Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

Δηλωτικός προσδιορισμός μεταφραστών Z39.50 με χρήση Περιγραφικής Λογικής

Ιωάννης Ε. Βελεγράκης

Μεταπτυχιακή Εργασία

Ηράκλειο, Ιούνιος 1998

© Copyright 1998 by Ioannis E. Velegrakis
All Rights Reserved

Πανεπιστήμιο Κρήτης
Σχολή Θετικών Επιστημών
Τμήμα Επιστήμης Υπολογιστών

Δηλωτικός προσδιορισμός μεταφραστών Z39.50 με χρήση Περιγραφικής Λογικής

Εργασία που υποβλήθηκε από τον
Ιωάννη Ε. Βελεγράκη
ως μερική εκπλήρωση των απαιτήσεων για την απόκτηση
ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΕΙΔΙΚΕΥΣΗΣ

Συγγραφέας:

Ιωάννης Ε. Βελεγράκης
Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Εισηγητική Επιτροπή:

Πάνος Κωνσταντόπουλος, Καθηγητής, Επόπτης

Γιώργος Τζιρίτας, Αναπληρωτής Καθηγητής, Μέλος

Γιώργος Γεωργακόπουλος, Επίκουρος. Καθηγητής, Μέλος

Βασίλης Χριστοφίδης, Δόκιμος Ερευνητής Ι.Π. ΙΤΕ, Επιβλέπων

Δεκτή:

Πάνος Κωνσταντόπουλος
Πρόεδρος Επιτροπής Μεταπτυχιακών Σπουδών

Ιούνιος 1998

Αφιερώνεται στους γονείς μου
και στον αδερφό μου Νίκο

Δηλωτικός προσδιορισμός μεταφραστών Z39.50 με χρήση Περιγραφικής Λογικής

Ιωάννης Ε. Βελεγράκης

Μεταπτυχιακή Εργασία

Τμήμα Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Περίληψη

Στις μέρες μας, ένας από τους ενεργούς τομείς στην περιοχή της Ανάκτησης Πληροφοριών είναι το πρόβλημα της πρόσβασης, ανάκτησης και ολοκλήρωσης στοιχείων αποθηκευμένων σε ένα μεγάλο αριθμό φυσικά κατανεμημένων ετερογενών πηγών. Προκειμένου να επιτευχθεί κάτι τέτοιο, μια συνηθισμένη προσέγγιση είναι ο ορισμός ενός κοινού μοντέλου δεδομένων. Πάνω από κάθε πηγή υπάρχει μια εφαρμογή η οποία μεταφράζει επερωτήσεις εκφρασμένες στο κοινό αυτό μοντέλο, σε επερωτήσεις στο μοντέλο της υποκείμενης πηγής. Η ίδια εφαρμογή είναι υπεύθυνη για τη μετάφραση των αποτελεσμάτων που επιστρέφει η πηγή, σε όρους του κοινού μοντέλου. Προγράμματα που επιτελούν τέτοιου είδους λειτουργίες αποκαλούνται *μεταφραστές*.

Το Z39.50 είναι ένα πρωτόκολλο για πρόσβαση και ανάκτηση πληροφοριών από ετερογενείς πηγές δεδομένων το οποίο αποτελεί το πρότυπο ANSI/NISO Z39.50 καθώς και το ISO 23950. Ορίζει ένα κοινό μοντέλο αναφοράς το οποίο αποτελείται από μια επίπεδη (χωρίς δομή) λίστα πεδίων τα λεγόμενα *Σημεία Πρόσβασης*, πάνω στα οποία ο χρήστης εκφράζει τις επερωτήσεις του, μέσω μιας γλώσσας λογικών τελεστών.

Η διαδικασία καθορισμού του τρόπου μετάφρασης των αρκετά απλών Z39.50 επερωτήσεων στη γλώσσα επερωτήσεων και τις δομές της υποκείμενης πηγής είναι μια πολύπλοκη διαδικασία, μια και τα δεδομένα της πηγής μπορεί να έχουν σύνθετη δομή. Αντικείμενο της εργασίας αυτής είναι η εύρεση μιας μεθόδου για το δηλωτικό προσδιορισμό του τρόπου μετάφρασης και γενικότερα της λειτουργίας ενός μεταφραστή Z39.50. Η μέθοδος αυτή θα δίνει στο χρήστη τη δυνατότητα να εκφράζει τις αντιστοιχίσεις από το επίπεδο σχήμα του Z39.50 και τους πρωτογενείς τελεστές της γλώσσας επερωτήσεων αυτού, στο δομημένο μοντέλο της πηγής και τη γλώσσα επερωτήσεών της, με ένα εύκολο, αρκετά εκφραστικό και αποδοτικό τρόπο.

Προκειμένου να επιτευχθεί κάτι τέτοιο χρησιμοποιήθηκε η *Περιγραφική Λογική*. Η *Περιγραφική Λογική* παράγει εννοιολογικές γλώσσες οι οποίες είναι αρκετά εκφραστικές και αρκετά κοντά στη φυσική γλώσσα με αποτέλεσμα να είναι εύκολος ο χειρισμός από τον απλό χρήστη. Επιπλέον, η *Περιγραφική Λογική* παρέχει ένα τυπικό πλαίσιο μέσα στο οποίο είναι δυνατή η αντιμετώπιση ενός αριθμού βασικών προβλημάτων του Z39.50.

Επόπτης: Πάνος Κωνσταντόπουλος
Καθηγητής Επιστήμης Υπολογιστών
Πανεπιστήμιο Κρήτης

Declarative Specification of Z39.50 wrappers using Description Logic

Ioannis E. Velegrakis

Master of Science Thesis

Computer Science Department
University of Crete

Abstract

Nowadays, an active topic in the area of Information Retrieval is the problem of accessing, retrieving and integrating information located in a number of physically distributed heterogeneous information sources. In such a case, in order to achieve interoperability, a usual approach is the use of a common reference data model. Over each source there should be an application which translates each query expressed in the common data model to the native language of the underlying data source. The same application is also responsible to translate the results returned from the source to the common model. Such special applications are called *wrappers.*

Z39.50 is a protocol for accessing and retrieving heterogeneous information sources, which is the ANSI/NISO Z39.50 Standard as well as ISO 23950. It presumes a common model which consists of a flat (unstructured) list of fields called *Access Points* which can be used by the user to construct queries. It also specifies a common boolean query language.

The topic of this thesis is the creation of a method for the declarative specification of Z39.50 wrappers. Translating the Z39.50 queries to the native language of the underlying source is a complicated process since the source may be a data base or any other repository with highly structured data. What we are trying to do is to find a method that would give the user the ability to express the mappings from the flat Z39.50 data model and query language primitive operators, to the structured source data model and query language constructors, in an expressive and efficient way.

In order to achieve this, we used *Description Logic. Description Logic* generates concept languages which are very expressive and easy to handle since they are quite close to natural language of the user. Finally, *Description Logic* provides a formal framework for

tackling several issues of Z39.50.

Supervisor: Panos Constantopoulos
Professor of Computer Science
University of Crete

Ευχαριστίες

Στο σημείο αυτό θα ήθελα να ευχαριστήσω θερμά τον επόπτη καθηγητή μου κ. Πάνο Κωνσταντόπουλο που μου εμπιστεύθηκε την εργασία αυτή καθώς και για τη μέχρι τώρα καθοδήγηση και συμβουλές του. Θεωρώ ότι η τετράχρονη συνεργασία μας υπήρξε καθοριστική για τον ερευνητικό μου προσανατολισμό.

Επίσης, ευχαριστώ τα μέλη της εισηγητικής επιτροπής κ. Γιώργο Τζιρίτα και κ. Γιώργο Γεωργακόπουλο για τις εποικοδομητικές τους παρατηρήσεις και διορθώσεις στην εργασία αυτή.

Δεν θα μπορούσα να παραλείψω να πω ένα μεγάλο ευχαριστώ στον επιβλέποντα της εργασίας μου κ. Βασίλη Χριστοφίδη για την άψογη πραγματικά συνεργασία μας, την σωστή του καθοδήγηση και τις γνώσεις, τις οποίες απέκτησα κοντά του τα τελευταία δύο χρόνια.

Θα πρέπει να ευχαριστήσω το Ινστιτουτο Πληροφορικής του Ι.Τ.Ε. και κυρίως την Ομάδα Πληροφοριακών Συστημάτων για την υλικοτεχνική υποστήριξη που μου παρείχε τα τελευταία χρόνια και τις ευκαιρίες που μου έδωσε για συμμετοχή σε διεθνή συνέδρια και συναντήσεις. Ευχαριστώ επίσης το Τμήμα Επιστήμης Υπολογιστών του Π.Κ. για το σύγχρονο περιβάλλον εργασίας που μου παρείχε και τις γνώσεις που μου έδωσε την ευκαιρία να αποκτήσω. Αν και οι σπουδές μου συνεχίζονται σε ένα μεγαλύτερο πανεπιστήμιο μιας μακρινής ηπείρου, τα 7 χρόνια που πέρασα σε αυτό θα μου μείνουν για πάντα αξέχαστα.

Ακόμη, θα ήθελα να ευχαριστήσω την Μαρία Τράκα για τη βοήθεια και συμπαράστασή της στις δύσκολες στιγμές της συγγραφής της εργασίας αυτής.

Το μεγαλύτερο όμως ευχαριστώ απ' όλους το αξίζουν οι γονείς μου. Αν έφτασα σε αυτό το σημείο το χρωστάω σε αυτούς και μόνο. Ακόμα και στην πλουσιότατη ελληνική γλώσσα είναι αδύνατο να βρεθούν λέξεις για να τους εκφράσω τις ευχαριστίες μου για τους κόπους τους, τις αγωνίες τους και όλα όσα θυσίασαν στη ζωή τους για την μόρφωση και πνευματική μου καλλιέργεια. Από την πλευρά μου το ελάχιστο που μπορώ να κάνω είναι να τους αφιερώσω την εργασία αυτή, ελπίζοντας ότι κατάφερα να φανώ αντάξιος των προσδοκιών τους.

Γιάννης Ε. Βελεγράκης

Περιεχόμενα

Περίληψη **i**

Abstract **iii**

Ευχαριστίες **v**

Περιεχόμενα **vii**

Κατάλογος Σχημάτων **xi**

Κατάλογος Πινάκων **xiii**

1 Εισαγωγή **1**

1.1 Το πρόβλημα 1

1.2 Αντικείμενο της εργασίας 3

1.3 Η οργάνωση της εργασίας 4

2 Το πρωτόκολλο Z39.50 **7**

2.1 Υπηρεσίες 8

2.2 Μοντέλο δεδομένων 10

2.3 Γλώσσα επερωτήσεων 10

2.4 Ανάκτηση δεδομένων 11

2.5 Προφίλ 12

2.6 Μελλοντικές επεκτάσεις 13

3 Το πρόβλημα της μετάφρασης **15**

3.1 Ορισμός του προβλήματος 16

3.2 Προβλήματα κατασκευής μεταφραστή 17

3.3 Μεταφραστές Z39.50 σε Ψηφιακές Βιβλιοθήκες και Μουσεία 19

3.3.1 Συστήματα Διαχείρισης Βάσεων Δεδομένων 19
3.3.1.1 Σχεσιακό ΣΔΒΔ 19
3.3.2 Συστήματα Ανάκτησης Πληροφοριών 23
3.3.2.1 Βιβλιογραφική Βάση Δεδομένων 23
3.3.2.2 Index+ 25
3.3.2.3 Mistral 27
3.4 Μεταφραστές σε Συστήματα Ολοκλήρωσης Πληροφορίας 29
3.4.1 Μεταφραστής στηριζόμενος σε υποδείγματα - TSIMMIS 29

4 SIS/Z39.50 Gateway **35**
4.1 Το Σύστημα Σημασιολογικού Ευρετηριασμού 36
4.1.1 Σημαντικά χαρακτηριστικά της γλώσσας Telos 36
4.1.2 Περιγραφή του ΣΣΕ 38
4.2 Αρχιτεκτονική του Z39.50 server 39
4.2.1 Z39.50 server 39
4.2.2 Τμήμα Αναζήτησης 41
4.2.3 Τμήμα Ανάκτησης 44
4.2.4 Z39.50 client 45
4.3 Προβλήματα του Z39.50 46
4.3.1 Προϋπάρχουσα μη αναπαριστάμενη γνώση 46
4.3.2 Συγκεντρωτική θεώρηση 47
4.3.3 Γλώσσα επερωτήσεων 48
4.3.4 Μετα-δεδομένα 50
4.3.5 Μη υποστηριζόμενα Σημεία Πρόσβασης 51
4.3.6 Επιστρεφόμενες εγγραφές 51

5 Περιγραφική Λογική και Z39.50 **53**
5.1 Τα πλεονεκτήματα της Περιγραφικής Λογικής 54
5.2 Η Περιγραφική Λογική 56
5.2.1 Περιγραφή του μοντέλου 56
5.2.2 Το γενικό πλαίσιο της Περιγραφικής Λογικής 57
5.2.2.1 TBox - Εντασιακή (Intensional) γνώση 57
5.2.2.2 ABox - Εκτασιακή (Extensional) γνώση 61
5.2.3 Ερμηνεία 61
5.2.4 Βάση Γνώσης 62

5.2.5 Σχέση Υπαγωγής 62
5.2.6 Υποθέσεις της Περιγραφικής Λογικής 63
5.2.7 Σύνταξη και σημασιολογία των εννοιολογικών γλωσσών 65
5.2.8 Συλλογισμοί της Περιγραφικής Λογικής 69
5.3 Υπολογιστική πολυπλοκότητα 71
5.4 Περιγραφική Λογική και Z39.50 72
5.4.1 Επεξεργασία επερωτήσεων Z39.50 μέσω Περιγραφικής Λογικής . . 72
5.4.2 Πλεονεκτήματα χρήσης της ΠΛ σε Z39.50 servers 76
5.4.2.1 Δόμηση του επιπέδου λεξιλογίου επερωτήσεων 76
5.4.2.2 Έλεγχος καταλληλότητας προφίλ 77
5.4.2.3 Βελτίωση του τρόπου απάντησης επερωτήσεων 78
5.4.2.4 Ιεράρχηση επερωτήσεων 79
5.4.2.5 Ενημέρωση για την ποιότητα των αποτελεσμάτων 82
5.4.2.6 Υποστήριξη πολλαπλών προφίλ 83
5.5 Περιγραφή υλοποίησης Z39.50 server με χρήση ΠΛ 84
5.5.1 Αρχιτεκτονική 84
5.5.2 Επεκτάσεις και σε άλλα συστήματα 89

6 Επίλογος **91**
6.1 Συμπεράσματα 91
6.2 Μελλοντικές κατευθύνσεις 92

A Συντομογραφίες **95**

B Ευρετήριο όρων **97**

Βιβλιογραφία **99**

Κατάλογος Σχημάτων

3.1 Γενική αρχιτεκτονική μεταφραστών και κοινού μοντέλου δεδομένων . . . 16
3.2 Η αρχιτεκτονική του Mistral 27
3.3 Η αρχιτεκτονική του TSIMMIS 30
3.4 Η αρχιτεκτονική του μεταφραστή στο TSIMMIS 32

4.1 Αρχιτεκτονική του Z39.50 server 40
4.2 Μια Z39.50 επερώτηση και το συντακτικό της δέντρο. 42
4.3 Οι σειρά των κλήσεων μεταξύ client, server, και ΣΣΕ 44
4.4 Μια N1NF καθολική σχέση 48

5.1 Αρχικές έννοιες, παράγωγες, και αντικείμενα 58
5.2 Ένα τμήμα μιας βάσης γνώσης κάποιου Μουσείου 68
5.3 Δόμηση ενός συνόλου από ΣΠ, βάσει της σχέσης υπάγεται 77
5.4 Ταξινόμηση των επερωτήσεων στην ΠΛ 81
5.5 TBox με υποστήριξη πολλαπλών προφίλ 83
5.6 Αρχιτεκτονική του Z39.50 server με χρήση ΠΛ 84
5.7 Τα τμήματα της ΒΓ της ΠΛ στην προτεινόμενη αρχιτεκτονική 87
5.8 Αρχείο καθορισμού λειτουργίας μεταφραστή 88

.

Κατάλογος Πινάκων

5.1 Τελεστές σύνθεσης εννοιών και η σημασιολογία τους 65
5.2 Τελεστές σύνθεσης ρόλων και η σημασιολογία τους 66
5.3 Τελεστές σύνθεσης εννοιών και οι αντίστοιχές συναρτήσεις του PQI . . . 85

.

Κεφάλαιο 1

Εισαγωγή

1.1 Το πρόβλημα

Οι πληροφορίες και τα δεδομένα που είναι σήμερα διαθέσιμα προς ανάκτηση, μέσω του Διαδικτύου, βρίσκονται αποθηκευμένα και κατανεμημένα σε ένα μεγάλο αριθμό από διαφορετικά ανεπτυγμένα συστήματα, παρουσιάζοντας μια μεγάλου εύρους ετερογένεια. Οι παράγοντες της ετερογένειας αυτής ποικίλουν. Οι διαφορετικές ανάγκες μεταξύ των χρηστών των διαφόρων εφαρμογών καθώς και οι διαφορετικές όψεις με τις οποίες καθένας βλέπει την πληροφορία, οδηγούν σε διαφορετικούς τρόπους παράστασής της [Mad96]. Επίσης, η ίδια πληροφορία μπορεί πολλές φορές να βρίσκεται σε πολλά συμπληρωματικά τμήματα, κατανεμημένα σε έναν αριθμό ανεξαρτήτων πηγών. Σε αυτά θα πρέπει να προστεθεί και το γεγονός της χρήσης διαφορετικών τεχνολογιών, συστημάτων και πλατφορμών, για την ηλεκτρονική διαχείριση των δεδομένων. Τα παραπάνω έχουν σαν αποτέλεσμα την ύπαρξη διαφοροποιήσεων μεταξύ των πηγών στον τρόπο αποθήκευσης (αρχεία δίσκου, ΣΔΒΔ, ΣΑΠ, ΣΑΓ), στο λειτουργικό σύστημα (UNIX, Windows NT), στο μοντέλο παράστασης (σχεσιακό, οντοκεντρικό), ή ακόμα και στο χρησιμοποιούμενο σύστημα (ORACLE, SYBASE, O2, IRIS). Μέσα σε ένα τέτοιο πλαίσιο οι γλώσσες και οι μέθοδοι προσπέλασης και ανάκτησης είναι πραγματικά κι αυτές με τη σειρά τους ετερογενείς.

Από την άλλη μεριά, με δεδομένο τον ολοένα και μεγαλύτερο όγκο πληροφοριών που γίνεται σήμερα ηλεκτρονικά διαθέσιμος, ο αποτελεσματικός και αποδοτικός εντοπισμός, ανάκτηση και συνδυασμός στοιχείων από γεωγραφικά κατανεμημένες πηγές αποτελεί πραγματικά μια πρόκληση για την ανάπτυξη της απαραίτητης τεχνολογικής υποδομής των λεωφόρων της πληροφορίας. Η τεχνολογία αυτή είναι απαραίτητη για

την αποτελεσματικότερη και αποδοτικότερη ανάκτηση δεδομένων και γνώσεων από το δίκτυο, στήριξη αποφάσεων, και ως ένα βαθμό την αυτοματοποίηση της δράσης οργανισμών και επιχειρήσεων. Γίνεται λοιπόν φανερό, ότι η διαλειτουργικότητα συστημάτων και εφαρμογών δηλ. η ικανότητα να ερμηνεύουμε, να μοιραζόμαστε και να χειριζόμαστε πληροφορίες από ετερογενείς πηγές, αποτελεί βασική ανάγκη. Για να επιτευχθεί κάτι τέτοιο, τα διάφορα συστήματα θα πρέπει να έχουν συμφωνήσει, τουλάχιστον, στο σημασιολογικό περιεχόμενο των ανταλλασσόμενων στοιχείων, ή όπως αλλιώς λέγεται να έχουν επιβεβαιωμένη τη *σημασιολογική διαλειτουργικότητα*, θέμα το οποίο αποτελεί βασικό πεδίο έρευνας στον τομέα της επιστήμης υπολογιστών.

Τα τελευταία χρόνια, ένας μεγάλος αριθμός έργων, έχει κάνει την εμφάνιση του, με σκοπό την αναζήτηση μεθόδων αντιμετώπισης του προβλήματος της σημασιολογικής, συντακτικής και λειτουργικής ετερογένειας. Παράλληλα, για τον ίδιο σκοπό, γίνεται και μια προσπάθεια προτυποποίησης των διαφόρων λειτουργιών πρόσβασης και ανάκτησης δεδομένων. Ένα από τα υπάρχοντα σήμερα, για το σκοπό αυτό, πρωτόκολλα είναι το Z39.50 [ANS95].

Πρόκειται για ένα πρωτόκολλο στηριζόμενο στην αρχιτεκτονική του client/server. Το Z39.50 καθορίζει τα σταθερά μηνύματα που θα πρέπει ανταλλαχθούν μεταξύ εξυπηρετούμενου και εξυπηρέτη, καθώς και ένα αυστηρό συντακτικό για αυτά, επιτρέποντας την προσπέλαση και ανάκτηση οποιουδήποτε είδους δεδομένων. Σύμφωνα με το πρωτόκολλο, ο Z39.50 εξυπηρετούμενος βλέπει τη δομή της πληροφορία της βάσης σαν μια επίπεδη λίστα πεδίων - σημείων πρόσβασης - και διατυπώνει τις επερωτήσεις του πάνω στα πεδία αυτά. Είναι ευθύνη του εξυπηρέτη να μεταφράσει τα σημεία πρόσβασης και τους τελεστές της γλώσσας επερωτήσεων του Z39.50 που χρησιμοποιεί ο εξυπηρετούμενος, στις δομές και τελεστές της γλώσσας της υποκείμενης του εξυπηρέτη πηγής. Με τον τρόπο αυτό λειτουργίας, το Z39.50 είναι δυνατό να αντιμετωπίσει το πρόβλημα της σημασιολογικής ετερογένειας, το πρόβλημα των διαφορετικών μεθόδων πρόσβασης στα ποικίλα αποθηκευτικά συστήματα, καθώς και το πρόβλημα των διαφορετικών γλωσσών επερωτήσεων. Γενικότερα, η προσέγγιση που ακολουθείται στα διάφορα επίπεδα ετερογένειας, είναι η επιλογή τις πιο πρωτογενούς λύσης, της λύσης που είναι περισσότερο κοινή.

1.2 Αντικείμενο της εργασίας

Από τα βασικότερα προβλήματα κατά την εφαρμογή του Z39.50 πάνω από μια πηγή πληροφορίας είναι ο καθορισμός του τρόπου με τον οποίο θα γίνεται η μετάφραση από τη γλώσσα επερωτήσεων του Z39.50 στη γλώσσα επερωτήσεων της πηγής. Κάτι τέτοιο γίνεται αντιστοιχίζοντας κάθε σημείο πρόσβασης (ΣΠ) σε δομές της υποκείμενης πηγής, καθώς και κάθε τελεστή αναζήτησης, σε τελεστές της γλώσσας αναζητήσεων της πηγής. Οι αντιστοιχίσεις αυτές θα πρέπει να ορίζονται όσο το δυνατό πιο εύκολα για το χρήστη, καθώς επίσης δεν θα πρέπει να απαιτούνται χρονοβόρες και επίπονες διαδικασίες σε πιθανές περιπτώσεις αλλαγών τους. Ο εύκολος και αποδοτικός καθορισμός του τρόπου μετάφρασης αποτελεί το αντικείμενο της παρούσας εργασίας.

