαγγελματίας
- Συνταξιούχος
- Εισοδηματίας
- Λοιπά
- Κατηγορία Επαγγέλματος**
- Μηχανικός
- Εκπαιδευτικός
- Υπάλληλος Γραφείου
- Ελεύθερο Επάγγελμα
- Νομικό Επάγγελμα
- Ιατρικό Επάγγελμα
- Τεχνικό Επάγγελμα
- Καλλιτεχνικό Επάγγελμα
- Λοιπά Επαγγέλματα
- Περιοχή Κατοικίας Δανειολήπτη**
- Περιφέρεια**
- Αττική
- Ανατολική Μακεδονία - Θράκη
- Κεντρική Μακεδονία
- Δυτική Μακεδονία
- Ήπειρος
- Θεσσαλία
- Δυτική Ελλάδα
- Στερεά Ελλάδα
- Πελοπόννησος
- Νησιά Ιονίου
- Βόρειο Αιγαίο
- Νησιά Αιγαίου
- Νότιο Αιγαίο
- Κρήνη
- Συσχετιζόμενο Ακίνητο**

Λογαριασμός: 139779763335

Στοιχεία Δανείου: Είδος Χορήγησης: *Ανοχή Πίστωση*, Ημερομηνία Ενεργοποίησης: 5 Ιουλ 2007, Τρέχον Υπόλοιπο: 1.651,24 €, --  
 Στοιχεία Δανειολήπτη: Είδος Απασχόλησης: *Ελεύθερος Επαγγελματίας*, Επάγγελμα: *ΠΟΛΙΤΙΚΟΣ ΜΗΧΑΝΙΚΟΣ* -- Περιοχή Κατοικίας  
 Δανειολήπτη: Νομός: *ΡΕΘΥΜΝΗΣ* -- Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: -, Τρέχουσα Αξία Ακινήτου: -

Λογαριασμός: 139784970768

Στοιχεία Δανείου: Είδος Χορήγησης: *Στεγαστικό Δάνειο*, Ημερομηνία Ενεργοποίησης: 30 Ιουν 2006, Τρέχον Υπόλοιπο: 29.790,97 €, --  
 Στοιχεία Δανειολήπτη: Είδος Απασχόλησης: *Ελεύθερος Επαγγελματίας*, Επάγγελμα: *ΠΟΛΙΤΙΚΟΣ ΜΗΧΑΝΙΚΟΣ* -- Περιοχή Κατοικίας  
 Δανειολήπτη: Νομός: *ΗΡΑΚΛΕΙΟΥ* -- Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: *Κατοικία*, Τρέχουσα Αξία Ακινήτου:  
 121.600,00 €

Λογαριασμός: 139784974332

Στοιχεία Δανείου: Είδος Χορήγησης: *Στεγαστικό Δάνειο*, Ημερομηνία Ενεργοποίησης: 15 Δεκ 2006, Τρέχον Υπόλοιπο: 32.142,63 €, --  
 Στοιχεία Δανειολήπτη: Είδος Απασχόλησης: *Ελεύθερος Επαγγελματίας*, Επάγγελμα: *ΠΟΛΙΤΙΚΟΣ ΜΗΧΑΝΙΚΟΣ* -- Περιοχή Κατοικίας  
 Δανειολήπτη: Νομός: *ΗΡΑΚΛΕΙΟΥ* -- Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: *Κατοικία*, Τρέχουσα Αξία Ακινήτου:  
 121.600,00 €

Λογαριασμός: 139784980008

Στοιχεία Δανείου: Είδος Χορήγησης: *Στεγαστικό Δάνειο*, Ημερομηνία Ενεργοποίησης: 12 Ιουν 2007, Τρέχον Υπόλοιπο: 28.031,54 €, --  
 Στοιχεία Δανειολήπτη: Είδος Απασχόλησης: *Ελεύθερος Επαγγελματίας*, Επάγγελμα: *ΠΟΛΙΤΙΚΟΣ ΜΗΧΑΝΙΚΟΣ* -- Περιοχή Κατοικίας  
 Δανειολήπτη: Νομός: *ΗΡΑΚΛΕΙΟΥ* -- Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: *Κατοικία*, Τρέχουσα Αξία Ακινήτου:  
 121.600,00 €