Πιο συγκεκριμένα, το πρόβλημα το οποίο θα μας απασχολήσει είναι η κατασκευή μιας πληροφοριακής εφαρμογής η οποία θα λειτουργεί ως μεταφραστής (wrapper) για επερωτήσεις και δεδομένα σύμφωνα με το πρωτόκολλο Z39.50 σε βάσεις δεδομένων ή γνώσεων. Ο ρόλος της θα είναι η μετάφραση των επερωτήσεων πάνω στα σημεία πρόσβασης (ΣΠ) που καθορίζονται από ένα προφίλ Z39.50, σε επερωτήσεις πάνω στην πραγματική δομή και δεδομένα της πηγής. Ο καθορισμός των αντιστοιχίσεων στις οποίες στηρίζεται η μετάφραση, γενικά δεν είναι απλή διαδικασία αλλά συνήθως αρκετά πολύπλοκη. Αυτό γιατί η πληροφορία που εκφράζουν τα ΣΠ δεν είναι πάντοτε ρητά παριστάμενη στην πηγή, αλλά μπορεί να αποτελεί τμήμα της. Αιτία αυτού, είναι το γεγονός ότι συνήθως τα διάφορα προφίλ που καθορίζουν τα ΣΠ και τη σημασιολογία τους, δημιουργούνται από οργανισμούς (CIMI [con], AQUARELLE [AQU]) οι οποίοι δεν έχουν πλήρη γνώση του σχήματος και της σημασιολογίας των δεδομένων των πηγών στις οποίες πρόκειται να εφαρμοστεί το συγκεκριμένο προφίλ. Κάτω από ένα τέτοιο γεγονός, το πρόβλημα της αντιστοίχισης μπορεί να θεωρηθεί σαν πρόβλημα ορισμού όψεων (views) πάνω στα δεδομένα της πηγής, μία όψη για κάθε αντιστοίχηση σημείου πρόσβασης.

Ειδικότερα, θα περιγραφεί ένας τρόπος για το δηλωτικό προσδιορισμό (declarative configuration) των όψεων αυτών και κατ' επέκταση του τρόπου λειτουργίας του μεταφραστή. Με τον όρο "δηλωτικό" εννοούμε ότι δεν απαιτείται η συγγραφή προγραμμάτων για κάθε δυνατή αντιστοίχιση ενός προφίλ Z39.50 με την υποκείμενη βάση, και η αναπαραγωγή εκτελέσιμου κώδικα σε περιπτώσεις αλλαγών. Ο χρήστης απλώς θα περιγράφει τις αντιστοιχίσεις με αφαιρετικό τρόπο σε μια υψηλού επιπέδου εννοιολογική γλώσσα (concept language), η οποία είναι κοντά στη φυσική γλώσσα του χρήστη. Τελικός στόχος

είναι η δημιουργία επερωτήσεων, βάσει των αντιστοιχίσεων αυτών και κατά συνέπεια, η αυτόματη μετάφραση των επερωτήσεων Z39.50 σε επερωτήσεις της υποστηριζόμενης από την υποκείμενη πηγή γλώσσας.

Για τον ορισμό των αντιστοιχίσεων χρησιμοποιήθηκε ένα αφηρημένο μοντέλο της πηγής, το οποίο αποτελείται από έννοιες, αντικείμενα, καθώς και μεταξύ τους συσχετίσεις, και το οποίο λειτουργεί σαν ένα κοινό εννοιολογικό μοντέλο που συνδέει τον κόσμο των τελικών χρηστών με τον κόσμο των υποκείμενων ανεξάρτητων πηγών πληροφορίας. Για το χειρισμό ενός τέτοιου είδους μοντέλου η χρησιμοποίηση της τεχνολογίας από τα συστήματα παράστασης γνώσεων, αναδεικνύεται σαν τον επικρατέστερο υποψήφιο. Πιο συγκεκριμένα αποφασίσθηκε η χρήση της Περιγραφικής Λογικής [Bor95] για τον ορισμό εξειδικευμένων οντολογιών και τη συσχέτισή τους με τα εννοιολογικά μοντέλα των πηγών χρησιμοποιώντας ένα κοινό θεωρητικό φορμαλισμό. Κάτι τέτοιο είναι δυνατό στο βαθμό που η εκφραστική δυνατότητα της Περιγραφικής Λογικής (ΠΛ) υπερκαλύπτει γνωστούς φορμαλισμούς που χρησιμοποιούνται σε βάσεις δεδομένων όπως μοντέλα οντοτήτων-συσχετίσεων και κληρονομικότητας κλάσεων. Επίσης, η ΠΛ, εκτός από γλώσσα παράστασης δεδομένων, μπορεί να χρησιμοποιηθεί και σαν μια αρκετά εκφραστική γλώσσα επερωτήσεων όπως παρουσιάζεται στο [Bre95a].

Πρέπει να επισημανθεί ότι ο δηλωτικός προσδιορισμός των διαφόρων αντιστοιχήσεων και της λειτουργίας του μεταφραστή, προσφέρει το πλεονέκτημα της εύκολης και γρήγορης διαχείρισης από το χρήστη, της εύκολης ενσωμάτωσης στο σύστημα καινούριων πηγών πληροφοριών, επιτρέπει τη συστηματική παράσταση των συλλογισμών που γίνονται για την απάντηση μιας επερώτησης, και τέλος κάνει δυνατή την ενημέρωση των χρηστών σχετικά με την ποιότητα της τελικής απάντησης. Παράλληλα με όλα αυτά η χρήση της ΠΛ παρέχει ορισμένα επιπλέον προϊόντα-δυνατότητες που συντελούν στην αντιμετώπιση ενός αριθμού χαρακτηριστικών προβλημάτων του πρωτοκόλλου.

Η εργασία αυτή έγινε στα πλαίσια των προγραμμάτων AQUARELLE [AQU] και CIMI Interoperability Testbed [prob] τα οποία έλαβαν χώρα πρόσφατα, και στα οποία κατασκευάστηκε ένας server-μεταφραστής Z39.50 για μια σημασιολογική βάση δεδομένων με μουσειακά αντικείμενα.

1.3 Η οργάνωση της εργασίας

Στο κεφάλαιο 2 γίνεται μια παρουσίαση του πρωτοκόλλου Z39.50, των βασικών δυνατοτήτων που προσφέρει καθώς και του τρόπου λειτουργίας του.

Στο κεφάλαιο 3 ορίζεται και περιγράφεται αναλυτικά το πρόβλημα της μετάφρασης, ειδικότερα στο Z39.50, το οποίο αποτελεί το βασικό πρόβλημα που η παρούσα εργασία προσπαθεί να επιλύσει. Στη συνέχεια, περιγράφονται διάφορες προσεγγίσεις στο πρόβλημα αυτό από διαφορετικές περιοχές εφαρμογών και με διαφορετικά είδη πηγών πληροφορίας.

Στο κεφάλαιο 4 γίνεται μια αναλυτική περιγραφή της υλοποίησης ενός μεταφραστή Z39.50, η οποία έγινε στα πλαίσια των έργων που προαναφέρθηκαν. Ο μεταφραστής αυτός κατασκευάστηκε ώστε να λειτουργεί πάνω από το Σύστημα Σημασιολογικού Ευρετηριασμού (ΣΣΕ). Η υλοποίηση αυτή έκανε ορατή μια σειρά προβλημάτων του πρωτοκόλλου τα οποία και περιγράφονται.

Στο κεφάλαιο 5 γίνεται μια πρόταση για τη χρησιμοποίησης της Περιγραφικής Λογικής για το δηλωτικό καθορισμό του τρόπου λειτουργίας του μεταφραστή. Αρχικά παρουσιάζεται η Περιγραφική Λογική και στη συνέχεια πώς αυτή μπορεί να εφαρμοστεί στους μεταφραστές Z39.50. Τέλος, παρουσιάζεται μια ομάδα από πλεονεκτήματα και λύσεις που παρέχονται από τη χρήση της αυτή.

Το κεφάλαιο 6 αποτελεί των επίλογο της παρούσας εργασίας και στο οποίο γίνεται μια γενική ανασκόπηση και ορισμένες κρίσεις. Παράλληλα, αναφέρονται και ορισμένες μελλοντικές κατευθύνσεις ή θέματα που θα πρέπει να μελετηθούν.

Κεφάλαιο 2

Το πρωτόκολλο Z39.50

Στην ενότητα αυτή θα γίνει μια εισαγωγή στο πρωτόκολλο Z39.50. Θα ασχοληθούμε με την περιγραφή του βασικού του μοντέλου, της γλώσσας επερωτήσεων που χρησιμοποιεί, τις λειτουργίες του, καθώς και με ορισμένες βασικές έννοιες αυτού. Για αναλυτικές πληροφορίες ο χρήστης μπορεί να επισκεφτεί την επίσημη ιστοσελίδα του πρωτοκόλλου [oC].

Επίσημα το Z39.50 [ANS95] ονομάζεται *American National Standard Institute/National Information Standards Organization Z39.50-1995 Information Retrieval (Z39.50): Application Service Definition and Protocol Specification.* Αποτελεί ένα πρότυπο για αναζήτηση και ανάκτηση πληροφοριών. Προτάθηκε το 1984 με σκοπό τη χρήση του για ανάκτηση βιβλιογραφικών δεδομένων κατανεμημένων στο Διαδίκτυο. Υπάρχουν πολλές εκδόσεις του πρωτοκόλλου με τελευταία εγκεκριμένη την έκδοση 3. Έχει επίσης γίνει αποδεκτό από τον Διεθνή Οργανισμό Τυποποίησης *ISO* σαν το πρότυπο *ISO 23950.*

Αποτελεί πρωτόκολλο για το *επίπεδο εφαρμογών (application layer)* του μοντέλου OSI και καθορίζει τον τρόπο με τον οποίο δύο υπολογιστές μπορούν να επικοινωνήσουν. Σκοπός του είναι να αντιμετωπίσει ορισμένα από τα προβλήματα που συναντώνται κατά την πρόσβαση και ανάκτηση δεδομένων από μια πληθώρα ετερογενών και κατανεμημένων πηγών πληροφοριών. Διευκολύνει την πρόσβαση σε αυτές, ορίζοντας αυστηρά τις συναρτήσεις και τα μήνυμα τα οποία θα πρέπει να ανταλλαχθούν μεταξύ των επικοινωνούντων υπολογιστικών συστημάτων. Ειδικότερα, υποστηρίζει την ανάκτηση δεδομένων μέσα από ένα κατανεμημένο client-server περιβάλλον, όπου ένας υπολογιστής λειτουργεί σαν client στέλνοντας μηνύματα-επερωτήσεις σε έναν άλλο που λειτουργεί σαν server. Ο τελευταίος αφού επεξεργαστεί την επερώτηση επιστρέφει στον client τα αποτελέσματά της. Το λογισμικό στον server επιτρέπει την εκτέλεση της επερώτησης σε

περισσότερες από μία βάσεις ταυτόχρονα, και επιστρέφει τον πληθάριθμο του σύνολο των εγγραφών που βρέθηκαν να ικανοποιούν τα προσδιοριζόμενα από την επερώτηση κριτήρια. Κατόπιν, ο client μπορεί να ζητήσει επιλεκτικά ανάκτηση εγγραφών.

Η ισχύς του Z39.50 έγκειται στο διαχωρισμό της διεπαφής χρήσης στη μεριά του client, από τις βάσεις δεδομένων, ή τις μηχανές αναζήτησης που διαχειρίζεται ο server. Σύμφωνα με τις προδιαγραφές του πρωτοκόλλου όλοι οι clients βλέπουν με τον ίδιο ομοιόμορφο τρόπο την πληροφορία, κάτι που επιτρέπει την ολοκλήρωση ετερογενών δεδομένων (heterogeneous information integration) πάνω από μια μεγάλη ποικιλία διαφορετικών πηγών πληροφοριών. [Moe95].

Το πρωτόκολλο είναι *προσανατολισμένο στις συνδέσεις (session oriented)* και *τη διατήρηση καταστάσεων (statefull)*. Το πρώτο σημαίνει ότι για την επίτευξη επικοινωνίας μεταξύ client και server, θα πρέπει να ανοιχτεί κάποιο κανάλι σύνδεσης μεταξύ τους το οποίο θα παραμείνει ανοικτό για όσο διάστημα διαρκεί η επικοινωνία αυτή. Το δεύτερο σημαίνει ότι ενδιάμεσες καταστάσεις της επικοινωνίας διατηρούνται και μπορούν να χρησιμοποιηθούν σε μεταγενέστερες χρονικές στιγμές. Για παράδειγμα, ο client μπορεί να στείλει μια επερώτηση στον server η οποία να χρησιμοποιεί ένα σύνολο απαντήσεων προερχόμενο από προγενέστερη επερώτηση. Παρόλο που το Z39.50 δημιουργήθηκε αρχικά για να ικανοποιήσει τις ανάγκες της κοινωνίας των βιβλιογραφικών πληροφοριών, οι τελευταίες εκδόσεις μπορούν να χρησιμοποιηθούν για αναζήτηση και ανάκτηση οποιουδήποτε είδους δεδομένων. Όμως, ο προσανατολισμένος προς τις βιβλιογραφικές πηγές πληροφορίας σχεδιασμός του, δημιουργεί μια σειρά προβλημάτων τα οποία και θα περιγραφούν σε επόμενη ενότητα.

Στην ενότητα που ακολουθεί περιγράφονται συνοπτικά οι βασικότερες υπηρεσίες του Z39.50. Στη συνέχεια, παρουσιάζεται το μοντέλο δεδομένων του, καθώς και η γλώσσα επερωτήσεων την οποία χρησιμοποιεί. Κατόπιν, γίνεται μια αναφορά στον τρόπο ανάκτησης των δεδομένων. Τέλος, πριν γίνει αναφορά στις μελλοντικές κατευθύνσεις του πρωτοκόλλου, περιγράφεται η βασική στο Z39.50 έννοια του προφίλ.

2.1 Υπηρεσίες

Οι δυνατότητες του Z39.50 καθώς και ο τρόπος λειτουργίας του, καθορίζονται από τις υπηρεσίες τις οποίες προσφέρει. Για κάθε υπηρεσία, ορίζεται ένα σύνολο μηνυμάτων τα οποία θα πρέπει να ανταλλαχθούν μεταξύ client και server κατά τη χρήση της υπηρεσίας.

Κατά την έναρξη μιας επικοινωνίας χρησιμοποιείται η υπηρεσία *Initialization*. Μόλις γίνει η σύνδεση των δύο υπολογιστικών συστημάτων, ο client στέλνει μια *Init request* και ο server απαντάει με μια *Init response*. Κατά την ανταλλαγή αυτή τα δύο συστήματα αποφασίζουν τις διάφορες παραμέτρους της επικοινωνίας, όπως η έκδοση του πρωτοκόλλου που θα χρησιμοποιηθεί, το μέγεθος των πακέτων που θα ανταλλάσσουν και άλλα. Επίσης, γίνεται πιστοποίηση της ταυτότητας του χρήστη για να ελεγχθεί η ύπαρξη ή μη δικαιώματος πρόσβασης στα δεδομένα του server.

Η *Search* αποτελεί τη βασική υπηρεσία του Z39.50. Ο client στέλνει ένα μήνυμα *Search request* στο οποία ζητάει την ανάκτηση πληροφοριών βάση κάποιων προδιαγραφών που θέτει. Αυτό γίνεται μέσω μιας επερώτησης η οποία περιέχεται στο μήνυμα αυτό. Ο server θα πρέπει να μεταφράσει την επερώτηση αυτή στη γλώσσα επερωτήσεων της (ή των) υποκείμενων πηγών. Μετά την εκτέλεσή της, στέλνεται πίσω στον client ένα μήνυμα *Search Response* με τον πληθάριθμο των εγγραφών των πηγών που βρέθηκαν να ικανοποιούν τις προδιαγραφές που όριζε η επερώτηση. Επίσης, στο μήνυμα περιέχεται και ένα όνομα το οποίο χαρακτηρίζει το σύνολο απαντήσεων. Ο client μπορεί να χρησιμοποιήσει το όνομα αυτό για μελλοντικές αναφορές στο σύνολο αυτό, σε νέες επερωτήσεις (λόγο της ιδιότητας των διατηρητέων καταστάσεων του πρωτοκόλλου), ή να ζητήσει ρητώς ανάκτηση ορισμένων στοιχείων από το σύνολο αυτό. Το τελευταίο γίνεται στέλνοντας μια *Present Request* της *Present* υπηρεσίας, η οποία προσδιορίζει τις προς ανάκτηση εγγραφές. Ο server απαντάει στέλνοντας τα δεδομένα αυτά μέσα σε ένα *Present Response* μήνυμα.

Τέλος, η υπηρεσία *Close* καθορίζει τα μηνύματα *Close Request* και *Close Response*, τα οποία στέλνονται προκειμένου να λάβει τέλος μια Z39.50 σύνδεση μεταξύ client και servre.

Σε περίπτωση που συμβεί κάποιο λάθος και η διαδικασία αρχικοποίησης, επερώτησης, ή ανάκτησης δεδομένων δεν μπορέσει να ολοκληρωθεί, τότε τα μηνύματα *Init*, *Search*, *Present* και *Close Request*, μπορούν να περιέχουν τον κωδικό του λάθους που έλαβε χώρα.

Τέλος, στο Z39.50 ορίζονται και ορισμένες άλλες υπηρεσίες όπως για παράδειγμα η *Scan* ή η *Sort*, οι οποίες όμως δεν θα μας απασχολήσουν στην παρούσα εργασία.

2.2 Μοντέλο δεδομένων

Προκειμένου να επιτευχθεί διαλειτουργικότητα, θα πρέπει τόσο οι servers όσο και οι clients να έχουν ένα κοινό μοντέλο αναφοράς, ένα κοινό μοντέλο δεδομένων. Στο Z39.50 το μοντέλο αυτό είναι μια επίπεδη λίστα πεδίων στην οποία ο client μπορεί να διατυπώνει τις επερωτήσεις του. Η σημασιολογία των πεδίων καθορίζεται από το εκάστοτε προφίλ (βλέπε 2.5), δεν καθορίζεται όμως κάποιος συγκεκριμένος τύπος, δομή, ούτε συσχετίσεις μεταξύ τους. Έτσι επερωτήσεις εκφρασμένες σ' αυτά, είναι σαν επερωτήσεις εκφρασμένες πάνω στο σχήμα μιας *καθολικής σχέσης (universal relation)* [Ull82].

Τα πεδία αυτά, στην ορολογία του Z39.50, αποκαλούνται *Use Attributes.* Αντιστοιχούν στα *Σημεία Πρόσβασης (access points)* των βάσεων δεδομένων, (με συντομία ΣΠ). Αυτός θα είναι ο όρος που θα χρησιμοποιείται όταν αναφερόμαστε σε αυτά από εδώ και στο εξής. Η λίστα των ΣΠ δεν είναι στατική. Μπορεί να εμπλουτιστεί με νέα ή να γίνει παράληψη ορισμένων, προκειμένου η λίστα να είναι κατάλληλη για εφαρμογή του πρωτοκόλλου σε νέων κοινοτήτων εφαρμογές. Το πρώτο σύνολο ΣΠ που ορίστηκε, αναφέρονταν σε βιβλιογραφικά δεδομένα και αποκαλείται ως *"BIB-1 Use Attribute Set"*. Μεταγενέστερα, δημιουργήθηκαν κι άλλα, όπως για παράδειγμα το *"CIMI Use Attribute Set"* που απευθύνεται στη μουσειακή και γενικότερα στην πολιτισμική κοινότητα.

2.3 Γλώσσα επερωτήσεων

Το πρωτόκολλο ορίζει ένα αριθμό από τύπους επερωτήσεων που μπορούν να χρησιμοποιηθούν. Οι βασικοί είναι δύο: ο **type-0** και ο **type-1**. Ο **type-0** χρησιμοποιείται προκειμένου να αναπαραστήσει μια γλώσσα επερωτήσεων η οποία δεν καθορίζεται από το πρωτόκολλο και προέρχεται από τοπική συμφωνία μεταξύ των δύο επικοινωνούντων υπολογιστικών συστημάτων. Κάτι τέτοιο όμως δεν συντελεί στην επίτευξη διαλειτουργικότητας. Ο τύπος **type-1** είναι ο βασικότερος και κάθε Z39.50 server ή client είναι υποχρεωμένος να τον αναγνωρίζει. Αποτελεί δε, τη χαρακτηριστική γλώσσα επερωτήσεων του Z39.50.

Ο **type-1** ιστορικά αποκαλείται και *Reverse Polish Notation (RPN).* Αναπαριστά λογικές επερωτήσεις, έχοντας μια σύνταξη στην οποία πρώτα αναφέρεται ο τελεστής και έπειτα ακολουθούν οι τελεσταίοι του. Η σύνταξη αυτή δεν δημιουργεί ασάφειες ως προς τη σειρά εκτέλεσης, και έτσι δεν απαιτείται η χρήση παρενθέσεων. Οι RPN επερωτήσεις μπορούν εύκολα να αναπαρασταθούν σαν δυαδικά δένδρα, με τους τελεστές

σαν εσωτερικούς κόμβους, και τους τελεσταίους σαν φύλλα ή υποδένδρα. Για να είναι περισσότερο κατανοητό για τον αναγνώστη, οι επερωτήσεις που θα παρουσιάζονται από εδώ και στο εξής, δεν θα είναι στη RPN μορφή τους.

Μια στοιχειώδης επερώτηση αποτελείται από έναν όρο αναζήτησης και από μια λίστα προσδιορισμών, οι οποίοι αποκαλούνται *προσδιοριστικές ιδιότητες (attributes).* Ένας όρος μπορεί να έχει πολλές προσδιοριστικές ιδιότητες. Περισσότερο πολύπλοκες επερωτήσεις μπορούν να γίνουν με τη χρήση των λογικών τελεστών *AND, OR* και *AND_NOT*. Μια αναφορά στις δυνατότητες και στα προβλήματα μιας τέτοιας γλώσσας γίνεται στο [CGMP96]. Έτσι, για παράδειγμα, η επερώτηση που αναζητάει τα μουσειακά αντικείμενα, τα οποία συσχετίζονται με τον ήρωα της ελληνικής επανάστασης *Οδυσσέα Ανδρούτσο* και χρονολογούνται στο έτος 1821, εκφράζεται ως εξής:

PersonalName="Androutsos" ***AND*** **Date**="1821"

όπου τα PersonalName και Date είναι σημεία πρόσβασης.

Οι προσδιοριστικές ιδιότητες δίνουν, στη γλώσσα επερωτήσεων, μεγαλύτερη εκφραστικότητα. Μέχρι στιγμής έχουν οριστεί 6 τύποι προσδιοριστικών ιδιοτήτων. Ο πρώτος τύπος θεωρείται ότι είναι τα ΣΠ. Υπάρχουν, επίσης, οι τύποι *Relation, Truncation, Position, Structure* και *Completeness.* Τα *Relation* περιγράφουν τη σχέση που θα πρέπει να έχουν τα ανακτόμενα δεδομένα με τον αναζητούμενο όρο. Έτσι, για παράδειγμα, η επερώτηση:

Date="1810" ***AND*** **Relation**="GreaterThan"

αναζητά τα αντικείμενα που συσχετίζονται με ημερομηνίες μεταγενέστερες του έτους 1810. Τα *Truncation* χρησιμοποιούνται για αναζητήσεις με βάση τμήματα αλφαριθμητικών, οπότε, η επερώτηση:

PersonalName="Andr" ***AND*** **Truncation**="Right"

αναζητά τα αντικείμενα που συσχετίζονται με ονόματα που αρχίζουν με το γράμμα "Andr" (όσα δηλαδή έχουν την μορφή "Andr∗"). Τα *Position, Structure* και *Completeness attributes* δε θα μας απασχολήσουν στην εργασία αυτή.

2.4 Ανάκτηση δεδομένων

Η απάντηση σε μια επερώτηση του χρήστη είναι ένα σύνολο δεδομένων. Το σύνολο αυτό έχει κάποιο όνομα που το χαρακτηρίζει και κάνει δυνατή τη χρησιμοποίησή του

σε μεταγενέστερες επερωτήσεις. Το σύνολο των δεδομένων είναι μια λίστα, της οποίας τα στοιχεία αναγνωρίζονται από τη θέση τους μέσα σε αυτή, δίνοντας την δυνατότητα στο χρήστη να αναφέρεται σε αυτά που τον ενδιαφέρουν να ανακτήσει.

Τα δεδομένα που επιστρέφονται από τον server στον client, θα πρέπει να έχουν μια συγκεκριμένη κωδικοποίηση. Οι δυνατές κωδικοποιήσεις ορίζονται από το πρωτόκολλο. Μια τέτοια είναι, για παράδειγμα, η *US-MARC*, η οποία αποτελεί μια λίστα δεδομένων χωρίς φωλιάσματα, παριστάνει δηλαδή εγγραφές *πρώτης κανονικής μορφής (1NF)*. Η κωδικοποίηση αυτή χρησιμοποιείται κυρίως για βιβλιογραφικά δεδομένα, δεν είναι όμως κατάλληλη για μη επίπεδες εγγραφές (εγγραφές σε N1NF). Για τέτοιες περιπτώσεις χρησιμοποιείται μια δεύτερη κωδικοποίηση που ορίζεται από το Z39.50, η *Generic Record Syntax 1* ή πιο σύντομα *GRS-1*. Η κωδικοποίηση αυτή είναι αρκετά εκφραστική και μπορεί να παραστήσει δομημένες εγγραφές, όπως αριθμούς, αλφαριθμητικά, εικόνες, SGML κείμενα, καθώς και σύνθετους τύπους. Λόγω της μεγάλης εκφραστικότητάς της, αποτελεί την κατεξοχήν χρησιμοποιούμενη κωδικοποίηση σε εφαρμογές του Z39.50. Τέλος, μια άλλη, συχνά συναντώμενη, είναι η *Simple Unstructured Text Record (SUTR)*, η οποία κωδικοποιεί ένα και μόνο αλφαριθμητικό. Για το λόγο αυτό έχει περιορισμένες δυνατότητες αλλά πολύ απλή και εύκολη υλοποίηση.

Ανεξάρτητα από την κωδικοποίηση που θα ακολουθηθεί για τις εγγραφές, η δομή τους, δηλαδή τα πεδία (elements) που περιέχουν και ο τρόπος παράταξής τους θα πρέπει να είναι επίσης αυστηρά καθορισμένος. Προκειμένου να καλυφθούν διαφορετικές ανάγκες χρηστών, συνήθως ορίζονται δύο ή τρία διαφορετικά είδη *ανακτόμενων εγγραφών (retrieved records)*, με διαφορετικά πεδία και δομή καθεμία, από τις οποίες ο client μπορεί να επιλέξει σε ποια θέλει να είναι τα ανακτώμενα από τον server δεδομένα.

2.5 Προφίλ

Για τη σωστή εφαρμογή του πρωτοκόλλου είναι απαραίτητος ο ορισμός κάποιου *προφίλ (profile)*. Το *προφίλ* χρησιμοποιείται για να καθορίσει τον τρόπο συμπεριφοράς του πρωτοκόλλου σε μια συγκεκριμένη λειτουργία ή γενικότερα σε μια εφαρμογή. Με τον όρο συμπεριφορά νοείται η επιλογή των τιμών διαφόρων παραμέτρων, των υπηρεσιών καθώς και των διαφόρων χαρακτηριστικών που θα υποστηρίζονται, όταν κάτι τέτοιο δεν καθορίζεται αυστηρά από το πρωτόκολλο. Έτσι, σε ένα προφίλ μπορεί να καθορίζονται τα Σημεία Πρόσβασης, οι υποστηριζόμενες υπηρεσίες, το μέγεθος των ανταλλασσόμενων πακέτων, και άλλα. Επίσης, κάθε προφίλ καθορίζει και εξυπηρετεί τις ανάγκες κάποιας

συγκεκριμένης κοινότητας.

Μέχρι σήμερα έχει οριστεί ένας ικανοποιητικός αριθμός από προφίλ του Z39.50 για διάφορες εφαρμογές [oC]. Το πρώτο προφίλ που ορίστηκε ήταν το **Bib-1 profile** [ANS95], και αναφερόταν σε βιβλιογραφικές πηγές δεδομένων. Η μετέπειτα ανάγκη για χρήση του πρωτοκόλλου και σε άλλες εφαρμογές, οδήγησε στη δημιουργία νέων όπως το **CIMI profile** [proa], το οποίο απευθύνεται στην κοινότητα των μουσείων και το **AQUARELLE profile** [Ltd97] το οποίο τώρα αναπτύσσεται. Υπάρχουν και άλλα προφίλ όπως το **GILS**, το **WAIS**, το **ATS-1** κλπ. Αναλυτική αναφορά στα προφίλ αυτά ο αναγνώστης μπορεί να βρει στον επίσημο κόμβο του Z39.50 [oC]. Κατά τον ορισμό κάθε νέου προφίλ, γίνεται προσπάθεια χρησιμοποίησης στοιχείων που έχουν ήδη οριστεί και σε άλλα, με σκοπό την όσο το δυνατό μικρότερη διαφορά μεταξύ των διαφορετικών προφίλ. Απώτερος σκοπός είναι ένας επεξεργαστής να μπορεί να δεχτεί και να απαντήσει επερωτήσεις από όσο το δυνατό περισσότερους clients που υποστηρίζουν διαφορετικά προφίλ, πράγμα που σημαίνει μεγαλύτερη διαλειτουργικότητα.

2.6 Μελλοντικές επεκτάσεις

Παρόλο που η γλώσσα επερωτήσεων του Z39.50 μπορεί να χρησιμοποιηθεί πάνω από πολλών ειδών πηγές δεδομένων, η εκφραστικότητά της είναι αρκετά περιορισμένη σε σχέση με αυτή των σχεσιακών η οντοκεντρικών γλωσσών των βάσεων δεδομένων [FC]. Η εκφραστικότητα της γλώσσας SQL είναι αρκετά μεγάλη και γίνεται ακόμα μεγαλύτερη στην SQL3 [DD95]. Έτσι, προτάθηκε ο εμπλουτισμός του πρωτοκόλλου με δυνατότητες επερωτήσεων SQL, και τη δημιουργία της λεγόμενης Z39.50/SQL+ έκδοσης [BF98].

Την Z39.50/SQL+ μπορεί να τη δει κανείς σαν μια επέκταση της έκδοσης 3 του Z39.50 που υπάρχει σήμερα, καταφέρνοντας να ολοκληρώσει τις δυνατότητες της εκφραστικής γλώσσας SQL με τις δυνατότητες του Z39.50. Μια τέτοια επέκταση είναι πολύ αποδοτική (αλλά όχι περιοριστική) σε πηγές πληροφορίας οι οποίες διαχειρίζονται SQL βάσεις δεδομένων. Παρέχεται έτσι η δυνατότητα στο χρήστη να έχει ένα διατηρητέων καταστάσεων πρωτόκολλο, και μια ισχυρή γλώσσα SQL όταν θα συνδέεται σε απομακρυσμένες και κυρίως ετερογενείς βάσεις δεδομένων.

Για να επιτευχθεί κάτι τέτοιο στο [BF98] προτείνεται η δημιουργία ενός καινούριου τύπου επερωτήσεων, όπως ο **type-1** που θα αποκαλείται **type-SQL** και θα είναι σύμφωνος με το πρότυπο SQL3. Απόδειξη της λειτουργικότητας της ιδέας της χρήσης SQL, είναι το έργο ZINC [FC] στο πανεπιστήμιο του Queensland της Αυστραλίας. Η ιδέα αυτή

αποτελεί μια πολύ σοβαρή πρόταση και η ενσωμάτωση της στην επόμενη έκδοση του πρωτοκόλλου θεωρείται πολύ πιθανή.

Κεφάλαιο 3

Το πρόβλημα της μετάφρασης

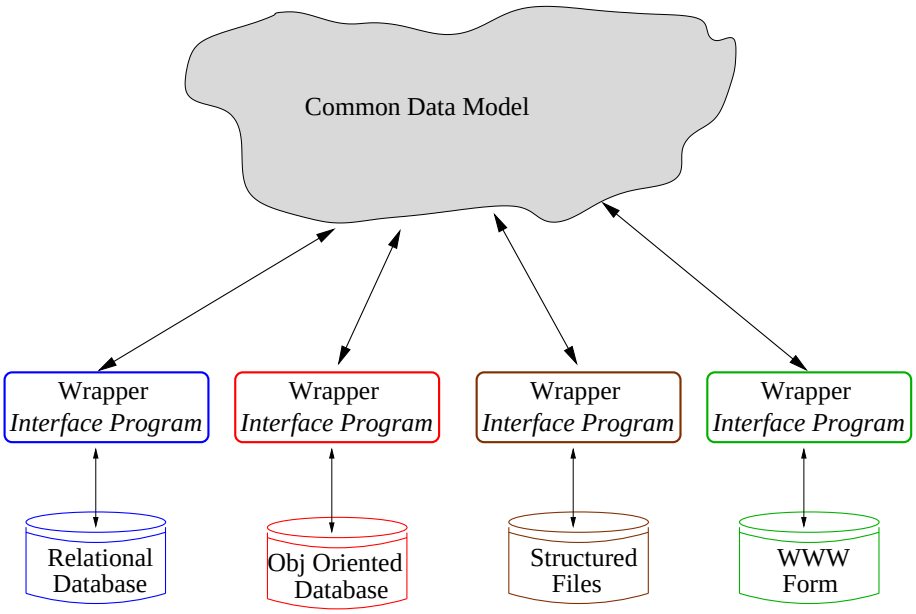
Στο κεφάλαιο αυτό αναλύεται το πρόβλημα της μετάφρασης (wrapping) το οποίο αποτελεί και το κύριο θέμα της εργασίας αυτής. Το πρόβλημα αυτό περιλαμβάνει τη μετάφραση επερωτήσεων εκφρασμένων πάνω σε κάποιο μοντέλο, σε επερωτήσεις πάνω σε κάποιο άλλο. Μια τέτοια μετατροπή γίνεται με μετάφραση των δομικών στοιχείων του ενός μοντέλου, στα δομικά στοιχεία του άλλου, καθώς και των τελεστών της γλώσσας επερωτήσεων του πρώτου, σε τελεστές της γλώσσας επερωτήσεων του δευτέρου. Τα αποτελέσματα των επερωτήσεων θα πρέπει κι αυτά με τη σειρά τους να μεταφραστούν από το ένα μοντέλο στο άλλο.

Στην ενότητα που ακολουθεί ορίζεται το πρόβλημα της μετάφρασης ειδικά για την περίπτωση του Z39.50. Στη συνέχεια περιγράφονται τα βασικότερα προβλήματα της κατασκευής ενός Z39.50 server τα οποία κάνουν φανερή την ανάγκη ύπαρξης κάποιας αυτοματοποιημένης μεθόδου για τον καθορισμό του τρόπου λειτουργίας του μεταφραστή. Τέλος, γίνεται μια παρουσίαση του τρόπου υλοποίησης και λειτουργίας μεταφραστών κατασκευασμένων σε διάφορα έργα. Ο τρόπος με τον οποίο γίνεται η μετάφραση είναι στενά συνδεδεμένος με την μέθοδο η οποία ακολουθείται για την ολοκλήρωση και διαχείριση της ετερογενούς και κατανεμημένης πληροφορίας, καθώς επίσης και με το είδος της υποκείμενης πηγής. Έτσι θα παρουσιαστούν περιπτώσεις όπου η υποκείμενη πηγή μπορεί να είναι Σύστημα Διαχείρισης Βάσεων Δεδομένων (Database Management System), ή Σύστημα Ανάκτησης Πληροφοριών (Information Retrieval System). Επίσης, το σύστημα μπορεί να είναι μια Ψηφιακή Βιβλιοθήκη (Digital Library) ή ένα Σύστημα Ολοκλήρωσης Πληροφοριών (Information Integration System).

3.1 Ορισμός του προβλήματος

Στην πληθώρα των πηγών πληροφοριών που βρίσκονται σήμερα συνδεδεμένες στο δίκτυο, συναντά κανείς πολλές διαφοροποιήσεις ως προς το είδος των δεδομένων (εικόνες, κείμενα), το μέσο αποθήκευσης (ΣΔΒΔ, αρχεία δίσκου), το μοντέλο παράστασης στον υπολογιστή (σχεσιακό, οντοκεντρικό), ή ακόμα και τον τρόπο με τον οποίο επιτυγχάνεται πρόσβαση στις πηγές αυτές. Για την επίτευξη της διαλειτουργικότητας και την αποτροπή του χρήστη από την ανάγκη γνώσης των διαφοροποιήσεων αυτών, μια λύση που συχνά χρησιμοποιείται [GMHI+95], είναι ο ορισμός ενός κοινού μοντέλου με το οποίο ο χρήστης συναλλάσσεται, ορίζοντας σε αυτό τις επερωτήσεις του και λαμβάνοντας, επίσης σε αυτό, τις απαντήσεις.

Κάτω από μια τέτοια προσέγγιση απαιτείται, η δυνατότητα μετατροπής από το ένα μοντέλο στο άλλο, και πιο συγκεκριμένα από το γενικό μοντέλο στο μοντέλο δεδομένων της κάθε πηγής και τανάπαλιν. Μια τέτοια διαδικασία υλοποιείται από ειδικά προγράμματα-εφαρμογές τα οποία αποκαλούνται **μεταφραστές (wrappers)**. Οι μεταφραστές τοποθετούνται πάνω από κάθε πηγή πληροφορίας όπως φαίνεται στο σχήμα 3.1.



Σχήμα 3.1: Γενική αρχιτεκτονική μεταφραστών και κοινού μοντέλου δεδομένων

Σύμφωνα με την προσέγγιση του Z39.50, τόσο το κοινό μοντέλο όσο και η γλώσσα επερωτήσεων αυτού είναι καθορισμένες από το πρωτόκολλο και από το αντίστοιχο κάθε

φορά προφίλ. Κατά την εισαγωγή μιας νέας πηγής, αυτό που χρειάζεται να γίνει είναι η κατασκευή του δικού της μεταφραστή ο οποίος λειτουργεί πάνω από αυτή, και κάνει τις μετατροπές ανάμεσα στο μοντέλο της και στο μοντέλο του Z39.50. Αναλυτικά, οι λειτουργίες που πρέπει να επιτελεί ένας τέτοιος Z39.50 μεταφραστής είναι οι ακόλουθες:

- Μετάφραση του μοντέλου του Z39.50, στις δομές και τη σημασιολογία της υποκείμενης πηγής. Πιο συγκεκριμένα, τα ΣΠ του προφίλ, θα πρέπει να μεταφράζονται σε σχέσεις, ρόλους, κλάσεις, κλπ.

- Μετάφραση των επερωτήσεων **type-1** που φθάνουν στο Z39.50 server, στα δομικά στοιχεία της γλώσσας επερωτήσεων της πηγής. Έτσι ανάλογα με το είδος της πηγής τα λογικά φίλτρα της γλώσσας **type-1** θα μετατρέπονται σε επερωτήσεις SQL, OQL, πλήρους κειμένου (full text queries), κλπ.

- Η τρίτη λειτουργία που πρέπει να υλοποιεί είναι η μετατροπή των αποτελεσμάτων των επερωτήσεων στις δομές, δηλαδή στις εγγραφές, που καθορίζονται από το προφίλ. Τα δεδομένα, από την αρχική τους μορφή, όποια κι αν είναι αυτή, αφού μετατραπούν στις κατάλληλες εγγραφές (records), πρέπει να κωδικοποιηθούν με βάση κάποια από τις προκαθορισμένες κωδικοποιήσεις (GRS-1, USMARC, SUTRS, κλπ.).

3.2 Προβλήματα κατασκευής μεταφραστή

Η δημιουργία και ο καθορισμός της λειτουργίας ενός μεταφραστή είναι γενικά μια χρονοβόρα και με αρκετές δυσκολίες διαδικασία. Βασική δυσκολία αποτελεί η μεγάλη ποικιλομορφία των πηγών δεδομένων που μπορούν να υπάρξουν. Η υποκείμενη του μεταφραστή πηγή μπορεί να είναι:

- Σύστημα Ανάκτησης Πληροφοριών (IRS),

- Σύστημα Διαχείρισης Βάσεως Δεδομένων (DBMS),

- Σύστημα Βάσης Γνώσης (KBS),

- Βάση εικόνων, κ.α.

Άλλη δυσκολία είναι το ότι το Z39.50 καθορίζει μονάχα τη σημασιολογία των ΣΠ και όχι τον τύπο και τη μορφή των δεδομένων [ANS95]. Έτσι ο μεταφραστής πρέπει

να υλοποιεί πολλές φορές εκτός των άλλων και μετατροπές τύπων όπου αυτό κρίνεται απαραίτητο για την ύπαρξη διαλειτουργικότητας.

Βασικότερη και δυσκολότερη διαδικασία είναι ο καθορισμός των αντιστοιχίσεων μεταξύ των ΣΠ και των πεδίων της πηγής. Από τη μια μεριά είναι ο σταθερός κόσμος του Z39.50 με τα προκαθορισμένα ΣΠ και από την άλλη ο προκαθορισμένος επίσης κόσμος της πηγής. Αυτό που καλείται να κάνει ο κατασκευαστής του μεταφραστή είναι η εύρεση αντιστοιχίσεων μεταξύ δομών των δύο αυτών κόσμων. Με δεδομένο ότι μόνο το σημασιολογικό και όχι το συντακτικό περιεχόμενο των ΣΠ ορίζεται από το πρωτόκολλο, οι αντιστοιχίσεις αυτές θα είναι σημασιολογικές. Η εύρεση αυτών μπορεί να είναι αρκετά απλή, πολύπλοκη ή αδύνατη. Αυτό γιατί η σημασιολογική πληροφορία κάποιου ΣΠ μπορεί να παριστάνεται ρητά, οπότε η αντιστοίχιση είναι τετριμμένη, κρυμμένη, οπότε απαιτείται ο ορισμός κάποιας όψης (view) μέσω της γλώσσας επερωτήσεων της πηγής, ή ανύπαρκτη, οπότε το ΣΠ δεν μπορεί να υποστηριχθεί. Οι οργανισμοί που καθορίζουν τα προφίλ, φροντίζουν ώστε τα ΣΠ να είναι αρκετά και πλούσια σε εννοιολογικό περιεχόμενο, ώστε να ικανοποιούν όσο το δυνατό καλύτερα τις ανάγκες της κοινότητας στην οποία το προφίλ απευθύνεται. Ωστόσο, επειδή είναι δύσκολο να προβλεφθούν όλες οι δυνατές περιπτώσεις πληροφοριών, που μπορεί να έχει μια πηγή μιας συγκεκριμένης κοινότητας, θα υπάρχουν πάντα περιπτώσεις μη υποστηριζόμενων ΣΠ.

Άλλο πρόβλημα είναι η απαίτηση από το χρήστη που ορίζει τις αντιστοιχίσεις, της πολύ καλής γνώσης τόσο των ΣΠ και της σημασιολογίας τους, όσο και του εννοιολογικού σχήματος και της γλώσσας επερωτήσεων της πηγής. Το πρόβλημα γίνεται ακόμα μεγαλύτερο με το μεγάλο αριθμό των διαφορετικών τεχνολογιών των πηγών.

Τελειώνοντας, θα πρέπει να σημειωθεί ότι δεν έχει βρεθεί κάποιος αυτόματος τρόπος για τον καθορισμό της μετάφρασης. Συνήθως, η όλη διαδικασία γίνεται από το χρήστη με το χέρι και οι αντιστοιχίσεις δηλώνονται ρητώς από αυτόν. Σύμφωνα με το [AK97], τρόποι έστω και ημιαυτοματοποίησης της διαδικασίας είναι εξαιρετικά χρήσιμοι γιατί:

- Οι πηγές πληροφορίας με ενδιαφέροντα στοιχεία είναι πάρα πολλές και ετερογενείς, ακόμα και μέσα σε μια μόνο συγκεκριμένη κοινότητα.
- Συνεχώς εισάγονται νέες πηγές δεδομένων με διαφορετικά μοντέλα, σχήματα και μεθόδους πρόσβασης η κάθε μία.
- Η μορφή, η δομή και τα περιεχόμενα των πηγών συχνά μεταβάλλεται.
- Η χρήση των μεταφραστών γνωρίζει μεγάλη εξάπλωση σήμερα.

Γίνεται λοιπόν φανερή η χρησιμότητα μεθόδων που τείνουν προς την κατεύθυνση της αυτοματοποίησης της διαδικασίας αυτής.

3.3 Μεταφραστές Z39.50 σε Ψηφιακές Βιβλιοθήκες και Μουσεία

3.3.1 Συστήματα Διαχείρισης Βάσεων Δεδομένων

3.3.1.1 Σχεσιακό ΣΔΒΔ

Μια εφαρμογή του Z39.50 πάνω από σχεσιακές βάσεις δεδομένων (SYBASE, ORACLE) έχει γίνει από το ινστιτούτο CNR-CNUCE στα πλαίσια του έργου AQUARELLE. Στο [SL97] γίνεται μια αναλυτική περιγραφή του τρόπου υλοποίησης. Η περίπτωση των σχεσιακών βάσεων είναι αρκετά διαφοροετική από τις περιπτώσεις των ΣΑΠ πάνω στα οποία στηρίζεται η φιλοσοφία του Z39.50. Έτσι θα πρέπει να γίνουν οι απαραίτητες ενέργειες ώστε να προσδοθούν στη βάση δεδομένων, δυνατότητες αντίστοιχες με αυτές των ΣΑΠ που απαιτούνται από το πρωτόκολλο.

Μια διαφορά είναι ότι στις βάσεις δεδομένων δεν υπάρχει η έννοια του συνόλου αποτελεσμάτων (result set) την οποία ο χρήστης μπορεί να αναφέρει και να χρησιμοποιήσει σε μεταγενέστερες επερωτήσεις. Επίσης, η ανάκτηση και η παράσταση είναι δύο διαφορετικές λειτουργίες στα ΣΑΠ, αλλά στις βάσεις δεδομένων ο χρήστης μαζί με την επερώτηση καθορίζει τα περιεχόμενα και τη μορφή που θα έχουν τα ανακτώμενα δεδομένα. Η βασική διαφορά είναι ότι στα ΣΑΠ η πληροφορία έχει μια "επίπεδη" δομή σε αντίθεση με την δομημένη οργάνωση των ΣΔΒΔ που απαιτείται για την επίτευξη της 3NF. Επίσης, στα ΣΑΠ, η βασική αποθηκευμένη πληροφορία είναι "το κείμενο", ή γενικά ένα συγκεκριμένο είδος αντικειμένου, το οποίο ανακτάται και στο οποίο όλες οι επερωτήσεις αναφέρονται. Για παράδειγμα η επερώτηση Date=1998 σε μια βιβλιογραφική πηγή δεδομένων, θα αναφέρεται στα κείμενα με ημερομηνία 1998. Σε αντίθεση, στις βάσεις δεδομένων, μπορεί να ζητηθεί η ανάκτηση οποιουδήποτε στοιχείου περιέχεται στη βάση, (π.χ. σε βιβλιογραφική βάση θα μπορούσαν να ζητηθούν πρόσωπα, ημερομηνίες, κλπ. και όχι μόνο κείμενα) που ίσως για την ανάκτησή του να απαιτούνται ενώσεις δύο ή περισσοτέρων πινάκων. Τέλος, η αντιστοίχιση στα ΣΔΒΔ δεν είναι πλέον ένα συντακτικό πρόβλημα, αλλά ένα πιο δύσκολο εννοιολογικό πρόβλημα. Πιο συγκεκριμένα, για να γίνει σωστή αντιστοίχιση, θα πρέπει αρχικά να βρεθεί στη βάση η ακριβής σημασιολογία των ΣΠ, κάτι που ίσως να απαιτεί πολλές και πολύπλοκες επερωτήσεις.