Λογαριασμός: 139784987763

Στοιχεία Δανείου: Είδος Χορήγησης: *Στεγαστικό Δάνειο*, Ημερομηνία Ενεργοποίησης: 14 Νοε 2007, Τρέχον Υπόλοιπο: 37.229,53 €, --  
 Στοιχεία Δανειολήπτη: Είδος Απασχόλησης: *Ελεύθερος Επαγγελματίας*, Επάγγελμα: *ΠΟΛΙΤΙΚΟΣ ΜΗΧΑΝΙΚΟΣ* -- Περιοχή Κατοικίας  
 Δανειολήπτη: Νομός: *ΗΡΑΚΛΕΙΟΥ* -- Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: *Κατοικία*, Τρέχουσα Αξία Ακινήτου:  
 205.200,00 €

Λογαριασμός: 162291036862815

Στοιχεία Δανείου: Είδος Χορήγησης: *Πιστωτική Κάρτα*, Ημερομηνία Ενεργοποίησης: 20 Ιουλ 1999, Τρέχον Υπόλοιπο: 0,00 €, --  
 Στοιχεία Δανειολήπτη: Είδος Απασχόλησης: *Ελεύθερος Επαγγελματίας*, Επάγγελμα: *ΗΛΕΚΤΡΟΛΟΓΟΣ ΜΗΧΑΝΙΚΟΣ* -- Περιοχή  
 Κατοικίας Δανειολήπτη: Νομός: *ΡΕΘΥΜΝΗΣ* -- Στοιχεία Συνδεδεμένου Ακινήτου: Χρήση Ακινήτου: -, Τρέχουσα Αξία Ακινήτου: -

Σελίδες αποτελεσμάτων: 10

◀ 1 2 3 4 5 6 7 8 9 10 ▶

## 6.3 Προβολή Αναλυτικών Στοιχείων

Με την επιλογή ενός συγκεκριμένου λογαριασμού από τα παραπάνω αποτελέσματα προβάλλονται σε νέο παράθυρο τα αναλυτικά του στοιχεία, όπως παρουσιάζεται στη συνέχεια:



Αποτελέσματα Λογαριασμού x Αναλυτικά Στοιχεία x

localhost:8080/LoansApp/accountDetails?id=139784987763

### Αναλυτικά Στοιχεία

Στοιχεία Λογαριασμού	
Αριθμός Λογαριασμού: 139784987763	Είδος Χορήγησης: Στεγαστικό Δάνειο
Σκοπός Χορήγησης: Επισκευή Ακινήτου	Ημερομηνία Ενεργοποίησης: 14 Νοε 2007
Αρχικό Ποσό Χορήγησης / Όριο: 85.000,00 €	Τρέχον Υπόλοιπο: 37.229,53 €
Επιτόκιο: 1,67%	Είδος Επιτοκίου: Κυμανόμενο
Νομισμα Χορήγησης: Ευρώ	Συνολική Διάρκεια σε Μήνες: 120
Πλήθος Καθυστερημένων Δόσεων: 1	Ύπαρξη Εγγυητή: Ναι
Στοιχεία Δανειολήπτη	
Κωδικός Πελάτη: 323536198098	Είδος Δανειολήπτη: Φυσικό Πρόσωπο
Φύλο: Γυναίκα	Μορφωτικό Επίπεδο: Άγνωστο
Είδος Απασχόλησης: Ελεύθερος Επαγγελματίας	Επάγγελμα: ΠΟΛΙΤΙΚΟΣ ΜΗΧΑΝΙΚΟΣ
Περιοχή Κατοικίας Δανειολήπτη	
Περιφέρεια: ΚΡΗΤΗ	Νομός: ΗΡΑΚΛΕΙΟΥ
Στοιχεία Προσημειωμένου Ακινήτου	
Κωδικός Ακινήτου: 13219207158	
Είδος Χρήσης Ακινήτου: Κατοικία	Τρέχουσα Αξία Ακινήτου: 205.200,00 €