Οποιαδήποτε υλοποίηση αντιστοίχισης ΣΠ και δομών της βάσης, θα πρέπει να

στηρίζεται στην αρχή ότι το ήδη υπάρχον σχήμα δεδομένων δεν θα πρέπει να μεταβληθεί. Δύο είναι οι δυνατές προσεγγίσεις. Η πρώτη είναι η χρησιμοποίηση βοηθητικών πινάκων οι οποίοι θα περιέχουν πληροφορίες για τα πρωταρχικά και ξένα κλειδιά των πινάκων και άλλα μέτα-δεδομένα τα οποία θα καθοδηγούν τη δημιουργία των απαραίτητων ενώσεων. Η δεύτερη είναι ο ορισμός όψεων μέσω SQL. Για την υλοποίηση της πρώτης θα μπορούσε να γραφεί μια ειδική εφαρμογή η οποία να διαβάζει τις πληροφορίες αυτές από τους πίνακες και να υλοποιούσε τις απαραίτητες συνθέσεις. Σύμφωνα με τους συγγραφείς του [SL97], ο δεύτερος τρόπος είναι προτιμότερος μια και βοηθάει περισσότερο στη διατήρηση της συνέπειας της βάσης σε περιπτώσεις αλλαγών.

Οι αντιστοιχίσεις των διαφόρων ΣΠ είναι αρκετά πολύπλοκες και αρκετά διαφορετικές μεταξύ τους, κάτι που καθιστά δύσκολη έως αδύνατη την υλοποίησή τους με απλές επιλογές πινάκων και προβολών. Κάτι τέτοιο κάνει αναγκαίο τον ορισμό μιας διαφορετικής σχεσιακής όψης για κάθε ΣΠ.

Για την αποθήκευση των αντιστοιχίσεων των ΣΠ με όψεις της βάσης, δημιουργήθηκε ένας βοηθητικός πίνακας ο οποίος κρατάει την έκφραση SQL που ορίζει την όψη. Ο ορισμός του πίνακα αυτού φαίνεται παρακάτω. Κάθε πλειάδα αυτού αντιστοιχεί στον ορισμό μίας αντιστοίχισης-όψης.

```
create table aqlmap (
        UseId          int           not null,
        Pdb            varchar(48)   not null,
        Ldb            varchar(48)   not null,
        cName          varchar(48)   not null,
        cType          int           not null,
        fromJoinExpr   varchar(254)  null,
        whereJoinExpr  varchar(254)  null
        )
```

Από τον παραπάνω πίνακα τα σημαντικότερα από άποψη ενδιαφέροντος πεδία είναι:

- UseId: Το ΣΠ στο οποίο η όψη αντιστοιχεί.
- Pdb: Το όνομα της βάσης στην οποία ανήκουν οι πίνακες που συνθέτουν την όψη. Αυτό είναι απαραίτητο γιατί υπάρχει περίπτωση ένας Z39.50 server να διαχειρίζεται δύο ή περισσότερες βάσεις.
- cType: Πρόκειται για ένα πεδίο το οποίο χρησιμοποιείται για συντακτικές μετα-

τροπές. Καθορίζει με άλλα λόγια αν η ερωτούμενη στην Z39.50 επερώτηση τιμή του όρου θα πρέπει να συμμετάσχει στην SQL επερώτηση όπως είναι, ή αν θα πρέπει να γίνει σε αυτή κάποιος μετασχηματισμός. Για παράδειγμα η επερώτηση Date=”1998” κατά τη μετατροπή της σε SQL θα πρέπει να της αφαιρεθούν τα quotes σύμβολα αν το πεδίο στη βάση που αναπαριστά την ημερομηνία είναι τύπου integer.

- FromJoinExpr: Αποτελεί την έκφραση που θα υπάρχει στο *from* τμήμα της SQL επερώτησης ορισμού της όψης.

- WhereJoinExpr: Όμοια, θα αποτελεί την έκφραση του *where* τμήματος της επερώτησης ορισμού της όψης.

Μια επερώτηση πάνω σε κάποιο ΣΠ, μεταφράζεται στην SQL επερώτηση:

```
select distinct ID
from <search view>
where <condition I>
   order by ID
```

όπου η `<search view>` είναι η όψη για το συγκεκριμένο ΣΠ η οποία ορίζεται από κάποια εγγραφή του πίνακα aqlmap που μόλις περιγράφηκε, και `<condition I>` κάποιοι περιορισμοί οι οποίοι βασίζονται στις τιμές που έχει θέσει ο χρήστης στην Z39.50 επερώτηση.

Δεδομένου του ότι στις βάσεις δεδομένων δεν υπάρχει η έννοια του συνόλου απαντήσεων, για να διατηρηθούν τα αποτελέσματα, ορίζεται ένας πίνακας, ως εξής:

```
create table aqlrs (
            rsID    varchar(255)  not null,
            pID     int           not null,
            Zdb     char(40)      not null,
            docPos  int           not null,
            docID   varchar(48)   not null
            )
```

Στον πίνακα αυτό το rsID είναι το όνομα του συνόλου απαντήσεων, docPos είναι η θέση της συγκεκριμένης πλειάδας στο σύνολο απαντήσεων, και docID είναι ο μοναδικός κωδικός αναγνώρισης του δεδομένου που αναπαριστά η πλειάδα.

Για την ανάκτηση κάποιας συγκεκριμένης εγγραφής από το σύνολο απαντήσεων, χρησιμοποιείται ο πίνακας `aqlrs` προκειμένου να βρεθεί ο μοναδικός κωδικός της εγγραφής. Έπειτα, γίνεται χρήση μιας συνάρτησης η οποία λαμβάνει σαν όρισμα τον μοναδικό κωδικό της εγγραφής καθώς επίσης και το είδος της εγγραφής που πρέπει να επιστραφεί στον client (full, brief, κλπ.). Η συνάρτηση αυτή είναι υπεύθυνη να ανακτήσει όλα τα απαιτούμενα πεδία από τη βάση και να τα δομήσει με τον τρόπο που αναφέρεται στο προφίλ, να τα κωδικοποιήσει σε GRS-1, και να τα επιστρέψει, ώστε να σταλούν στον client.

Η προσέγγιση αυτή είναι αρκετά καλή. Ένα βασικό στοιχείο είναι ότι οι απόψεις βρίσκονται αποθηκευμένες μέσα στο σύστημα, το οποίο και τις διαχειρίζεται. Έτσι, μειώνονται σημαντικά οι υποχρεώσεις του χρήστη για τη σωστή διαχείριση και συντήρηση αυτών. Ένα άλλο μεγάλο πλεονέκτημα είναι ότι γίνεται πλήρης εκμετάλλευση της γλώσσας επερωτήσεων SQL, η οποία είναι αρκετά εκφραστική, καθώς και του σχήματος της βάσης. Μπορεί δηλαδή κανείς να χρησιμοποιήσει τη σημασιολογική πληροφορία που υπάρχει στον τρόπο δόμησης των δεδομένων της βάσης. Η αντιστοίχιση ΣΠ και δομών της βάσης δεν είναι μονάχα συντακτική αλλά και σημασιολογική. Αν η σημασιολογία κάποιου ΣΠ βρίσκεται αποθηκευμένη στη βάση, με τη χρήση της εκφραστικής γλώσσας επερωτήσεων, μπορεί να ανακτηθεί και να χρησιμοποιηθεί. Έστω, για παράδειγμα, ότι στη βάση υπάρχει ο πίνακας Date με πεδία την ημερομηνία (date) και τον τύπο της (type). Το ΣΠ DateOfPublication μπορεί να αντιστοιχισθεί στις δομές της βάσης αν οριστεί η όψη `Date_of_Publication` ως εξής:

```
create view Date_of_Publication as
{
        select *
        from Date
        where Date. type="Publication"
}
```

Μια όψη στην ουσία δεν είναι τίποτε άλλο παρά μια απάντηση σε μια επερώτηση. Όταν υπάρχουν πολλές απόψεις είναι φυσικό να υπάρχουν και διάφορες μεταξύ τους συσχετίσεις. Για παράδειγμα δύο απόψεις μπορεί να είναι εντελώς ασυσχέτιστες ή η μία να εμπεριέχει την άλλη. Κάτι τέτοιο συμβαίνει και με τα ΣΠ. Για παράδειγμα το ΣΠ DateOfCreation εμπεριέχεται στο ΣΠ Date. Προκειμένου να επιτελέσει κανείς λειτουργίες οι οποίες αφορούν τη διαχείριση μεγάλου αριθμού απόψεων, θα ήταν καλό ένα σύστημα να έχει τη δυνατότητα να κάνει συλλογισμούς και να εξάγει συμπεράσματα για τις

συσχετίσεις μεταξύ των απόψεων ή μεταξύ απόψεων και επερωτήσεων [LR96].

Το πρόβλημα όμως που υπάρχει με τον ορισμό και τη χρήση απόψεων για τις απεικονίσεις, είναι ότι ο χρήστης είναι αναγκασμένος να γνωρίζει την γλώσσα επερωτήσεων SQL η οποία είναι αρκετά πολύπλοκη. Επίσης απαιτείται πολύ καλή γνώση λεπτομερειών του σχήματος της βάσης, όπως για παράδειγμα τα *ξένα κλειδιά (foreign keys)* κλπ.

3.3.2 Συστήματα Ανάκτησης Πληροφοριών

3.3.2.1 Βιβλιογραφική Βάση Δεδομένων

Ένας χαρακτηριστικός Z39.50 server για βιβλιογραφικές βάσεις δεδομένων είναι αυτός που κατασκευάστηκε στο πανεπιστήμιο του Tubingen στη Γερμανία [LBG97]. Ο server αυτός λειτουργεί σαν ενδιάμεσο επίπεδο ανάμεσα σε μια βιβλιογραφική βάση δεδομένων και το Διαδίκτυο αποτελώντας παράλληλα και την πόρτα εισόδου (gateway) για τη βάση αυτή. Για την επικοινωνία με τη βάση, η εφαρμογή χρησιμοποιεί το πρωτόκολλο telnet και για την επικοινωνία με τους άλλους clients το Z39.50.

Πρόσβαση στα δεδομένα παρέχεται μέσω μιας γλώσσας επερωτήσεων, η οποία χρησιμοποιεί τους τελεστές ”and”, ”or”, και ”not”. Επίσης, παρέχεται μια σειρά από πεδία τα οποία μπορούν να χρησιμοποιηθούν για την κατασκευή των επερωτήσεων, όπως ”Τίτλος”, ”Συγγραφέας”, κλπ. Η σύνταξη των επερωτήσεων που δέχεται η βάση είναι της μορφής:

s **AUthor**="John Von Neumann" ***and*** **TItle**="The computer and the brain"

Η πρόσβαση στη βάση γίνεται μέσω του πρωτοκόλλου Telnet.

Η μετάφραση από τη γλώσσα του Z39.50 στη γλώσσα της βάσης, καθορίζεται από κάποιο αρχείο του οποίου τα περιεχόμενα αποκαλούνται ”πίνακες μετάφρασης”. Ένα τμήμα του αρχείου αυτού φαίνεται παρακάτω.

```
[Use]
4 $Field:="TI"
20 $Field:="CC"
31 $Field:="PD"
55 $Field:="CY"
1003 $Field:="AU"
```

```
[Relation]
1 $Res:=format("%s<%s", $Term, $Field);
2 $Res:=format("%s<=%s", $Term, $Field);

[Truncation]
1 $Term:=format("%s?", $Term);

[Boolean]
1  $Res:=format("(%s and %s)", $op1, $op2);
2  $Res:=format("(%s or %s)", $op1, $op2);
3  $Res:=format("(%s not %s)", $op1, $op2);

$
```

Για κάθε λογικό τελεστή του Z39.50 υπάρχει αντίστοιχος στη γλώσσα επερωτήσεων της βάσης όπως μπορεί να φανεί από το τμήμα `[Boolean]` του αρχείου. Η μόνη αλλαγή που απαιτείται να γίνει είναι στον τρόπο σύνταξης. Για τις ιδιότητες *διάταξης* και *αποκοπής*, όπως δείχνουν τα αντίστοιχα τμήματα του πίνακα μετάφρασης, υπάρχουν, και για αυτές, αντίστοιχες συντακτικές εκφράσεις στη γλώσσα επερωτήσεων.

Τέλος, όσον αφορά τα ΣΠ, από το τμήμα `[Use]` παραπάνω, μπορεί κανείς να δει ότι κάθε ΣΠ αντιστοιχίζεται σε ένα και μόνο πεδίο της βάσης κάθε φορά. Για παράδειγμα, το ΣΠ 4 (title) αντιστοιχίζεται στο πεδίο TI.

Από τα παραπάνω γίνεται φανερό ότι η γλώσσα επερωτήσεων που παρέχει η βάση είναι ανάλογη με αυτή του Z39.50. Για το λόγο αυτό η μετατροπή των επερωτήσεων από το ένα μοντέλο στο άλλο, είναι λίγο πολύ τετριμμένη. Η αντιστοίχιση από τα ΣΠ στα πεδία της βάσης είναι απλή και ένα προς ένα. Αν για κάποιο ΣΠ υπάρχει αντίστοιχο πεδίο στη βάση, τότε αντιστοιχίζεται σε αυτό. Αν όμως δεν υπάρχει κάποιο που να εκφράζει ακριβώς την ίδια σημασιολογία με το συγκεκριμένο ΣΠ, τότε δεν γίνεται αντιστοίχιση. Αποκλείεται με τον τρόπο αυτό να γίνουν περισσότερο πολύπλοκες (όχι ένα προς ένα) αντιστοιχίσεις. Για παράδειγμα, μια επερώτηση πάνω στο ΣΠ Personal Name θα έπρεπε να μεταφραστεί σε μια επερώτηση τόσο στο πεδίο AUthor της βάσης, όσο και στο πεδίο ReViewer[1]. Κάτι τέτοιο όμως με την παρούσα μέθοδο αντιστοίχησης δεν είναι εφικτό. Η όλη σημασιολογική πληροφορία που προέρχεται από τη δόμηση των δεδομένων στη

[1] Η σημασιολογία των πεδίων αυτών είναι η προφανής από το όνομά τους

βάση δεν μπορεί να χρησιμοποιηθεί. Η αντιστοίχηση που γίνεται με την μέθοδο αυτή είναι καθαρά συντακτική.

Ένα πλεονέκτημα της μεθόδου αυτής είναι η χρήση αρχείων παραμέτρων για τον καθορισμό των αντιστοιχήσεων. Επιτρέπει στο χρήστη την εύκολη αλλαγή τους χωρίς να είναι απαραίτητη η επαναδημιουργία του δυαδικού εκτελέσιμου αρχείου. Επίσης, ο τρόπος δήλωσης των αντιστοιχήσεων είναι κατανοητός και αρκετά εύχρηστος για τον διαχειριστή του συστήματος.

3.3.2.2 Index+

Η System Simulation Ltd. στη Βρετανία έχει κατασκευάσει στα πλαίσια του έργου AQUARELLE ένα Z39.50 server πάνω από το ΣΑΠ Index+ [SSL97]. Όσο αφορά το Index+, πρόκειται για ένα πρόγραμμα καθώς και ένα σύνολο από εργαλεία λογισμικού για τη δημιουργία, ανάπτυξη και υποστήριξη βάσεων δεδομένων οι οποίες περιέχουν κείμενα, υλικό πολυμέσων, καθώς και αριθμητικά δεδομένα. Μπορεί να λειτουργήσει καλά και σε περιβάλλοντα στα οποία υπάρχει μίξη λειτουργικού UNIX με άλλα λειτουργικά των PC. Στηρίζεται στην αρχιτεκτονική client/server. Οι λειτουργίες αναζήτησης που υποστηρίζονται είναι λογικοί τελεστές, κανονικές εκφράσεις, καθώς και αριθμητικά και αλφαβητικά διαστήματα (ranges).

Η λειτουργία του Index+ στηρίζεται πάνω στις ίδιες αρχές της θεωρίας Ανάκτησης Πληροφοριών (Information Retrieval) με το Z39.50. Αυτό έχει σαν αποτέλεσμα την ύπαρξη μιας άμεσης σχέσης-αντιστοιχίας μεταξύ των εννοιών και λειτουργιών αυτού, με τις έννοιες και λειτουργίες του Z39.50. Για κάθε μια από τις λειτουργίες *Init*, *Search*, *Present* και *Close* του Z39.50 υπάρχουν παρόμοιες λειτουργίες στο Index+. Μία επερώτηση στο Index+ εκφράζεται με ένα *δένδρο αναζήτησης*. Το *δένδρο αναζήτησης* αυτό, έχει την ίδια δομή με το συντακτικό δένδρο αναπαράστασης των Z39.50 επερωτήσεων.

Το Index+ ως ΣΑΠ έχει μια λίστα από πεδία πάνω στα οποία μπορεί να γίνει αναζήτηση. Τα ΣΠ αντιστοιχούνται στα πεδία αυτά μέσω ενός αρχείου παραμέτρων. Κατά την εκτέλεση μιας επερώτησης, για την μετάφραση κάποιου ΣΠ, αναζητείται στο αρχείο αυτό ο κωδικός του και βρίσκεται το αντίστοιχο πεδίο του Index+. Ένα τμήμα από ένα τέτοιο αρχείο είναι το εξής:

```
type 1=use
   val 1=au  #Personal Name
   val 2=cn  #Corporate Name
   val 4=ti  #title
```

Στο τμήμα αυτό ορίζεται, για παράδειγμα, ότι το ΣΠ με κωδικό 1 (το ”PersonalName”) αντιστοιχεί στο πεδίο ”au” του Index+.

Για τις περιπτώσεις των ιδιοτήτων *διάταξης* ή *αποκοπής*, το ίδιο αρχείο καθορίζει τον τρόπο μετάφρασης στη σύνταξη του Index+ με χρήση του τελεστή *expr*. Έτσι, για παράδειγμα, το τμήμα του αρχείου:

```
type 5=truncation
    val 1=expr(this+"*")
```

δηλώνει ότι όταν για κάποιο ΣΠ ζητηθεί η ιδιότητα αποκοπής (κωδικός 5) με τιμή right (κωδικός 1), η τιμή αναζήτησης της του ΣΠ, θα πρέπει να δοθεί στο αντίστοιχο πεδίο του Index+ με έναν αστερίσκο στο τέλος. Η επερώτηση επομένως

PersonalName="Andr" ***AND*** **Truncation**="Right"

θα μεταφραστεί στην έκφραση au=Andr* στο Index+.

Η ανάκτηση και μετάφραση των δεδομένων είναι γενικά απλή διαδικασία. Μια εγγραφή που επιστρέφεται από το Index+ είναι στη μορφή που πρέπει για να σταλεί στον client που τη ζήτησε. Το μόνο που απαιτείται είναι η κωδικοποίηση της σε GRS-1. Τα στοιχεία του Index+ που θα αποτελέσουν την εγγραφή περιγράφονται σε ένα δεύτερο αρχείο. Μια τέτοια περιγραφή είναι η:

```
rectype object
    element title G 1
    element authorOrCreator G 2
    element description G 17
```

Το παραπάνω ορίζει την εγγραφή *object* η οποία αποτελείται από 3 πεδία, τα: *title*, *authorOrCreator* και *description*. Οι κωδικοί μετά τα ονόματα των πεδίων χρησιμοποιούνται για την GRS-1 κωδικοποίηση.

Όπως μπορεί να φανεί από τα παραπάνω, η όλη μετάφραση από το Z39.50 στις δομές και τελεστές του Index+ και αντιστρόφως, είναι μια άμεση και τετριμμένη διαδικασία. Αυτό γιατί οι δομές, τελεστές και τρόπος λειτουργίας τους είναι ανάλογος, επιτρέποντας απλές ”ένα προς ένα” αντιστοιχίσεις. Οι αντιστοιχίσεις αυτές είναι μονάχα συντακτικές. Το μόνο που χρειάζεται είναι να υπάρχει για κάθε ΣΠ ένα πεδίο του Index+ με την ίδια σημασιολογία. Αν κάτι τέτοιο δεν υπάρχει, η αντιστοίχηση δεν μπορεί να γίνει. Επίσης, δεν μπορεί να γίνει σύνθεση πεδίων. Για παράδειγμα, έστω ότι υπάρχουν τα πεδία Owner

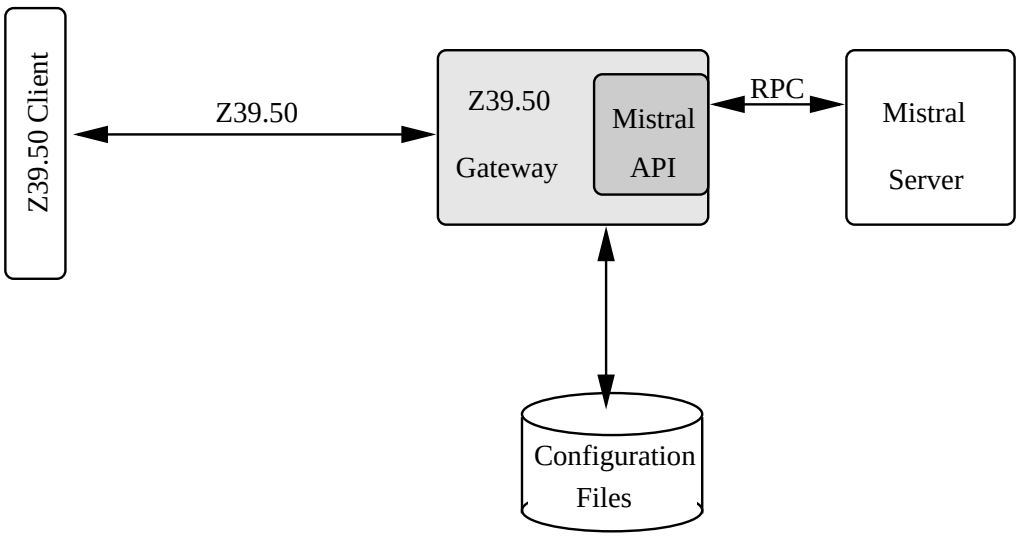
και Actor. Το ΣΠ Person θα έπρεπε να αντιστοιχηθεί και στα δύο αυτά πεδία, κάτι που όμως δεν γίνεται.

Άξιο λόγου είναι το ότι ο καθορισμός των αντιστοιχήσεων γίνεται με χρήση αρχείων τα οποία είναι αρκετά κατανοητά από τον απλό χρήστη.

3.3.2.3 Mistral

Η εταιρία BULL στη Γαλλία έχει επίσης κατασκευάσει ένα Z39.50 server πάνω από το ΣΑΠ Mistral [BDH97]. Το Mistral χρησιμοποιείται για αποθήκευση και ανάκτηση κειμένων τα οποία επιτρέπουν αναζήτηση με πολλά πεδία, χρησιμοποιώντας λέξεις κλειδιά, και ευρετηριασμό πλήρους κειμένου (full text indexing). Οι αναζητήσεις γίνονται με χρήση λογικών τελεστών, συγκρίσεων, και κανονικών εκφράσεων. Επίσης, κατά τις αναζητήσεις μπορούν να χρησιμοποιηθούν θησαυροί όρων, για μεγαλύτερη απόδοση.

Η αρχιτεκτονική του server φαίνεται στο σχήμα 3.2. Ο μεταφραστής-server αποτελείται από το τμήμα που δέχεται τα Z39.50 μηνύματα, και είναι χτισμένο με YAZ [Ind97], και από το τμήμα το οποίο επικοινωνεί με το Mistral με χρήση απομακρυσμένων κλήσεων συναρτήσεων (remote procedure calls). Επίσης, υπάρχει και ένα σύνολο αρχείων που καθορίζει τον τρόπο λειτουργίας του μεταφραστή.



Σχήμα 3.2: Η αρχιτεκτονική του Mistral

Η γλώσσα επερωτήσεων του Mistral είναι μια λογική γλώσσα, η οποία υποστηρίζει τους τελεστές *AND, OR* και *XCPT* (αντίστοιχος του AND_NOT). Η μετάφραση επομένως

των τριών τελεστών της **type-1** γλώσσας επερωτήσεων σε τελεστές της γλώσσας επερωτήσεων του Mistral, είναι άμεση δεδομένης της ένα προς ένα μεταξύ τους αντιστοίχησης.

Στο Mistral υποστηρίζονται, επίσης, κανονικές εκφράσεις μέσω των συμβόλων *+* (οσοιδήποτε χαρακτήρες), *?* (ένας ή κανένας χαρακτήρας), και *#* (ένας χαρακτήρας). Οι δυνατότητες *αποκοπής* του Z39.50 υπερκαλύπτονται από τους παραπάνω τελεστές, επομένως ούτε αυτών η μετάφραση παρουσιάζει κάποια πολυπλοκότητα.

Η αντιστοίχηση των ΣΠ με τα πεδία επερωτήσεων του Mistral καθορίζονται κι εδώ με χρήση ειδικών αρχείων παραμέτρων (configuration files). Οι αντιστοιχίσεις αυτές μπορεί να είναι "ένα προς ένα" ή "πολλά προς ένα". Η "ένα προς πολλά" αντιστοίχηση, δεν υποστηρίζεται, δηλαδή, σε κάθε ΣΠ μπορεί να αντιστοιχηθεί ένα και μόνο πεδίο του Mistral. Οι δηλώσεις στο αρχείο παραμέτρων έχουν την εξής μορφή:

```
Fld: 1, AUTR
Fld: 12, REF
```

Οι δύο αυτές γραμμές δηλώνουν ότι τα ΣΠ του Z39.50 με κωδικούς 1 και 12 αντιστοιχούνται στα πεδία AUTR και REF του Mistral.

Μετά από μια αναζήτηση, το Mistral επιστρέφει ένα σύνολο αντικειμένων τα οποία βρέθηκαν να ικανοποιούν τους περιορισμούς που έθετε η επερώτηση. Το Mistral μπορεί να κρατήσει το τελευταίο σύνολο αποτελεσμάτων για περαιτέρω επεξεργασία. Οι ανακτόμενες εγγραφές κωδικοποιούνται σε GRS-1 μορφή. Για τον προσδιορισμό των πεδίων του Mistral που αντιστοιχούν στα στοιχεία των εγγραφών, που ορίζονται από το προφίλ, υπάρχουν στο αρχείο παραμέτρων δηλώσεις της μορφής:

```
Full: AUTR, element.author
Full: DENA, element.title
```

Αυτό το τμήμα του αρχείου ορίζει ότι στην εγγραφή με το όνομα Full υπάρχουν τα πεδία *author* και *title* τα οποία προέρχονται από τα πεδία *AUTR* και *DENO* του Mistral αντίστοιχα.

Το Mistral όπως και το Index+ ακολουθούν ένα μοντέλο ανάκτησης πληροφορίας παρόμοιο με αυτό του Z39.50. Για το λόγο αυτό η μετάφραση των τελεστών της γλώσσας του Z39.50 στη δική τους γλώσσα επερωτήσεων είναι απλή και άμεση διαδικασία. Οι επερωτήσεις γίνονται πάνω σε μια λίστα πεδίων, στηριζόμενες σε αναζητήσεις με βάση λέξεις κλειδιά, χωρίς καμία περαιτέρω δομή. Ακόμα όμως και για τις λέξεις κλειδιά, είναι πιθανή η ύπαρξη κάποιου είδους δομή όπως για παράδειγμα όταν αυτές προέρχονται από ένα θησαυρό όρων. Το σχήμα που ακολουθεί το Mistral έχει τα ίδια μειονεκτήματα

που αναφέρθηκαν και στο Index+, ίσως όμως να είναι λίγο πιο εκφραστικό μια και υποστηρίζει επιπλέον την "πολλά προς ένα" αντιστοίχηση.

3.4 Μεταφραστές σε Συστήματα Ολοκλήρωσης Πληροφορίας

3.4.1 Μεταφραστής στηριζόμενος σε υποδείγματα - TSIMMIS

Το **TSIMMIS** [GMHI+95] αποτελεί ένα από τα βασικότερα έργα στον τομέα των Συστημάτων Ολοκλήρωσης Πληροφορίας (Information Integration Systems ή IIS) και κατασκευάστηκε στο Πανεπιστήμιο Stanford της Καλιφόρνιας. Αποσκοπεί στην ολοκλήρωση ενός μεγάλου εύρους πληροφοριών δομημένων (βάσεις δεδομένων), ημιδομημένων (κείμενα SGML), ή αδόμητων (αρχεία στο δίσκο), οι οποίες δεν έχουν κάποιο συγκεκριμένο σχήμα.

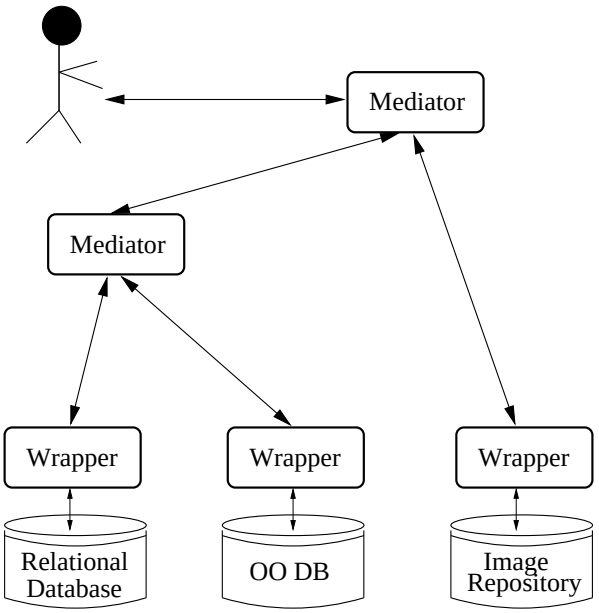
Για μια τέτοιου είδους ολοκλήρωση, απαιτείται η ύπαρξη κάποιου κοινού μοντέλου αναφοράς. Για το TSIMMIS αναπτύχθηκε ένα τέτοιο μοντέλο δεδομένων, το *Object Exchange Model (OEM)* [GMPQ+97]. Βασικό στοιχείο του TSIMMIS είναι ότι δεν υπάρχει μια κεντρική βάση δεδομένων η οποία να κρατάει κάποιο κοινό σχήμα μια και κάτι τέτοιο δεν υπάρχει. Το όλο σύστημα αποτελείται από διαμεσολαβητές και μεταφραστές. Ο διαμεσολαβητής (mediator) είναι ένα αυτόνομο πρόγραμμα το οποίο έχει γνώση της απαραίτητης πληροφορίας σχετικά με το είδος των δεδομένων που υπάρχει στις πηγές, καθώς και τον τρόπο με τον οποίο μπορεί, αν χρειαστεί, να γίνει συγχώνευση και μορφοποίηση των αποτελεσμάτων επερωτήσεων από δύο ή περισσότερες πηγές πριν αυτά αποσταλούν στο χρήστη. Οι διαμεσολαβητές επικοινωνούν μεταξύ τους, αλλά και με τους μεταφραστές, με μια κοινή γλώσσα, την OEM-QL, η οποία στηρίζεται στο OEM. Μια γενική όψη του τρόπου με τον οποίο διασυνδέονται μεταφραστές, διαμεσολαβητές και πηγές παρουσιάζεται στο σχήμα 3.3.

Οι επερωτήσεις στο TSIMMIS αποτελούν μια περιγραφή, μέσω της OEM-QL και του OEM, η οποία προσδιορίζει το σχήμα των προς ανάκτηση δεδομένων. Για παράδειγμα, στο [PGMGU95] αναφέρεται η επερώτηση:

```
*P :- <P person {<L last_name 'Van Gogh'>}
```

η οποία αναζητά όλα τα πρόσωπα εκείνα τα οποία έχουν όνομα "Van Gogh".

Για την υλοποίηση μεταφραστών έχει αναπτυχθεί μια μέθοδος καθορισμού της λειτουργίας του μεταφραστή, η οποία κάνει την όλη διαδικασία απλή, ανεξάρτητα από το είδος της υποκείμενης πηγής. Ο μεταφραστής εξάγει στους διαμεσολαβητές απόψεις



Σχήμα 3.3: Η αρχιτεκτονική του TSIMMIS

για τη σημασιολογία, τη σύνταξη και τελεστές των επερωτήσεων τις οποίες μπορεί να απαντήσει. Προκειμένου να περιγράψει τις επερωτήσεις, τις οποίες μπορεί να απαντήσει, ο μεταφραστής χρησιμοποιεί *Ελεύθερες Συμφραζομένων Γραμματικές (Context Free Grammars).*

Ένας μεταφραστής στο TSIMMIS παίρνει μια επερώτηση και αποφασίζει πρώτα απ' όλα αν η υποκείμενη πηγή μπορεί ή όχι να την απαντήσει, ή όπως αλλιώς λέγεται, αν είναι *άμεσα υποστηριζόμενη.* Αν ναι, τότε τη μεταφράζει σε μια επερώτηση στη γλώσσα της υποκείμενης πηγής και τη στέλνει σε αυτήν. Αφού πάρει τα αποτελέσματα, τα κωδικοποιεί στο OEM και τα στέλνει στο διαμεσολαβητή που τα ζήτησε. Αν η επερώτηση δεν είναι άμεσα υποστηριζόμενη, τότε ο μεταφραστής κοιτάζει αν μπορεί να απαντηθεί με *τοπικό φιλτράρισμα*, δηλαδή με την αναπαράστασή της μέσω μιας σειράς από άμεσα υποστηριζόμενες επερωτήσεις [GMPQ+97].

Για την δημιουργία και καθορισμό της λειτουργίας του μεταφραστή έχει κατασκευαστεί ένα εργαλείο το οποίο δέχεται ένα σύνολο από κανόνες της μορφής:

```
template
   // action //
```

Δέχεται, επίσης, ένα σύνολο συναρτήσεων, γραμμένες από το χρήστη, που χρησιμοποιούνται για τη σύνδεση με την υποκείμενη πηγή.

Έστω ένα παράδειγμα από ένα σύστημα με μουσειακά αντικείμενα όπου υπάρχει ο εξής κανόνας του μεταφραστή:

```
<paintings X> :-
    <library {X: <painting {<name X> <painter $AU>}>}>@s1
    //sprintf(lookup-query, "find painter %s", $AU)//
```

Οι κανόνες, γενικά, αποτελούνται από την επικεφαλίδα-φόρμα (template), και τη λειτουργία (action). Η επικεφαλίδα προσδιορίζει το είδος της επερώτησης στο μοντέλο OEM στην οποία αντιστοιχεί ο κανόνας αυτός, και η λειτουργία την επερώτηση που θα πρέπει να σταλεί στην υποκείμενη βάση. Έστω η παρακάτω επερώτηση στο OEM η οποία ταιριάζει με τον παραπάνω κανόνα και η οποία αναζητά τους πίνακες του Van Gogh.

```
<paintings B> :-
     <library {X: <painting {<name X> <painter "Van Gogh">}>}>@s1
```

Όταν η επερώτηση αυτή φτάσει στο μεταφραστή αυτός θα ψάξει όλους τους κανόνες του. Όταν φτάσει στον παραπάνω κανόνα θα βρει ότι ταιριάζει και θα αντικαταστήσει τη μεταβλητή AU με το όνομα Van Gogh. Έπειτα θα δημιουργήσει σύμφωνα με όσα λέει το τμήμα `action` την έκφραση: `find painter "Van Gogh"` και θα τη στείλει στην υποκείμενη πηγή προς εκτέλεση.

Σε περιπτώσεις που ο μεταφραστής του TSIMMIS δεν μπορεί να βρει κάποια φόρμα που να ταιριάζει με μια επερώτηση που φτάνει σε αυτόν, τότε η επερώτηση αυτή λέγεται ότι δεν είναι *άμεσα υποστηριζόμενη.* Σε μια τέτοια περίπτωση, ο μεταφραστής έχει τη δυνατότητα εκτέλεσης συλλογισμών και εύρεση τρόπων με τους οποίους μπορεί να απαντήσει την επερώτηση αυτή με χρήση *άμεσα υποστηριζόμενων* επερωτήσεων. Για παράδειγμα, έστω ότι η επερώτηση:

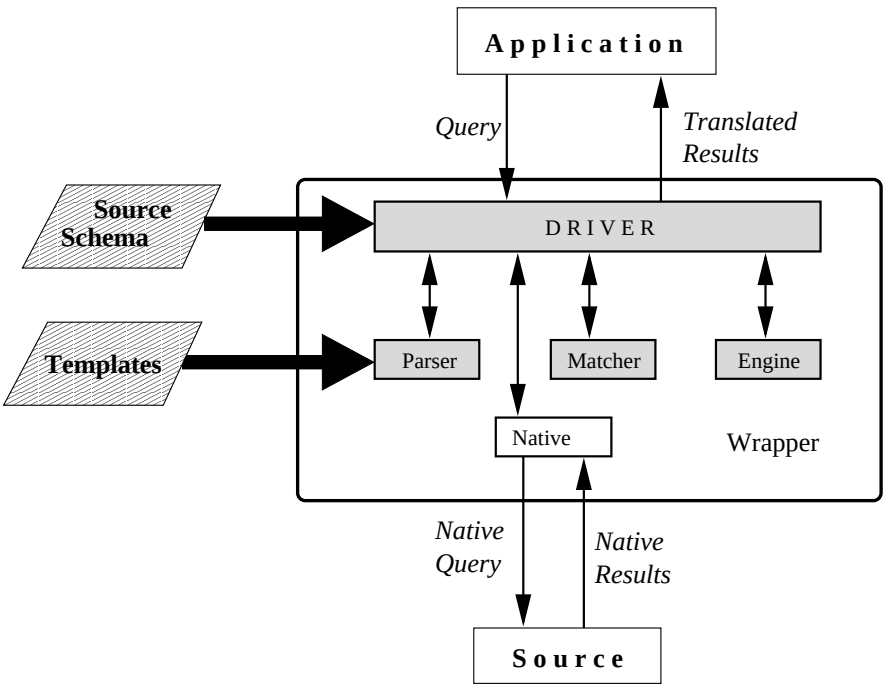
(1) "Δοσμένου ενός στοιχείου C, βρες τους γονείς του" (1)

είναι *άμεσα υποστηριζόμενη.* Η επερώτηση *"Δοσμένου ενός Α, Βρες όλους τους παππούδες και γιαγιάδες του"* μπορεί να υλοποιηθεί εφαρμόζοντας διαδοχικά δύο φορές την (1), ως εξής:

1. Βρες όλους τους γονείς του Α με χρήση της (1)
2. Για κάθε έναν που βρέθηκε από το βήμα 1 βρες τους γονείς του εφαρμόζοντας την (1).

Περισσότερα και αναλυτικότερα παραδείγματα ο αναγνώστης μπορεί να βρει στο [PGMGU95].

Με τη χρήση των φορμών ο μεταφραστής στο TSIMMIS περιγράφει στο διαμεσολαβητή το σύνολο των επερωτήσεων τις οποίες μπορεί να επεξεργαστεί, καθορίζοντας το τι είναι αυτό που μπορεί να ρωτήσει ο διαμεσολαβητής, καθώς και το πως θα πρέπει να το ρωτήσει. Στις φόρμες γίνεται χρήση δύο ειδών μεταβλητών. Των ελεύθερων (free) και των δεσμευμένων (bound). Ελεύθερη μεταβλητή είναι η "X" στην παραπάνω επερώτηση `<paintings X>` και η οποία καθορίζει το είδος των αντικειμένων που θα επιστρέψει η επερώτηση, στο παρόν παράδειγμα τα ονόματα των ζωγράφων. Οι δεσμευμένες μεταβλητές καθορίζουν τι είναι αυτό που μπορεί να ρωτήσει ο διαμεσολαβητής. Στο συγκεκριμένο παράδειγμα η μεταβλητή "AU" είναι δεσμευμένη και καθορίζει το ότι ο διαμεσολαβητής θα πρέπει, όταν κάνει την επερώτηση, να προσδιορίζει το όνομα του ζωγράφου, του οποίου τα ονόματα των πινάκων θέλει να ανακτήσει.



Σχήμα 3.4: Η αρχιτεκτονική του μεταφραστή στο TSIMMIS

Στο έργο αυτό έχει κατασκευαστεί μια εφαρμογή για την εύκολη και γρήγορη κατασκευή και προσδιορισμό του τρόπου λειτουργίας ενός μεταφραστή πάνω από κάποια πηγή. Στο σχήμα 3.4 φαίνεται η αρχιτεκτονική του μεταφραστή. Τα ελαφρά σκιασμένα τμήματα είναι κοινά για όλους τους μεταφραστές στο TSIMMIS ανεξάρτητα

από το είδος της υποκείμενης πηγής. Τα λευκά είναι εξαρτώμενα από τη πηγή και θα πρέπει να κατασκευάζονται από τον διαχειριστή για κάθε διαφορετικό είδος πηγής. Τα έντονα γραμμοσκιασμένα εξαρτώνται κι αυτά από την πηγή και αποτελούν τις προδιαγραφές λειτουργίας του μεταφραστή.

Το βασικό τμήμα είναι ο *driver*, ο οποίος ελέγχει τη διαδικασία μετάφρασης και καλεί τα εξής τμήματα:

- Τον *parser* ο οποίος αναλύει συντακτικά τις φόρμες (template) το τοπικό σχήμα της βάσης καθώς και τις επερωτήσεις που φτάνουν στον server.
- Τον *matcher* ο οποίος ψάχνει τις διάφορες φόρμες να βρει αυτή με την οποία ταιριάζει κάποια επερώτηση, και αν δεν βρει προσπαθεί να βρει ένα τρόπο για να απαντήσει έμμεσα την επερώτηση.
- Το *Native* τμήμα το οποίο στέλνει στην υποκείμενη πηγή την επερώτηση της φόρμας που βρέθηκε να ταιριάζει με την επερώτηση που έφτασε στο μεταφραστή. Επίσης μεταφράζει τα αποτελέσματα της πηγής στο κοινό μοντέλο δεδομένων.
- Το *Engine* τμήμα το οποίο μετατρέπει και δημιουργεί πακέτα με τα αποτελέσματα που έδωσε η πηγή, για να σταλούν στο διαμεσολαβητή

Τα πλεονεκτήματα της προσέγγισης που ακολουθείται στο TSIMMIS είναι πολλά και ποικίλα. Ορισμένα από αυτά είναι ότι:

- Ο συνδυασμός φόρμας και επερώτησης της πηγής, έχει το πλεονέκτημα ότι δεν περιορίζεται από άμεσες μεταφράσεις-αντιστοιχίσεις δομών και τελεστών του κοινού μοντέλου και γλώσσας στο μοντέλο και γλώσσα επερωτήσεων της πηγής.

  Η χρήση επερωτήσεων μέσω φορμών όπως στο TSIMMIS είναι ιδανική για περιπτώσεις όπου τα δεδομένα των πηγών αλλάζουν συχνά, χωρίς προειδοποίηση. Επίσης, η γλώσσα που χρησιμοποιείται στις επερωτήσεις είναι η γλώσσα της πηγής και έτσι μπορούν να αξιοποιηθούν στο έπακρο οι δυνατότητές της από ένα διαμεσολαβητή.
- Οι φόρμες αποτελούν την γνώση του μεταφραστή για τις δυνατότητες επερωτήσεων τις οποίες μπορεί να απαντήσει η πηγή. Τις δυνατότητες αυτές τις επαυξάνει με την ικανότητα συλλογισμών και σύνθεσης άμεσα υποστηριζόμενων επερωτήσεων για την απάντηση άλλων μη άμεσα υποστηριζόμενων.

Μια διαφοροποίηση μεταξύ της προσέγγισης που ακολουθεί το TSIMMIS και αυτής του Z39.50 είναι ότι στο TSIMMIS οι επερωτήσεις οι οποίες φτάνουν στον μεταφραστή

είναι σίγουρο ότι μπορούν να απαντηθούν από αυτόν, είτε άμεσα είτε έμμεσα. Σε αντίθεση στο Z39.50 ένα μέρος της δουλειάς του διαμεσολαβητή μεταφέρεται στο μεταφραστή. Όταν μια επερώτηση φτάνει σε αυτόν δεν είναι σίγουρο ότι μπορεί να υποστηριχθεί από την υποκείμενη βάση. Ο έλεγχος αν υποστηρίζεται ή όχι γίνεται από τον ίδιο το μεταφραστή.

Επίσης, το μοντέλο δεδομένων που χρησιμοποιεί το TSIMMIS είναι πολύ περισσότερο εκφραστικό παρέχοντας τη δυνατότητα να εκφράσει καλύτερα τη σύνταξη και σημασιολογία των δεδομένων της πηγής. Από την άλλη στο Z39.50 έχει επιλεγεί ο ελάχιστος πυρήνας, το ελάχιστο δυνατό κοινό τμήμα για την επίτευξη μεγαλύτερης διαλειτουργικότητας, με αποτέλεσμα το μοντέλο να μην είναι τόσο εκφραστικό. Το ίδιο ισχύει και για τη γλώσσα επερωτήσεων.

Κεφάλαιο 4

SIS/Z39.50 Gateway

Στην ενότητα αυτή παρουσιάζεται ο Z39.50 server ο οποίος κατασκευάστηκε στα πλαίσια της παρούσας εργασίας. Η ανάπτυξη του server αυτού έγινε με σκοπό την παροχή πρόσβασης στο *Σύστημα Σημασιολογικού Ευρετηριασμού (ή ΣΣΕ)* μέσω του πρωτοκόλλου Z39.50. Το *ΣΣΕ* είναι ένα σύστημα διαχείρησης βάσεων πληροφοριών τύπου σημασιολογικού δικτύου με δυνατότητες συλλογισμών. Η ανάγκη πρόσβασης σε αυτό μέσω του πρωτοκόλλου Z39.50 προέκυψε σε δύο ερευνητικά προγράμματα στα οποία συμμετέχει ο τομέας Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του Ινστιτούτου Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας: το **AQUARELLE** και το **CIMIzit**.

Το **AQUARELLE**[1] είναι ένα έργο του προγράμματος Telematics, το οποίο αποσκοπεί στο να κάνει δυνατή την πρόσβαση στις ψηφιακές πληροφορίες τις σχετικές με Ευρωπαϊκή πολιτισμική κληρονομιά. Σε αυτό συμμετέχουν εταιρίες και ερευνητικά ιδρύματα από την Ελλάδα, την Ιταλία, τη Γαλλία και την Αγγλία. Συντονίζεται από το ERCIM[2] και χρηματοδοτείται από την Ευρωπαϊκή Ένωση. Στο έργο αυτό το ΣΣΕ χρησιμοποιείται σαν ευρετήριο για κείμενα SGML στα οποία πρέπει να υπάρχει η δυνατότητα πρόσβασης, αναζήτησης και ανάκτησης μέσω του Z39.50.