# 7

## *Επίλογος*

### *7.1 Σύνοψη*

Στην παρούσα διπλωματική εργασία υλοποιήθηκε μία εφαρμογή που συνδυάζει σύγχρονες και ευφυείς μεθόδους αναπαράστασης γνώσης, συγκεκριμένα ένα τμήμα της οικογένειας γλωσσών Περιγραφικής Λογικής, με παραδοσιακές υλοποιήσεις αποθήκευσης της πληροφορίας όπως είναι οι σχεσιακές βάσεις δεδομένων. Ένα μέρος της γνώσης αναπαρίσταται μέσω οντολογίας που αναπτύχθηκε και περιγράφηκε στη γλώσσα OWL, η οποία είναι βασισμένη στις Περιγραφικές Λογικές. Η γνώση αυτή περιλαμβάνει τον ορισμό γενικών και ειδικότερων εννοιών, τις σχέσεις στις οποίες συμμετέχουν αυτές οι έννοιες, καθώς και το σώμα ορολογίας που δηλώνει την υπαγωγή και την ισοδυναμία των εννοιών. Ο μεγάλος όγκος της πληροφορίας βρίσκεται αποθηκευμένος σε μία κλασσική βάση δεδομένων, όπου με τη χρήση της οντολογίας ουσιαστικά σχηματίζεται το σώμα ισχυρισμών της βάσης γνώσης.

Η Περιγραφική Λογική DL-Lite που χρησιμοποιήθηκε παρέχει δυνατότητες γρήγορης και αποδοτικής εκτέλεσης συζευκτικών ερωτημάτων. Αυτά τα ερωτήματα σχηματίζονται μέσω της διεπαφής της διαδικτυακής εφαρμογής που αναπτύχθηκε, με τη βοήθεια faceted φίλτρων, όπου το κάθε facet αναπαριστά μία συγκεκριμένη κατηγορία πληροφοριών. Για την επανεγγραφή των συζευκτικών ερωτημάτων, η οποία αποτελεί τον μηχανισμό

συλλογιστικής στη βάση γνώσης και του σχηματισμού ένωσης τελικών SQL ερωτημάτων που επιστρέφουν τα επιθυμητά αποτελέσματα, χρησιμοποιήθηκε σχετικό σύστημα που έχει αναπτυχθεί από το εργαστήριο IVML της σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του Ε.Μ.Π.

Μέσω του συνδυασμού των παραπάνω τεχνολογιών επιτυγχάνεται ενοποίηση της πληροφορίας (data integration), ενώ παράλληλα παρέχεται η ευελιξία ορισμού νέων εννοιών ή τροποποίησης των υφιστάμενων στην οντολογία, ανάλογα με τις ανάγκες που μπορεί να παρουσιαστούν, χωρίς σημαντικές επεμβάσεις στα υπόλοιπα μέρη του συστήματος.

## **7.2 Μελλοντικές Επεκτάσεις**

Για την ενίσχυση της ευελιξίας μέσω της τροποποίησης των οριζόμενων εννοιών και σχέσεων της οντολογίας, όπως αναφέρθηκε στην προηγούμενη παράγραφο, μία χρήσιμη επέκταση της εφαρμογής είναι η δημιουργία ενός αυτόματου μηχανισμού σχηματισμού των φίλτρων. Κάτι τέτοιο μπορεί να επιτευχθεί μέσω της χρήσης OWL annotation στις έννοιες που θα περιγράφουν τα ονόματα των επιλογών και τα facets στα οποία αντιστοιχούν οι ρόλοι. Επειδή η λειτουργικότητα της εφαρμογής υλοποιείται στην πλευρά του εξυπηρετητή, για να επιτευχθεί κάτι τέτοιο μπορούν να χρησιμοποιηθούν τεχνολογίες δημιουργίας ασύγχρονων διαδικτυακών εφαρμογών όπως η Ajax.