Στο **CIMIzit**[3] (CIMI Interoperability Testbed Project), ένας αριθμός από ερευνητικά ιδρύματα και εταιρίες παροχής τεχνολογίας, καθώς και οργανισμοί από την πολιτισμική κοινότητα, δοκίμασαν τις δυνατότητες του πρωτοκόλλου Z39.50 όσον αφορά τη διαλειτουργικότητα κατά την ανταλλαγή ετερογενών δεδομένων. Απώτερος σκοπός ήταν η διαμόρφωση της τελικής μορφής του **CIMI**[4]**profile**. Για τη μελέτη της διαλειτουργι-

[1] http://aqua.inria.fr
[2] European Research Consortium for Informatics and Mathematics
[3] http://www.cimi.org/projects/testbed.html
[4] http://www.cimi.org

κότητας, ο Z39.50 server που κατασκευάστηκε, χρησιμοποιήθηκε με ενδεικτικό σύνολο δεδομένων από τη βάση CLIO [Chr94] του Μουσείου Μπενάκη.

Πριν γίνει η περιγραφή της υλοποίησης κρίνεται απαραίτητη μια σύντομη παρουσίαση του ΣΣΕ (ή SIS), κάτι που γίνεται στην επόμενη ενότητα. Στη συνέχεια περιγράφεται η αρχιτεκτονική και ο τρόπος λειτουργίας του μεταφραστή Z39.50. Στο τέλος του κεφαλαίου ακολουθεί μια σύντομη ανάλυση των προβλημάτων και των αδυναμιών του πρωτοκόλλου, τις οποίες ανέδειξε η εφαρμογή του στις δύο βάσεις δεδομένων που αναφέρθηκαν προηγουμένως.

4.1 Το Σύστημα Σημασιολογικού Ευρετηριασμού

Το *Σύστημα Σημασιολογικού Ευρετηριασμού (Semantic Index System ή SIS)* είναι ένα εργαλείο για περιγραφή και τεκμηρίωση μεγάλου πληθυσμού, ιδιόμορφων, εξελισσόμενων και πολλαπλώς συνδεδεμένων δεδομένων. Για το λόγο αυτό είναι κατάλληλο για την παράσταση επιστημονικής σχεδιαστικής και κατασκευαστικής γνώσης. Το ΣΣΕ αποτελείται από ένα μηχανισμό αποθήκευσης, που βασίζεται σε ένα συνδυασμό οντοκεντρικού μοντέλου και σημασιολογικού δικτύου, και από μια γενική διαλογική επαφή χρήσης για την εισαγωγή και ανάκτηση πληροφοριών. Το μοντέλο των δεδομένων περιγράφεται με τη γλώσσα παράστασης γνώσης Telos. Το ΣΣΕ και η ενσωματωμένη σε αυτό υλοποίηση του δομικού μέρους της γλώσσας Telos έχουν αναπτυχθεί από την Ομάδα Πληροφοριακών Συστημάτων και Τεχνολογίας Λογισμικού του Ινστιτούτου Πληροφορικής του Ιδρύματος Τεχνολογίας και Έρευνας[5] (ΙΤΕ).

4.1.1 Σημαντικά χαρακτηριστικά της γλώσσας Telos

Η Telos [MBJK90] είναι μια γλώσσα παράστασης γνώσης η οποία υποστηρίζει ένα οντοκεντρικό μοντέλο δεδομένων. Προσφέρει τους εξής στατικούς εκφραστικούς μηχανισμούς: Μη φραγμένη ιεραρχία ταξινόμησης, πολλαπλή και αυστηρή κληρονομικότητα και πλειότιμα γνωρίσματα, τα οποία με τη σειρά τους μπορεί να έχουν κι αυτά δικά τους γνωρίσματα. Επίσης, παρέχει μηχανισμούς χρονικής λογικής.

Η Telos παρέχει πολλές στάθμες αφαίρεσης. Άρα προσφέρει τη δυνατότητα οργάνωσης του σχήματος σύμφωνα με ένα μέτα-σχήμα, το οποίο περιγράφει πιο αφηρημένες έννοιες και ιδιότητες, επιτρέποντας έτσι την διατύπωση και απάντηση επερωτήσεων που βασίζονται σε αφηρημένες ιδιότητες. Με τη σειρά τους, οι έννοιες του μέτα-σχήματος

[5] http://www.csi.forth.gr

μπορεί να υπάγονται σε ένα μέτα-σχήμα κ.ο.κ. Στην εφαρμογή που θα παρουσιαστεί παρακάτω, η δυνατότητα αυτή δεν χρησιμοποιείται.

Οι βασικοί εκφραστικοί μηχανισμοί που χρησιμοποιεί η Telos για την δημιουργία μοντέλων παράστασης γνώσεων είναι οι παρακάτω.

- **Ονοματοδοσία**: Κάθε οντότητα έχει ένα εσωτερικό, παραγόμενο από το σύστημα, *αναγνωριστικό όνομα*, και ένα καθοριζόμενο από το χρήστη *λογικό όνομα*.

- **Ταξινόμηση**: Με τον μηχανισμό αυτό, μία ατομική οντότητα περιγράφεται ως *μέλος (περίπτωση)* μιας κλάσης, της οποίας κληρονομεί τα γνωρίσματα. Η Telos απαιτεί κάθε οντότητα να αποτελεί περίπτωση κάποιας κλάσης. Η απαίτηση αυτή εκπληρώνεται με την ύπαρξη ειδικών *κλάσεων συστήματος* οι οποίες δεν μπορούν να αλλαχτούν από το χρήστη. Αποτελούν τον αρχικό πληθυσμό της βάσης και οποιαδήποτε δεδομένα που θα εισαχθούν στο σύστημα, θα πρέπει άμεσα ή έμμεσα να συνδέονται με αυτές. Μια οντότητα μπορεί να ανήκει σε παραπάνω από μια κλάσεις.

- **Απόδοση γνωρίσματος**: Με τον μηχανισμό αυτό αποδίδονται γνωρίσματα στις οντότητες. Τα γνωρίσματα αυτά μπορούν να θεωρηθούν ως σχέσεις μεταξύ οντοτήτων μια και έχουν ένα *πεδίο ορισμού* και ένα *πεδίο τιμών*.

- **Περιορισμός στην ταξινόμηση γνωρισμάτων**: Αν ένα γνώρισμα είναι περίπτωση μιας κατηγορίας γνωρισμάτων, τότε το πεδίο ορισμού και το πεδίο τιμών του πρέπει να είναι περιπτώσεις των πεδίων ορισμού των τιμών της κλάσης γνωρισμάτων στην οποία ανήκει.

- **Γενίκευση (αντίστροφο: Εξειδίκευση)**: Ο μηχανισμός αυτός ισχύει μόνο για κλάσεις. Ορίζει μια σχέση υποσυνόλου μεταξύ των κλάσεων που ονομάζεται *isA*. Αν A *isA* B, τότε η A ονομάζεται *υποκλάση* της B και η B *υπερκλάση* της A. Η A κληρονομεί όλα τα γνωρίσματα της B και είτε έχει επιπλέον γνωρίσματα, είτε περιορίζει το σύνολο τιμών των γνωρισμάτων που κληρονομεί από την B. Μια κλάση μπορεί να έχει παραπάνω από μία υπερκλάσεις. Έτσι, η σχέση *isA* υποστηρίζει πολλαπλή και αυστηρή κληρονόμηση. Ο μηχανισμός αυτός επιτρέπει την οργάνωση των κλάσεων σε διάφορες ιεραρχίες γενίκευσης οι οποίες προσδίδουν οικονομία και συνέπεια στο μοντέλο, αφού δεν χρειάζεται να επαναληφθεί ο ορισμός ενός γνωρίσματος που έχει ήδη αποδοθεί σε μια γνωστή υπερκλάση.

4.1.2 Περιγραφή του ΣΣΕ

Το *Σύστημα Σημασιολογικού Ευρετηριασμού* παρέχει τους παρακάτω μηχανισμούς διαχείρισης.

- **Μηχανισμούς Αποθήκευσης Αντικειμένων**: Ο μηχανισμός αυτός υποστηρίζει δοσοληψίες και ταυτόχρονη πρόσβαση από πολλούς χρήστες.

- **Μηχανισμός επερωτήσεων**: Το ΣΣΕ συνοδεύεται από ένα μηχανισμό επερωτήσεων ο οποίος παρέχει ένα σύνολο από ερωτηματικές εντολές, με τις οποίες είναι δυνατή η πλοήγηση στο σύνολο της πληροφορίας που έχει παρασταθεί στη βάση καθώς και η διατύπωση αναδρομικών επερωτήσεων με πολλαπλά κριτήρια. Οι επερωτήσεις μπορούν να γίνουν από το χρήστη είτε μέσω ενός διαλογικού εργαλείου, είτε από άλλες εφαρμογές με τη βοήθεια της *προγραμματιστικής διεπαφής επερωτήσεων (PQI)* [Ins97].

- **Μηχανισμός Εισαγωγής Δεδομένων**: Η εισαγωγή δεδομένων στο *ΣΣΕ* γίνεται με δύο τρόπους:

  ▷ Ασύγχρονη εισαγωγή δεδομένων με χρήση του συντακτικού αναλυτή της Telos

  ▷ Χρήση *Δελτίων Εισαγωγής Δεδομένων* για σύγχρονη εισαγωγή και τροποποίηση στοιχείων, με αλληλεπίδραση χρήστη και *ΣΣΕ*.

- **Μηχανισμός Προγραμματικής Διεπαφής Επερωτήσεων**: Η διασύνδεση και επικοινωνία του *ΣΣΕ* με εξωτερικές εφαρμογές υποστηρίζονται από ένα σύνολο συναρτήσεων με τις οποίες είναι δυνατή η άντληση πληροφορίας που έχει παρασταθεί στο *ΣΣΕ* από τις εφαρμογές αυτές.

  Η επικοινωνία μίας εξωτερικής εφαρμογής με το *ΣΣΕ* γίνεται ως εξής:

  1. Εγκαθίσταται η επικοινωνία της εφαρμογής και του *ΣΣΕ*
  2. Τίθεται ο κόμβος στον οποίο θα εφαρμοστεί κάποια εκ των ερωτηματικών συναρτήσεων του PQI.
  3. Χρησιμοποιούνται οι κατάλληλες ερωτηματικές συναρτήσεις του PQI για αναζήτηση πληροφορίας στη Βάση Γνώσης (ΒΓ) του *ΣΣΕ*.
  4. Επαναλαμβάνονται τα βήματα 2 και 3 (προαιρετικά) και χρησιμοποιούνται κατάλληλες συναρτήσεις διαχείρισης συνόλων απαντήσεων του PQI.

5. Τερματίζεται η επικοινωνία της εφαρμογής και του *ΣΣΕ*.

Το σύνολο συναρτήσεων που προσφέρει το PQI κατατάσσονται ανάλογα με τη λειτουργικότητά τους σε πέντε κατηγορίες:

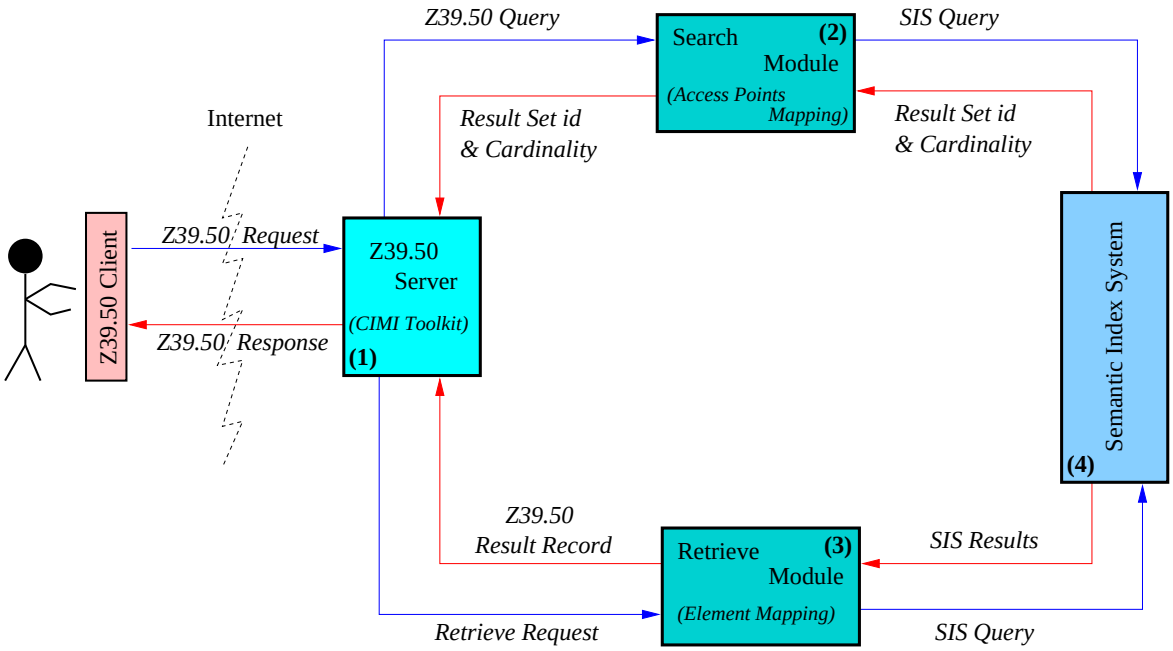
- ▷ Συναρτήσεις οι οποίες είναι υπεύθυνες για την έναρξη/λήξη της επικοινωνίας μεταξύ της εξωτερικής εφαρμογής και του *ΣΣΕ*.
- ▷ Συναρτήσεις για τροποποίηση γενικών παραμέτρων που χρησιμοποιούνται από τις ερωτηματικές συναρτήσεις.
- ▷ Συναρτήσεις που χρησιμοποιούνται για επερωτήσεις στη ΒΓ. Με τις συναρτήσεις αυτές είναι δυνατό να ανακτηθεί πληροφορία από τη βάση γνώσης όπως για παράδειγμα:
  - -- Τιμές γνωρισμάτων μιας οντότητας
  - -- Οντότητες των οποίων αποτελεί γνώρισμα μια οντότητα
  - -- Κλάσεις των οποίων μια οντότητα αποτελεί περίπτωση
  - -- Περιπτώσεις μιας οντότητας
  - -- Υπερκλάσεις/Υποκλάσεις μιας οντότητας
- ▷ Συναρτήσεις χειρισμού συνόλων αποτελεσμάτων
- ▷ Συναρτήσεις ανάκτησης πληροφορίας από τα σύνολα αποθήκευσης.

4.2 Αρχιτεκτονική του Z39.50 server

Για την υλοποίηση του συστήματος προτιμήθηκε ο δρόμος της ανοικτής αρχιτεκτονικής, ώστε να είναι εύκολη η πιθανή αντικατάσταση τμημάτων με άλλα που έχουν καλύτερη απόδοση. Η όλη αρχιτεκτονική της λειτουργίας του μεταφραστή φαίνεται στο σχήμα 4.1.

4.2.1 Z39.50 server

Το τμήμα που αποκαλείται ”Z39.50 Server” (το (1)) είναι το τμήμα εκείνο της εφαρμογής το οποίο είναι υπεύθυνο να δέχεται μηνύματα-πακέτα Z39.50 από το δίκτυο, να τα αποκωδικοποιεί, καθώς επίσης να τα κωδικοποιεί και να τα στέλνει. Είναι δηλαδή υπεύθυνο για την επικοινωνία στο δίκτυο. Για την κατασκευή του τμήματος αυτού χρησιμοποιήθηκε το *CIMI toolkit* της εταιρίας SSL και το οποίο στηρίζεται στο πακέτο εφαρμογών YAZ [Ind97]. Για την επικοινωνία του με το υπόλοιπο σύστημα το *CIMI*



Σχήμα 4.1: Αρχιτεκτονική του Z39.50 server

toolkit παρέχει μια σειρά από συναρτήσεις, τις οποίες και καλεί με κάποια συγκεκριμένα ορίσματα, ανάλογα με το μήνυμα το οποίο λαμβάνεται από το δίκτυο. Ο κατασκευαστής της εφαρμογής θα πρέπει να γράψει τον κώδικα των συναρτήσεων αυτών, για να γίνει δυνατή η επικοινωνία του εργαλείου με το υπόλοιπο σύστημα. Οι συναρτήσεις αυτές είναι οι εξής:

- **<cimi_open>:** Καλείται κάθε φορά που ένας καινούριος client συνδέεται στο Z39.50 server, και σκοπός της είναι να κάνει έλεγχο δικαιώματος πρόσβασης, και αρχικοποίηση ορισμένων παραμέτρων της επικοινωνίας. Σε περίπτωση λάθους επιστρέφει τον κωδικό λάθους που έλαβε χώρα.

- **<cimi_search>:** Καλείται όταν στον server φτάνει μια *Search Request*. Ένα από τα ορίσματά της είναι ένα δυαδικό δένδρο το οποίο αναπαριστά την επερώτηση που περιέχονταν στην αίτηση. Σκοπός της είναι η εκτέλεση της Z39.50 επερώτησης και η επιστροφή ενός συνόλου απαντήσεων. Για το σκοπό αυτό καλεί το τμήμα *Search* (τμήμα (2))με τα ορίσματά της, και επιστρέφει την τιμή που θα της δώσει το τμήμα αυτό, το οποίο είναι κωδικός λάθους σε περίπτωση που συμβεί κάποιο λάθος, ή το όνομα και ο πληθάριθμος ενός συνόλου απαντήσεων.

- **<cimi_present>:** Καλείται κάθε φορά που μια αίτηση ανάκτησης φτάνει στον server. Σαν όρισμα δέχεται ένα αριθμό ο οποίος προσδιορίζει ένα αντικείμενο μέσα σε ένα σύνολο απαντήσεων, και την πληροφορία που πρέπει να ανακτηθεί γι' αυτό (προσδιορίζοντας κάποια από τις εγγραφές που ορίζονται στο προφίλ). Σκοπός της είναι η ανάκτηση και επιστροφή της εγγραφής του αντικειμένου αυτού. Για να γίνει αυτό καλεί το τμήμα *Retrieve* (τμήμα (3)), με τα ίδια ορίσματα. Το τμήμα αυτό θα επιστρέψει τη ζητούμενη εγγραφή, σε κάποια από τις προκαθορισμένες κωδικοποιήσεις (GRS-1, SUTRS, κλπ).

- **<cimi_close>:** Όπως είναι προφανές από το όνομα, καλείται κάθε φορά που κάποιος client διακόψει τη σύνδεση του με τον server, είτε εκούσια επειδή τελείωσε με τα όσα ήθελε να ανακτήσει, είτε ακούσια, όπως για παράδειγμα λόγο προβλήματος του δικτύου.

Οι παράμετροι της επικοινωνίας με τον client όσο και τα ΣΠ τα οποία θα υποστηρίζει ο server καθορίζονται στο *CIMI toolkit* μέσω κάποιων αρχείων παραμέτρων.

4.2.2 Τμήμα Αναζήτησης

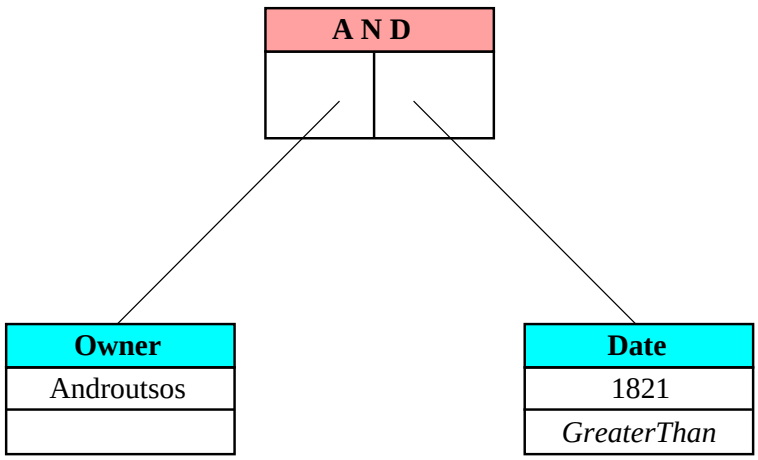
Το τμήμα αυτό, (τμήμα (2)), είναι υπεύθυνο για τη μετάφραση των Z39.50 επερωτήσεων, σε επερωτήσεις PQI οι οποίες θα σταλούν στο ΣΣΕ και θα επιστρέψουν ένα σύνολο απαντήσεων. Για την απάντηση μιας επερώτησης γίνεται bottom-up διάσχιση του συντακτικού της δένδρου, και καθώς γίνεται αυτό, κάθε κόμβος μεταφράζεται σε επερώτηση PQI και εκτελείται. Ένας κόμβος του δένδρου μπορεί να είναι είτε *κόμβος-όρος* είτε *κόμβος-πράξη*. Στο σχήμα 4.2 φαίνεται το συντακτικό δένδρο της επερώτησης:

Owner="Androutsos" ***AND*** **(Date**=1821 ***AND*** **Relation**="GreaterThan" **)**

Ένας *κόμβος-όρος* αναφέρεται πάντοτε σε ένα και μόνο ΣΠ. Για παράδειγμα, η επερώτηση **Owner**="Androutsos" παρίσταται από ένα τέτοιο κόμβο. Στην περίπτωση που έχουν ζητηθεί για το ΣΠ του κόμβου *Truncation* ή *Relation* ιδιότητες, οι πληροφορίες αυτές θα εμπεριέχονται στον *κόμβο-όρο*. Ένας τέτοιος κόμβος στο συντακτικό δένδρο της επερώτησης είναι πάντοτε φύλλο. Στο σχήμα 4.2 οι κόμβοι-όροι είναι τα δύο φύλλα.

Για την αντιστοίχιση και παράλληλα τη μετάφραση των ΣΠ στις δομές του ΣΣΕ έχει οριστεί ο πίνακας *mappingS* ως εξής:

```
typedef int (*mapFunc)(int);
typedef int (*initFunc)(void);
```



Onwer="Androutsos" AND (Date=1821 AND Relation="GreaterThan")

Σχήμα 4.2: Μια Z39.50 επερώτηση και το συντακτικό της δέντρο.

```
struct    mappingS
{
        char *val;
        initFunc ifunc;
        mapFunc mfunc;
}
```

Κάθε εγγραφή του πίνακα αυτού αντιστοιχεί σε ένα ΣΠ. Το πεδίο *val* αναπαριστά το όνομα του ΣΠ. Η *ifunc* είναι ένας δείκτης σε μία συνάρτηση η οποία περιέχει μια σειρά από επερωτήσεις PQI οι οποίες δημιουργούν ένα σύνολο οντοτήτων της βάσης που είναι σημασιολογικά ισοδύναμες με το ΣΠ. Έτσι, αν για παράδειγμα το ΣΠ είναι το *Date*, η αντίστοιχη σε αυτό *ifunc* θα επέστρεφε το σύνολο των ημερομηνιών που βρίσκονται αποθηκευμένες στη βάση. Η *mfunc* είναι ένας δείκτης σε μια συνάρτηση η οποία περιέχει και αυτή ένα σύνολο από επερωτήσεις PQI και η οποία δέχεται σαν όρισμα ένα σύνολο αντικειμένων που αντιστοιχούν στο ΣΠ (αν θεωρήσουμε το προηγούμενο παράδειγμα, ένα σύνολο ημερομηνιών), και επιστρέφει ένα σύνολο από αντικείμενα της βασικής έννοιας της βάσης τα οποία συσχετίζονται με τα αντικείμενα του συνόλου που της δόθηκε σαν όρισμα. Έτσι στο παρόν παράδειγμα, αν υποτεθεί ότι έχουμε μουσειακή βάση δεδομένων, η *mfunc* θα επιστρέψει το σύνολο των μουσειακών αντικειμένων τα

οποία σχετίζονται με τις ημερομηνίες που περιέχονταν στο σύνολο που της δόθηκε σαν όρισμα.

Για την μετάφραση επομένως ενός κόμβου-όρου, ακολουθούνται τα εξής βήματα:

- Αναζητείται στον πίνακα *mappingS* μια εγγραφή με το όνομα του ΣΠ που περιέχεται στον κόμβο.

- Καλείται η αντίστοιχη *ifunc* η οποία κάνει ουσιαστικά την αντιστοίχιση, και κατασκευάζει ένα σύνολο αντικειμένων.

- Επιλέγεται από το σύνολο που έδωσε η *ifunc* τα αντικείμενα που ικανοποιούν τα κριτήρια της επερώτησης. Στο παράδειγμα της Date του σχήματος 4.2, η *ifunc* θα επέστρεφε όλες τις ημερομηνίες της βάσης, και από αυτές θα επιλέγονταν μόνο όσες είναι μεγαλύτερες του 1821.

- Το τελικό σύνολο του προηγούμενου βήματος δίδεται στην αντίστοιχη $mfunc$ του ΣΠ, και το σύνολο που θα επιστραφεί αποτελεί το σύνολο απαντήσεων για τον κόμβο όρο.

Με τον αλγόριθμο που μόλις περιγράφηκε μεταφράζονται τα ΣΠ και καθώς και τα *Relation* και *Truncation attributes* του Z39.50 στις δομές του ΣΣΕ.

Από την άλλη μεριά οι τελεστές της **type-1** γλώσσας, *AND*, *OR*, και *AND_NOT* διαχειρίζονται πιο απλά. Ένα τέτοιος τελεστής στο συντακτικό δένδρο αναπαρίσταται από τους *κόμβους-πράξεις*. (Οι *κόμβοι-πράξεις* δεν είναι ποτέ φύλλα). Για τη μετάφραση αυτών, καλείται μια αντίστοιχη συνάρτηση συνόλων του PQI η οποία σαν ορίσματα παίρνει τα σύνολα απαντήσεων που έδωσαν τα παιδιά του κόμβου κατά τη μετάφραση και εκτέλεσή τους, κάτι που είναι εγγυημένο ότι έχει γίνει τη στιγμή που επεξεργάζεται και μεταφράζεται ο *κόμβος-πράξη* λόγω της **bottom-up** διάσχισης. Έτσι, για τους τελεστές *AND*, *OR*, και *AND_NOT*, καλούνται οι συναρτήσεις για τομή, ένωση, ή διαφορά συνόλων του PQI αντίστοιχα. Τα σύνολα που δίνονται σαν ορίσματα σε αυτές τις συναρτήσεις θα είναι πάντοτε σύνολα ομοειδών αντικειμένων, μια και όλα τα σύνολα απαντήσεων είναι σύνολα από αντικείμενα της βασικής έννοιας της βάσης (βλέπε 4.3.2).

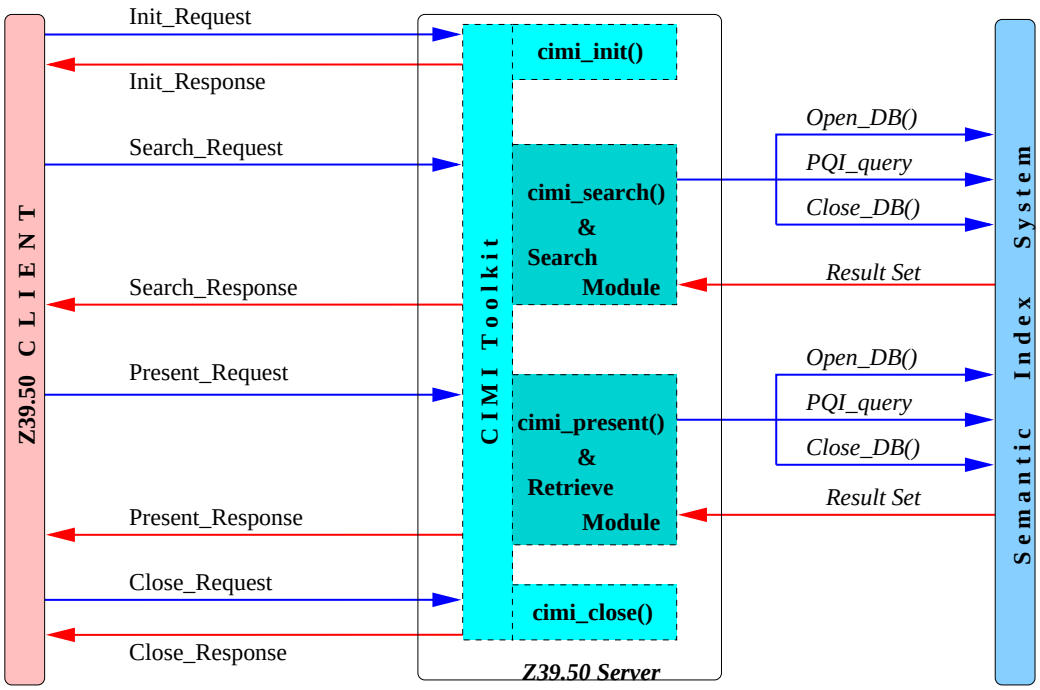
Όλα τα ενδιάμεσα σύνολα απαντήσεων ελευθερώνονται και μονάχα το τελικό που θα προκύψει από την μετάφραση και εκτέλεση (evaluation) του ριζικού κόμβου του συντακτικού δένδρου, αποτελεί την τελική απάντηση, το οποίο και επιστρέφεται στο τμήμα (1).

4.2.3 Τμήμα Ανάκτησης

Το τμήμα *Retrieve* (τμήμα (3)), ή τμήμα *Ανάκτησης* είναι υπεύθυνο για την ανάκτηση των απαραίτητων δεδομένων από το ΣΣΕ, και την κατασκευή της εγγραφής κάποιου στοιχείου ενός συνόλου απαντήσεων. Σαν παράμετρο, δέχεται το μοναδικό κωδικό του αντικειμένου που αναπαριστά το στοιχείο.

Έστω για παράδειγμα, ότι ζητείται ανάκτηση της εγγραφής του 5ου στοιχείου ενός συνόλου απαντήσεων. Το τμήμα (1) γνωρίζει τον κωδικό που έχει το 5ο στοιχείο και καλεί το τμήμα *Ανάκτησης* με παράμετρο τον κωδικό αυτό (έστω ότι ο κωδικός αυτός αντιστοιχεί στην "σπάθα" του Ο. Ανδρούτσου). Αυτό με τη σειρά του, μέσω μιας σειράς επερωτήσεων PQI θα ανακτήσει τις σχετικές με τη "σπάθα" πληροφορίες, όπως για παράδειγμα τον ιδιοκτήτη, το έτος κατασκευής, την αξία της, το υλικό από το οποίο έχει φτιαχτεί, πιθανές φωτογραφίες της, κλπ. Με όλα τα στοιχεία αυτά κατασκευάζει μια εγγραφή της οποίας η δομή περιγράφεται από το προφίλ, και την επιστρέφει στο τμήμα (1) αφού πρώτα την κωδικοποιήσει σε GRS-1.

Το ποια θα είναι τα πεδία της βάσης που θα πρέπει να ανακτηθούν για να συνθέσουν την εγγραφή, είναι ενσωματωμένο μέσα στο δυαδικό κώδικα του τμήματος *Ανάκτησης*, και ακολουθεί τις οδηγίες που δίνονται στο προφίλ.



Σχήμα 4.3: Οι σειρά των κλήσεων μεταξύ client, server, και ΣΣΕ

Στο σχήμα 4.3 φαίνεται η σειρά με την οποία καλούνται οι συναρτήσεις των διαφόρων τμημάτων, ανάλογα με τη την αίτηση (Request) η οποία φτάνει στον server. Οι σειρά εκτέλεσης είναι αυτή που φαίνεται στο σχήμα, από πάνω προς τα κάτω.

4.2.4 Z39.50 client

Μέχρι σήμερα έχει κατασκευαστεί ένας ικανοποιητικός αριθμός από Z39.50 client για διάφορες εφαρμογές. Τα περιβάλλοντα στα οποία έχουν κτιστεί είναι πολλά και ποικίλα. Περισσότερο συνηθισμένοι είναι οι server οι οποίοι λειτουργούν μέσω κάποιου εργαλείου πλοήγησης του Διαδικτύου, όπως ο Netscape Communicator[6], ή ο Internet Explorer[7]. Πολλοί client όπως αυτός της Blue Angel Technologies[8] έχουν κτιστεί σε Java[9], ωστόσο η μεγάλη πλειοψηφία βασίζεται στη χρήση cgi-scripts. Μία καλή έρευνα για τους υπάρχοντες Z39.50 server έχει γίνει από το DSTC[10] της Αυστραλίας. Ο λόγος για τον οποίο χρησιμοποιούνται τα εργαλεία πλοήγησης είναι ότι η χρήση τους είναι απλή και αρκετά προσιτή στον απλό χρήστη, καθώς επίσης γιατί μπορούν εύκολα να υποστηρίξουν αντικείμενα πολυμέσων.

Κατά τη φάση κατασκευής του Z39.50 server στα δύο έργα που έχουμε αναφέρει, ήταν απαραίτητη η ύπαρξη κάποιου client με τη βοήθεια του οποίου θα δοκιμάζονταν η σωστή λειτουργία του server. Ο client ο οποίος θα χρησιμοποιούνταν στα έργα αυτά ήταν ευθύνη κατασκευής άλλων συνεργαζομένων φορέων. Στο επίσημο κόμβο του AQUARELLE [AQU] μπορεί κανείς να δει τη διεπαφή χρήσης του χρησιμοποιούμενου στο έργο αυτό client.

Ο client τον οποίο εμείς κατασκευάσαμε για τις δοκιμές του server, στηρίζεται στο εργαλείο YAZ [Ind97] και δεν έχει κάποια γραφική διεπαφή χρήσης. Λειτουργεί από τη γραμμή εντολών του φλοιού. Όταν τρέχει εμφανίζει στο χρήστη ένα μήνυμα προτροπής στο οποίο τον καλεί να γράψει μια εντολή. Οι εντολές που υποστηρίζονται είναι οι βασικές για άνοιγμα ή κλείσιμο επικοινωνίας με κάποιο server, για αποστολή επερωτήσεων, καθώς και για αιτήσεις ανάκτησης εγγραφών. Παρόλη την έλλειψη διεπαφής χρήσης, λειτουργεί πολύ καλά και αποδοτικά σε περιπτώσεις πρόσβασης σε βιβλιογραφικές βάσεις δεδομένων ή γενικότερα σε πηγές πληροφοριών όπου δεν υπάρχουν αντικείμενα προς ανάκτηση τα οποία να απαιτούν ειδική διαχείριση για την

[6] http://www.netscape.com
[7] http://www.microsoft.com
[8] http://www.bluangel.com
[9] http://sunsite.unc.edu/javafaq
[10] http://www.dstc.edu.au/RDU/reports/zreviews/z3950-client- survey.html

παρουσίασή τους στο χρήστη όπως για παράδειγμα οι εικόνες.

4.3 Προβλήματα του Z39.50

Κατά την περίοδο υλοποίησης του παραπάνω server, έγινε ορατή μια σειρά προβλημάτων του πρωτοκόλλου. Στην ενότητα αυτή γίνεται μια σύντομη ανάλυση των προβλημάτων αυτών.

4.3.1 Προϋπάρχουσα μη αναπαριστάμενη γνώση

Πολλές φορές, υπάρχουν πληροφορίες οι οποίες είναι χαρακτηριστικές για μια πηγή δεδομένων, ωστόσο, δεν έχουν ρητά παρασταθεί σε αυτή. Για παράδειγμα, έστω μια βάση δεδομένων στην οποία έχουν καταχωρηθεί τα αντικείμενα από την συλλογή όπλων του Μουσείου Μπενάκη. Έστω επίσης, ότι δεν υπάρχει κανένα πεδίο στη βάση το οποίο να δίνει την πληροφορία για τη θέση των αντικειμένων. Είναι όμως προφανές, ότι για όλα τα καταχωρημένα στη βάση αντικείμενα, η πληροφορία για τη θέση υπάρχει και είναι κατ' αρχήν το Μουσείο Μπενάκη.

Αν για την βάση αυτή κατασκευαστεί ένας Z39.50 server, τίθεται το ερώτημα της υποστήριξης ή μη σημείων πρόσβασης όπως αυτό της θέσης που αναφέρθηκε προηγουμένως. Αν δεν υποστηριχθεί, ένα τμήμα της περιεχόμενης πληροφορίας της πηγής αποκρύπτεται. Από την άλλη μεριά, αν υποστηριχθεί, θα πρέπει να τύχει ειδικού χειρισμού διότι δεν υπάρχει κάποια δομή στην πηγή η οποία να εκφράζει την πληροφορία αυτή. Το φαινόμενο αυτό αποτελεί μια από τις περιπτώσεις οι οποίες δημιουργούν την ανάγκη για την εισαγωγή ενός ενδιάμεσου επιπέδου ανάμεσα στην πηγή και στο Z39.50 server αυτής.

Σε περιπτώσεις όπου η πηγή δεδομένων είναι ένα ΣΑΠ, η μετάφραση των Z39.50 επερωτήσεων όπως έχει ήδη παρουσιαστεί σε προηγούμενο κεφάλαιο, είναι απλή και άμεση. Τα ΣΑΠ χρησιμοποιούν τεχνικές ευρετηριασμού πλήρους κειμένου, καθώς και αναζήτηση με βάση λέξεις-κλειδιά. Για τέτοιου είδους τεχνικές, πολλές φορές είναι απαραίτητο να χρησιμοποιηθεί κάποιος θησαυρός όρων.

Η χρήση του θησαυρού όρων σε κατανεμημένα και ετερογενή συστήματα αποτελεί ένα πολύ χρήσιμο εργαλείο. Από τα γνωστότερα προβλήματα ετερογένειας είναι το πρόβλημα των διαφορετικών ονομάτων, η περίπτωση δηλαδή όπου η ίδια πληροφορία, σε διαφορετικές πηγές δεδομένων, παρίσταται με διαφορετικό όνομα. Αν λοιπόν ο χρήστης χρησιμοποιήσει στην επερώτησή του ένα όρο για κάποιου είδους πληροφορία, η

αναζήτηση με βάση αυτόν θα αποτύχει, ακόμα κι αν η πηγή περιέχει την αναζητούμενη πληροφορία, αν αυτή έχει παρασταθεί με διαφορετικό όνομα.

Παρόμοιο είναι το πρόβλημα των διαφορετικών γλωσσών. Αν ο server στέλνει μια επερώτηση όπου οι όροι είναι σε κάποια γλώσσα Α, ενώ τα δεδομένα στην πηγή είναι σε γλώσσα Β, η επερώτηση θα αποτύχει.

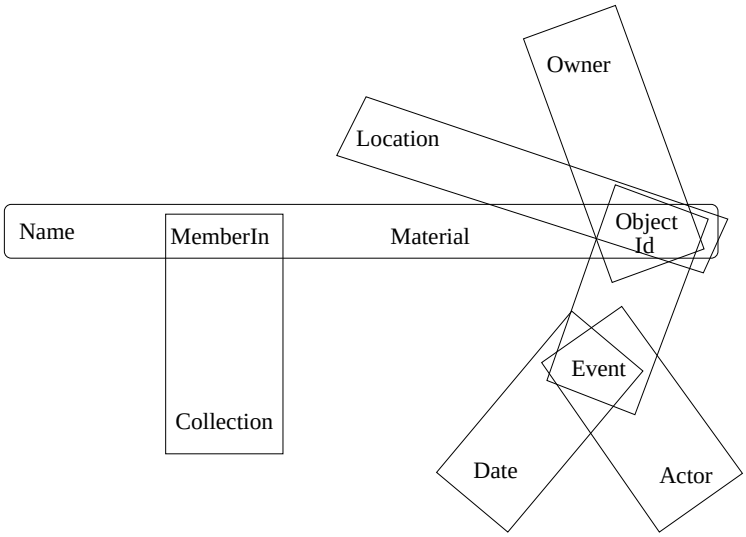
Από τα παραπάνω, γίνεται φανερή η ανάγκη ύπαρξης κάποιου ενδιάμεσου επιπέδου ανάμεσα στο Z39.50 server και στην πηγή, το οποίο θα αναλαμβάνει να κάνει τις απαραίτητες μετατροπές στους περιεχόμενους στην επερώτηση όρους ώστε να είναι δυνατή η επιτυχής εκτέλεσή της.

4.3.2 Συγκεντρωτική θεώρηση

Ένα άλλο βασικό πρόβλημα του Z39.50 είναι η συγκεντρωτική θεώρηση του για τις διάφορες πηγές πληροφορίας. Επηρεασμένο πάλι από τις βιβλιογραφικές πηγές, οι οποίες περιγράφουν ένα και μόνο είδος αντικειμένου, τα κείμενα, το Z39.50 υιοθετεί τη θεώρηση ότι όλα τα στοιχεία τα οποία βρίσκονται αποθηκευμένα σε μια πηγή πληροφοριών αναφέρονται σε ένα και μόνο είδος δεδομένου. Έτσι σε μια μουσειακή πηγή, για παράδειγμα, μπορεί να θεωρείται ότι όλα τα ΣΠ περιγράφουν πληροφορία που συσχετίζεται με μουσειακά αντικείμενα, τα οποία και αποτελούν το αντικείμενο ενδιαφέροντος προς ανάκτηση. Κάτι τέτοιο είναι πολύ περιοριστικό γιατί δε δίνει στο χρήστη τη δυνατότητα να ανακτήσει την πληροφορία που αυτός θέλει, ακόμα και αν η πληροφορία αυτή υπάρχει στα δεδομένα της πηγής. Για παράδειγμα, παρόλο που στη βάση του Μουσείου Μπενάκη μπορεί να υπάρχουν πληροφορίες για ιστορικά πρόσωπα, ο χρήστης δεν μπορεί ποτέ να τις ανακτήσει μέσω του Z39.50. Ακόμα και η επερώτηση πάνω στο ΣΠ *PersonalName* το οποίο αναφέρεται σε αυτά, έχει το νόημα των μουσειακών αντικειμένων που συσχετίζονται με τα συγκεκριμένα πρόσωπα, κι όχι τα πρόσωπα αυτά καθ' αυτά. Το αντικείμενο ενδιαφέροντος σε κάθε βάση ονομάζεται *κεντρική έννοια*.

Κάτω από μια τέτοια θεώρηση το εννοιολογικό σχήμα της βάσης δεδομένων θα πρέπει να έχει μια μορφή όπως αυτή που φαίνεται στο σχήμα 4.4. Η θεώρηση αυτή λέει ότι το σχήμα της βάσης θα πρέπει να ακολουθεί μια καθολική σχέση (σε πρώτη ή μη κανονική μορφή). Στο σχήμα παρίσταται μια όχι πρώτη κανονικής μορφής (N1NF) καθολική σχέση [DGS87]. Αν κάτι τέτοιο δεν συμβαίνει, κατά τη διαδικασία των αντιστοιχίσεων οι διαχειριστές θα πρέπει να πάρουν αποφασείς για το αν θα υποστηριχθεί ή όχι κάποιο ΣΠ για το οποίο δεν υπάρχει μονοπάτι στο σχήμα το οποίο να το συνδέει με την κεντρική έννοια. Τέτοιου είδους αποφάσεις επηρεάζουν την ανάκληση (recall) και

την ακρίβεια (precision) των απαντήσεων, ανάλογα με την επερώτηση του client.



Σχήμα 4.4: Μια N1NF καθολική σχέση

4.3.3 Γλώσσα επερωτήσεων

Ένα από τα βασικά χαρακτηριστικά των πολιτισμικών εφαρμογών που είναι σήμερα διαθέσιμες στο Διαδίκτυο, είναι ότι δεν υπάρχει κάποιο σταθερό ομοιόμορφο σχήμα για τα δεδομένα. Οι υπάρχουσες πληροφορίες εκτείνονται από ισχυρά δομημένες (σχεσιακές ή οντοκεντρικές βάσεις δεδομένων), σε ημιδομημένες όπου η δομή είναι ρητώς δηλωμένη, έμμεση ή πολύ χαλαρή (SGML/HTML κείμενα, βιβλιογραφικές βάσεις), έως αδόμητες (εικόνες, σχέδια).

Ο βασικός περιορισμός του Z39.50 είναι ότι είναι ισχυρά επηρεασμένο από τα επίπεδου σχήματος Συστήματα Ανάκτησης Πληροφοριών (ΣΑΠ), τα οποία βασίζονται σε μια προσανατολισμένη προς τις εγγραφές περιγραφή χρησιμοποιώντας μια λίστα από σημεία πρόσβασης. Για παράδειγμα η πλειοψηφία των μηχανών αναζήτησης που υπάρχουν σήμερα πάνω στο Διαδίκτυο (π.χ. Altavista[11], Lycos[12], Excite[13], Yahoo[14]) βασίζονται σε αναζητήσεις με βάση λέξεις-κλειδιά, υποστηριζόμενες οι περισσότερες από έναν ευρετηριασμό πλήρους κειμένου (full-text indexing). Ωστόσο, τέτοιες μέθοδοι είναι περισσότερο κατάλληλες για επερωτήσεις σε συλλογές αδόμητων ή ημιδομημένων

[11]http://www.altavista.digital.com/cgi-bin/query
[12]http://www.lycos.com
[13]http://www.excite.com
[14]http://www.yahoo.com

κειμένων, παρά για υψηλά δομημένες πηγές πληροφοριών, όπως αυτές των πολιτισμικών εφαρμογών. Η δομή μπορεί να υπάρχει σε σχεσιακές ή οντοκεντρικές βάσεις δεδομένων, ή σε σύγχρονα SGML-ΣΑΠ (SGML-based IRS), και μπορεί να είναι φανερή, άγνωστη, ή κρυμμένη και να πρέπει να ανακαλυφθεί [CCP+97].

Η δομή της πληροφορίας είναι εξαιρετικά χρήσιμη:

- για συμβολή στη βελτιστοποίηση επερωτήσεων και αύξηση της ακρίβειας,
- για περισσότερο εύκολη διαχείριση τμημάτων πληροφορίας και πλοήγηση σε αυτά, κάτι αντίστοιχο με την πλοήγηση σε υπερκείμενα, και επίσης,
- για πιο αξιόπιστη και πολύπλοκη ολοκλήρωση της πληροφορίας.

Κάτω από το επίπεδο λεξιλόγιο των ΣΠ του Z39.50, το μεγαλύτερο μέρος από την υπάρχουσα δομική και σημασιολογική πληροφορία της βάσης, χάνεται κατά τις επερωτήσεις. Από την έρευνα που κάναμε δεν βρήκαμε κάποια πρόταση για ένα ικανοποιητικό τρόπο μετάφρασης μεταξύ της δομημένης πληροφορίας μια πηγής από τη μια μεριά και των ΣΠ του Z39.50 από την άλλη.

Έστω ένα συγκεκριμένο παράδειγμα που περιγράφει το πρόβλημα. Συνήθως ένας αριθμός από ενέργειες συσχετίζονται με κάποιο μουσειακό αντικείμενο. Κάθε ενέργεια χαρακτηρίζεται από κάποιο πρόσωπο το οποίο την έκανε και μια χρονολογία στην οποία έλαβε χώρα. Έστω για παράδειγμα ότι χρειαζόμαστε να αναζητήσουμε τα αντικείμενα που συσχετίζονται με τον Οδυσσέα Ανδρούτσο όταν αυτός βρέθηκε στην Πελοπόννησο στις αρχές της ελληνικής κατά των Τούρκων επανάστασης το 1821. Κάτι τέτοιο εκφράζεται με την παρακάτω επερώτηση:

PersonalName="Androutsos" ***AND*** **Date**=1821

Ο χρήστης θα λάβει σαν απάντηση ένα σύνολο εγγραφών οι οποίες θα ικανοποιούν τους περιορισμούς που είχε θέσει. Τίποτα όμως δεν εγγυάται ότι τα ανακτώμενα αντικείμενα σχετίζονται με τον Οδυσσέα Ανδρούτσο και εκείνη την περίοδο. Στην πραγματικότητα ένα αντικείμενο μπορεί να χρονολογείται το 1821, αλλά η σχέση του με τον Οδυσσέα Ανδρούτσο να αποτελεί ένα μεταγενέστερο γεγονός. Οποιαδήποτε λογική διασύνδεση μεταξύ αντικειμένων, προσώπων, ημερομηνιών και γεγονότων, έχει χαθεί. Για να φανεί καλύτερα το πρόβλημα, υπενθυμίζεται, ότι στην ενότητα 2.3, η ίδια ακριβώς επερώτηση είχε χρησιμοποιηθεί για την ανάκτηση κάποιας διαφορετικής πληροφορίας. Καταλαβαίνει λοιπόν κανείς, ότι η **type-1** γλώσσα του Z39.50 είναι πολύ απλή για να εκφράσει την δομική πολυπλοκότητα.