Τέλος, μία επιπλέον επέκταση της υλοποίησης θα μπορούσε να αποτελεί η περαιτέρω βελτιστοποίηση των ερωτημάτων που προκύπτουν από το σύστημα επανεγγραφής, με την απαλοιφή κάποιων ερωτημάτων στην τελική ένωση (union). Αυτό μπορεί να γίνει στις περιπτώσεις όπου δύο περισσότερα ερωτήματα που συμμετέχουν στην ένωση διαφοροποιούνται μόνο στην επιλεγθείσα τιμή κάποιου συγκεκριμένου πεδίου, μέσω του μετασχηματισμού τους σε ένα ερώτημα στο οποίο θα χρησιμοποιείται ο τελεστής της SQL 'in'. Αυτή η προσθήκη θα μπορούσε να βελτιώσει ακόμα περισσότερο τους χρόνους εκτέλεσης των τελικών ερωτημάτων στη βάση δεδομένων.

# 8

## *Βιβλιογραφία*

- [1] Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, 2003.
- [2] R. Brachman and H. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann, 2004.
- [3] S. Staab, R. Studer. *Handbook on Ontologies*. Springer, 2009.
- [4] Frank Van Harmelen, Vladimir Lifschitz, Bruce Porter. *Handbook of Knowledge Representation*. Elsevier, 2008.
- [5] Alessandro Artale, Diego Calvanese, Roman Kontchakov, and Michael Zakharyashev. The DL-Lite family and relations. *Journal of Artificial Intelligence Research*, 36:1–69, 2009.
- [6] Diego Calvanese, Giuseppe De Giacomo, Domenico Lembo, Maurizio Lenzerini, Riccardo Rosati. Tractable Reasoning and Efficient Query Answering in Description Logics: The DL-Lite Family. *Journal of Automated Reasoning*, 39(3):385–429, 2007.
- [7] Mariano Rodriguez-Muro and Diego Calvanese. High performance query answering over DL-Lite ontologies. In *Proceedings of the 13th International Conference Principles of Knowledge Representation and Reasoning, KR, 2012*.

- [8] Bernardo Cuenca Grau, Evgeny Kharlamov, Dmitriy Zheleznyakov, Marcelo Arenas, and Sarunas Marciuska. On Faceted Search over Knowledge Bases. In Proceedings of the 27th International Workshop on Description Logics (DL). Vienna, Austria, July 2014.
- [9] Γ. Στοϊλος. Εισαγωγή στις Περιγραφικές Λογικές. Σημειώσεις, 2007.
- [10] John A. Robinson and Andrei Voronkov. Handbook of automated reasoning. Elsevier Science Limited, 2001.
- [11] Stuart Russell, Peter Norvig. Artificial Intelligence: A Modern Approach. Prentice Hall, 2009.
- [12] Marcelo Arenas, Bernardo Cuenca Grau, Evgeny Kharlamov, Sarunas Marciuska, Dmitriy Zheleznyakov, Ernesto Jimenez-Ruiz. SemFacet: Semantic Faceted Search over Yago. Proceedings of the companion publication of the 23rd international conference on World wide web companion, April 2014, Seoul, Korea.
- [13] Joel Murach, Andrea Steelman. Murach's Java Servlets and JSP. Mike Murach & Associates, 2008.
- [14] Sébastien Ferré, Alice Hermann, and Mireille Ducassé. Combining Faceted Search and Query Languages. Semantic Search over the Web (SSW) - Advanced Information Systems Engineering Workshops - CAiSE Int. Workshops 83, 2011, 554-563.
- [15] Daniel Tunkelang. Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool Publishers, 2009.