Επίσης, με την παρούσα προσέγγιση, ολοκλήρωση της πληροφορίας από διαφορετικές πηγές είναι αδύνατη. Σύμφωνα με την προσέγγιση του Z39.50, για παράδειγμα, δεν υπάρχει τρόπος ανάκτησης των αντικειμένων που συσχετίζονται με τον Οδυσσέα Ανδρούτσο μαζί με τη σχετική τους βιβλιογραφία όταν η τελευταία βρίσκεται αποθηκευμένη σε μια διαφορετική πηγή πληροφοριών στο δίκτυο. Το μόνο που μπορεί να γίνει είναι η ανάκτηση των στοιχείων από κάθε πηγή και μετά η ενοποίησή τους σε ένα μοναδικό σύνολο.

4.3.4 Μετα-δεδομένα

Μια από τις ιδέες που θα μπορούσαν να προταθούν ως λύση στο παραπάνω πρόβλημα είναι η γνωστοποίηση στον client μετα-δεδομένα του server. Μια τέτοια δυνατότητα όμως δεν παρέχεται από το πρωτόκολλο.

Έστω για παράδειγμα η περίπτωση όπου ο χρήστης ξέρει τι είναι αυτό που αναζητά, δεν ξέρει όμως τον τρόπο με τον οποίο θα το αναζητήσει. Στην περίπτωση αυτή πληροφορίες για μετα-δεδομένα της πηγής θα ήταν απαραίτητες. Μια λύση, που υιοθετείται συχνά, είναι τα ΣΠ ευρείας έννοιας όπως το ΣΠ *"Any"*. Επερώτηση πάνω σε αυτό το ΣΠ είναι σαν να γίνεται επερώτηση σε όλα τα υποστηριζόμενα ΣΠ. Αν τα δεδομένα της πηγής τα διαχειρίζεται ένα παραδοσιακό ΣΔΒΔ ή ΣΠΓ, η επερώτηση στη βάση πάνω σε όλα τα πεδία θα έχει πολύ μεγάλο κόστος, τόσο σε υπολογιστική ισχύ, όσο και σε χρόνο. Για το λόγο αυτό, η προσέγγιση αυτή δεν είναι ό,τι καλύτερο.

Μια λύση θα ήταν η δυνατότητα του client να μπορεί να ανακτά πληροφορίες για τον τρόπο με τον οποίο έχουν γίνει οι αντιστοιχίσεις των ΣΠ στις δομές της πηγής. Οι αντιστοιχίσεις αυτές μπορεί να είναι πολύ απλές (μια προς μία), αλλά και πολύπλοκες (πολλά προς ένα, πολλά προς πολλά, ένα προς πολλά κλπ.), ή ακόμα, υπερβολικά σύνθετες όπου απαιτείται ο ορισμός μια όψης. Το πρόβλημα γίνεται ακόμα μεγαλύτερο όταν δεν υπάρχει πλήρης εννοιολογική αντιστοίχιση μεταξύ ενός ΣΠ και κάποιων οντοτήτων της πηγής. Σε τέτοιες περιπτώσεις οι αποφάσεις που λαμβάνει ο διαχειριστής που ορίζει τις αντιστοιχίσεις είναι σημαντικές γιατί επηρεάζουν την *ακρίβεια (precision)* και την *ανάκληση (recall)* των απαντήσεων των επερωτήσεων.

Το παραπάνω κάνει την ανάγκη γνώσης των αντιστοιχίσεων από τον server ακόμα μεγαλύτερη. Δυστυχώς όμως το πρωτόκολλο δεν προβλέπει μια τέτοια δυνατότητα.

4.3.5 Μη υποστηριζόμενα Σημεία Πρόσβασης

Από τα βασικά προβλήματα διαλειτουργικότητας, για την πρόσβαση σε ετερογενείς και κατανεμημένες πηγές πληροφοριών, είναι η περίπτωση κατά την οποία γίνεται στο Z39.50 server μια επερώτηση στην οποία περιέχεται ένα ΣΠ, μη υποστηριζόμενο από την συγκεκριμένη πηγή. Σύμφωνα με το πρωτόκολλο, η επερώτηση θα πρέπει να αποτύχει και να επιστραφεί στον client ένας κωδικός και ένα διαγνωστικό μήνυμα τα οποία θα προσδιορίζουν την αιτία της αποτυχίας της ερώτησης.

Όμως, σε ένα κατανεμημένο περιβάλλον μια τέτοια λύση δεν είναι καθόλου επιθυμητή. Μια επερώτηση η οποία στέλνεται σε πολλούς και διαφορετικούς server χρησιμοποιώντας ένα μεγάλο αριθμό ΣΠ, οι πιθανότητες για ύπαρξη μη υποστηριζόμενων από την υποκείμενη πηγή ΣΠ και κατά συνέπεια αποτυχία της ερώτησης είναι αρκετά μεγάλη. Ακόμα κι αν η επερώτηση αποτύχει για διαφορετικό ΣΠ σε κάθε πηγή, ο χρήστης δεν θα λάβει καμία απάντηση.

Για την αποφυγή μιας τέτοιας κατάστασης, η λύση που συχνά χρησιμοποιείται σήμερα είναι η παράλειψη του μη υποστηριζόμενου ΣΠ και η εκτέλεση της επερώτησης όπως θα γινόταν αν το συγκεκριμένο ΣΠ δεν υπήρχε καθόλου. Η λύση αυτή δεν είναι ό,τι καλύτερο. Το πρόβλημα είναι ότι ο client δεν λαμβάνει γνώση του γεγονότος αυτού, και έτσι δεν μπορεί να καταλάβει αν το επιστρεφόμενο σε αυτόν σύνολο απαντήσεων έχει προκύψει βάσει όλων των κριτηρίων τα οποία είχε θέσει ο χρήστης στην επερώτηση, μέσω των ΣΠ, ή αν ορισμένα απ'αυτά έχουν παραλειφθεί λόγω μη υποστήριξης.

4.3.6 Επιστρεφόμενες εγγραφές

Η δομή και τα περιεχόμενα των εγγραφών που θα επιστρέφονται στον client ορίζονται από το εκάστοτε προφίλ. Μια εγγραφή αποτελείται από ένα σύνολο *πεδίων (elements)* τα οποία έχουν μια προκαθορισμένη σειρά και σημασιολογία. Βασικό πρόβλημα είναι ο τρόπος με τον οποίο το Z39.50 χειρίζεται τις κενές (NULL) τιμές και τα πεδία της εγγραφής, για τα οποία δεν υπάρχουν αντίστοιχες δομές στην πηγή. Το Z39.50 ορίζει τα μηνύματα *"Element Not exists"* και *"Element Not There"* για τις περιπτώσεις όπου για ένα πεδίο της εγγραφής δεν υπάρχει αντίστοιχη δομή στην πηγή και για την περίπτωση όπου υπάρχει η τιμή του, αλλά δεν είναι διαθέσιμη τη στιγμή εκείνη που ζητείται. Μια τέτοια προσέγγιση προκαλεί τη διακίνηση περιττής πληροφορίας στο δίκτυο.

Ένα άλλο θέμα, είναι ο χαρακτηρισμός που γίνεται στα πεδία της εγγραφής σαν

επαναλαμβανόμενα και *υποχρεωτικά*. Το πρώτο δηλώνει την δυνατότητα πολλαπλών τιμών σε ένα πεδίο της εγγραφής, όπως για παράδειγμα τα ονόματα των προσώπων που σχετίζονται με κάποιο μουσειακό αντικείμενο, και τα οποία μπορεί να είναι περισσότερα του ενός. Το δεύτερο δηλώνει ότι σε μια επιστρεφόμενη από τον server εγγραφή, το πεδίο αυτό θα πρέπει, οπωσδήποτε, να έχει να έχει τιμή. Τέτοιου είδους περιορισμοί αποτελούν πρόβλημα ειδικά σε ετερογενείς πηγές πληροφορίας όπου το σχήμα και τα δεδομένα είναι μεταβλητά, και κυρίως μη γνωστά εκ των προτέρων.

Ένα τρίτο πρόβλημα, είναι ότι ο client δεν έχει την δυνατότητα να επιλέξει ποια από τα πεδία της εγγραφής τον ενδιαφέρουν και να ζητήσει την ανάκτηση αυτών και μόνο. Αντίθετα, θα πρέπει κάθε φορά να ανακτά ολόκληρη την εγγραφή.

Τα τρία παραπάνω προβλήματα, τα σχετικά με τις επιστρεφόμενες εγγραφές, δεν θα μας απασχολήσουν στην παρούσα εργασία.

Κεφάλαιο 5

Περιγραφική Λογική και Z39.50

Για τη διαχείριση της σημασιολογικής ετερογένειας μεταξύ των ΣΠ του Z39.50 και των δομών της πηγής, καθώς και για τον αποδοτικό προσδιορισμό και υλοποίηση του τρόπου μετάφρασης, μια καλή προσέγγιση είναι η δημιουργία ενός συμβατικού μοντέλου το οποίο θα μπορεί να παριστάνει τα αποθηκευμένα στην πηγή στοιχεία, να δέχεται ερωτήσεις, να τις επεξεργάζεται και να επιστρέφει τα αποτελέσματα. Το μοντέλο που χρησιμοποιήθηκε αποτελείται από αντικείμενα, τις μεταξύ τους συσχετίσεις (ρόλους) και έννοιες. Λειτουργεί δε, σαν ένα κοινό εννοιολογικό μοντέλο που συνδέει τον κόσμο των ΣΠ με τον κόσμο της υποκείμενης πηγής. Στις περιπτώσεις που ένας Z39.50 server κατασκευάζεται πάνω από μια βάση δεδομένων, ή γενικότερα μια πηγή πληροφοριών με αυστηρή δόμηση των δεδομένων της, είναι φανερή η ανάγκη ύπαρξης του ενδιάμεσου επιπέδου μεταξύ πηγής και server, το οποίο θα κάνει τη μετατροπή από τις δομές του Z39.50 στις δομές της πηγής. Από την άλλη μεριά υπάρχουν περιπτώσεις όπου η μετάφραση αυτή είναι άμεση και δε δικαιολογείται η ύπαρξη ενός τέτοιου επιπέδου. Μια τέτοια περίπτωση είναι αυτή κατά την οποία η πηγή είναι ένα ΣΑΠ όπως τα συστήματα Mistral και Index+ τα οποία παρουσιάστηκαν στο κεφάλαιο 3. Όπως όμως αναφέρθηκε, είναι πιθανό να χρησιμοποιούνται λέξεις-κλειδιά από θησαυρούς όρων, οπότε κάποιο επίπεδο για τη διαχείριση των θησαυρών κρίνεται απαραίτητο.

Για τον ορισμό του μοντέλου αυτού, μια καλή ιδέα ήταν η χρησιμοποίηση της τεχνολογίας των Συστημάτων Παράστασης Γνώσεων (ΣΠΓ). Η *Περιγραφική Λογική*[1] *(Description Logic)* [DLNS96], από εδώ και στο εξής ΠΛ για συντομία, με τις *εννοιολογικές γλώσσες (concept languages)* οι οποίες στηρίζονται σε σύνθετους όρους και συσχετίσεις, αποτελεί ένα πολύ καλό υποψήφιο. Η ΠΛ χρησιμοποιήθηκε για τον ορισμό των

[1] http://www.dl.kr.org/dl

διαφόρων οντολογιών, καθώς και των συσχετίσεών τους με το εννοιολογικό μοντέλο της υποκείμενης πηγής, χρησιμοποιώντας ένα κοινό θεωρητικό φορμαλισμό. Κάτι τέτοιο ήταν εφικτό, στο βαθμό που η εκφραστική δυνατότητα της ΠΛ καλύπτει άλλους γνωστούς φορμαλισμούς που χρησιμοποιούνται σε βάσεις δεδομένων. Η βιβλιογραφία έχει να επιδείξει περιπτώσεις όπου η ΠΛ έχει επιτυχώς χρησιμοποιηθεί σε εφαρμογές ως γλώσσα παράστασης μοντέλων δεδομένων [BMS+91], εννοιολογικών σχημάτων [BBS88], ή ως γλώσσα επερωτήσεων [ABN92]. Επίσης, υπάρχει μια σειρά από υλοποιήσεις όπου η ΠΛ έχει χρησιμοποιηθεί προκειμένου να παρέχει μια διεπαφή βάσης γνώσης σε μια βάση δεδομένων[KSS+96],[NRL+94]. Για το χώρο των πολιτισμικών συστημάτων ο οποίος αποτελεί το βασικό πεδίο πειραματισμού της έρευνάς μας η χρήση της ΠΛ οδήγησε σε πολύ ικανοποιητικά αποτελέσματα.

Στην επόμενη ενότητα περιγράφονται γενικά τα πλεονεκτήματα της ΠΛ που την καθιστούν κατάλληλη για το σκοπό μας, καθώς επίσης και μια σειρά από επιπλέον προϊόντα δυνατότητες που παρέχει κατά τη χρήση της για τον προσδιορισμό της λειτουργίας του μεταφραστή. Στη συνέχεια γίνεται μια παρουσίαση του μοντέλου της ΠΛ και πώς αυτή εφαρμόζεται στην περίπτωση του Z39.50. Το κεφάλαιο τελειώνει με μια περιγραφή του τρόπου υλοποίησης του Z39.50 server ο οποίος στηρίζεται στην ΠΛ.

5.1 Τα πλεονεκτήματα της Περιγραφικής Λογικής

Η χρήση της ΠΛ για τον καθορισμό του τρόπου λειτουργίας του μεταφραστή παρέχει τα παρακάτω πλεονεκτήματα:

- Επιτυγχάνεται ο δηλωτικός προσδιορισμός των αντιστοιχίσεων ΣΠ και δομών της βάσης. Με άλλα λόγια δηλώνεται το τι αντιστοιχίζεται με τι και όχι το πώς. Έτσι ο χρήστης δεν χρειάζεται να έχει προγραμματιστικές και άλλες γνώσεις συγγραφής κώδικα. Επίσης, οι εννοιολογικές γλώσσες είναι περισσότερο κοντά στη φυσική γλώσσα και χειρίζονται ευκολότερα από τον απλό χρήστη, σε αντίθεση με τις παραδοσιακές γλώσσες επερωτήσεων των βάσεων δεδομένων.

- Τόσο η γλώσσα αντιστοιχίσεων, όσο και τα πρωτογενή στοιχεία της γλώσσας επερωτήσεων του Z39.50 (query primitives), χειρίζονται με ομοιόμορφο τρόπο. Οι δηλωμένες αντιστοιχίσεις τηρούνται και χειρίζονται από τον server σαν πρωταρχικά δομικά του στοιχεία (first class citizens). Δεν αποτελούν δηλαδή κάποιο επιπρόσθετο κομμάτι αλλά ένα δομικό του στοιχείο. Οι επερωτήσεις μπορούν να παρασταθούν και να διαχειριστούν κι αυτές σαν έννοιες.

- Λόγω της σαφούς σημασιολογίας των τελεστών της ΠΛ, μας επιτρέπει να χειριστούμε ένα μεγάλο αριθμό προβλημάτων του πρωτοκόλλου.

- Ο χρήστης που ορίζει τις αντιστοιχίσεις δεν είναι απαραίτητο να έχει γνώση της συντακτικής δόμησης των δεδομένων στην πηγή, αλλά μονάχα του εννοιολογικού τους σχήματος.

- Η ΠΛ δεν επηρεάζεται ούτε εξαρτάται από το είδος της υποκείμενης πηγής, και έτσι μπορεί να χρησιμοποιηθεί σε περιπτώσεις όπου η υποκείμενη πηγή μπορεί να είναι μια σχεσιακή [Bre94], [BB93] βάση δεδομένων, οντοκεντρική [RLNR95], [BSV95], ή ακόμα και μια εφαρμογή αρχειοθέτησης κειμένων SGML [GR90].

  Παρέχει δε παράλληλα, ένα ομοιόμορφο τρόπο για το δυναμικό καθορισμό της λειτουργίας του μεταφραστή, ανεξάρτητα από το είδος της υποκείμενης πηγής. Επίσης, έχει το πλεονέκτημα έναντι άλλων γλωσσών (π.χ. SQL) ότι είναι περισσότερο κοντά στη φυσική γλώσσα και περιγράφει καλύτερα εννοιολογικές αντιστοιχίσεις, ενώ από την άλλη μεριά η SQL περιγράφει καλύτερα δομικές (structural) αντιστοιχίσεις.

Εκτός από τα παραπάνω, η χρήση της ΠΛ για τον προσδιορισμό του τρόπου μετάφρασης, παρέχει ένα αριθμό από ενδιαφέροντα επιπλέον προϊόντα-δυνατότητες.

- Δόμηση του επιπέδου σχήματος των ΣΠ του Z39.50.
  Με τον ίδιο τρόπο με τον οποίο ορίζονται οι αντιστοιχίσεις μεταξύ ΣΠ και δομών της πηγής, μπορούν να συσχετιστούν ΣΠ μεταξύ τους και τελικά το επίπεδο σχήμα των ΣΠ, να αποκτήσει κάποια συγκεκριμένη εννοιολογική δομή, σε μια ή περισσότερες τοπικές πηγές.

- Έλεγχος συνέπειας (Consistency Validation).
  Η ΠΛ παρέχει τη δυνατότητα ελέγχου της συνέπειας παράστασης των ΣΠ στα πεδία της πηγής. Κάτι τέτοιο μπορεί να αποτελέσει μέτρο καταλληλότητας του προφίλ για το συγκεκριμένο είδος δεδομένων που περιέχονται στην πηγή.

- Βελτίωση της διαδικασίας απάντησης επερωτήσεων.
  Με τη χρήση της ΠΛ, τα ΣΠ που δεν υποστηρίζονται μπορούν να χειριστούν με ειδικό τρόπο ώστε επερωτήσεις που περιέχουν τέτοια ΣΠ να μην αποτυγχάνουν.

- Ενημέρωση για την ποιότητα των απαντήσεων (Quality Assessed Retrieval).
  Οι εννοιολογικές εκφράσεις της ΠΛ με τις οποίες καθορίζονται οι αντιστοιχίσεις μπορούν και αυτές να ανακτηθούν με μεθόδους αντίστοιχες αυτών των δεδομένων, και έτσι ο χρήστης να έχει μια εκ των προτέρων γνώση του τρόπου μετάφρασης.

- Ταξινόμηση επερωτήσεων (Taxonomy of queries).
  Οι επερωτήσεις και τα αποτελέσματά τους μπορούν να παρασταθούν ως έννοιες και να ταξινομηθούν σε μια ιεραρχία. Έτσι, γίνεται δυνατή η χρησιμοποίηση συνόλων απαντήσεων προγενέστερων επερωτήσεων σε μεταγενέστερες, καθώς επίσης και κάποιας μορφής βελτίωση της διαδικασίας απάντησης επερωτήσεων.

- Υποστήριξη πολλαπλών προφίλ.
  Ο τρόπος με τον οποίο τα ΣΠ αντιστοιχώνται στις δομές της βάσης καθορίζεται από το εκάστοτε προφίλ. Ωστόσο, πολλά ΣΠ είναι κοινά σε πολλά προφίλ με την ίδια ακριβώς σημασιολογία. Με την ΠΛ, μπορεί να οριστεί μία αντιστοίχιση για κάθε ΣΠ, ανεξάρτητα από το εκάστοτε προφίλ. Αυτό δίνει τη δυνατότητα στον server να υποστηρίζει πολλαπλά προφίλ και επομένως να μπορεί να δεχτεί επερωτήσεις από όσο το δυνατό περισσότερους client.

5.2 Η Περιγραφική Λογική

5.2.1 Περιγραφή του μοντέλου

Η θεωρία της ΠΛ είναι άμεσα συσχετισμένη με το μοντέλο δεδομένων που χρησιμοποιείται και το οποίο είναι αρκετά γενικό. Η ΠΛ αποτελεί ένα φορμαλισμό για παράσταση γνώσης και συλλογισμών. Προήλθε από το έργο KL-ONE [BS85], και παρέχει γλώσσες οι οποίες διαχειρίζονται *έννοιες, ρόλους* και *ανεξάρτητα αντικείμενα.*

Ορισμός 1 *Ένα* **Αντικείμενο (αγγ. Object)** *ορίζεται ως η ηλεκτρονική παράσταση μιας οντότητας του πραγματικού κόσμου, στο σύστημα. Κάθε αντικείμενο είναι μοναδικό και χαρακτηρίζεται από κάποιο μοναδικό κωδικό που το διαφοροποιεί από όλα τα υπόλοιπα. Το σύνολο των αντικειμένων θα συμβολίζεται με* Δ.

Σύνολα αντικειμένων του πραγματικού κόσμου με κάποια κοινή σημασιολογία ή ιδιότητα, ομαδοποιούνται σε έννοιες (ή αλλιώς κλάσεις). Κάθε έννοια του μοντέλου παριστάνει μια έννοια του πραγματικού κόσμου. Οι όροι ”έννοια” και ”κλάση” από εδώ και στο εξής θα χρησιμοποιούνται ισοδύναμα. Για παράδειγμα, η έννοια ”GOLD-GREEK-ARTIFACTS” περιγράφει το σύνολο των μουσειακών αντικειμένων τα οποία είναι φτιαγμένα από χρυσό και είναι ελληνικής προελεύσεως. Στα αντικείμενα προσδίδονται ιδιότητες, με την κατάταξή τους κάτω από έννοιες (concept membership), (π.χ. η ”Mona Lisa” είναι ”Painting”), και συνδέονται τα με άλλα αντικείμενα μέσω ρόλων (π.χ. ”Mona

Lisa" είναι "made_from" "DaVinci"). Ένα αντικείμενο το οποίο ανήκει σε μια έννοια, λέγεται ότι αποτελεί *περίπτωση (instance)* αυτής.

Ορισμός 2 *Σαν ρόλος (role) ορίζεται μια αντιστοίχιση από τα αντικείμενα στα αντικείμενα. Κάθε ρόλος έχει ένα χαρακτηριστικό όνομα, μια αρχή και ένα τέλος. Πεδίο τιμών είναι το* $\Delta \times \Delta$.

Οι ρόλοι παριστάνουν συσχετίσεις μεταξύ αντικειμένων. Για παράδειγμα ο ρόλος "made-off" μπορεί να ενώνει ένα μουσειακό αντικείμενο με κάποιο υλικό και να δηλώνει το υλικό, από το οποίο είναι κατασκευασμένο το αντικείμενο.

Οι γλώσσες οι οποίες διαχειρίζονται έννοιες, ρόλους και αντικείμενα, ονομάζονται εννοιολογικές γλώσσες (concept languages), και οι οποίες εν γένει στηρίζονται στη λογική πρώτης τάξης.

Στις επόμενες ενότητες παρουσιάζονται οι βασικές αρχές σχετικές με τις εννοιολογικές γλώσσες, τις βάσεις γνώσεων που κατασκευάζονται από αυτές, καθώς και μεθόδους για εξαγωγή συμπερασμάτων, από τέτοιου είδους βάσεις, στηριζόμενοι στην ΠΛ.

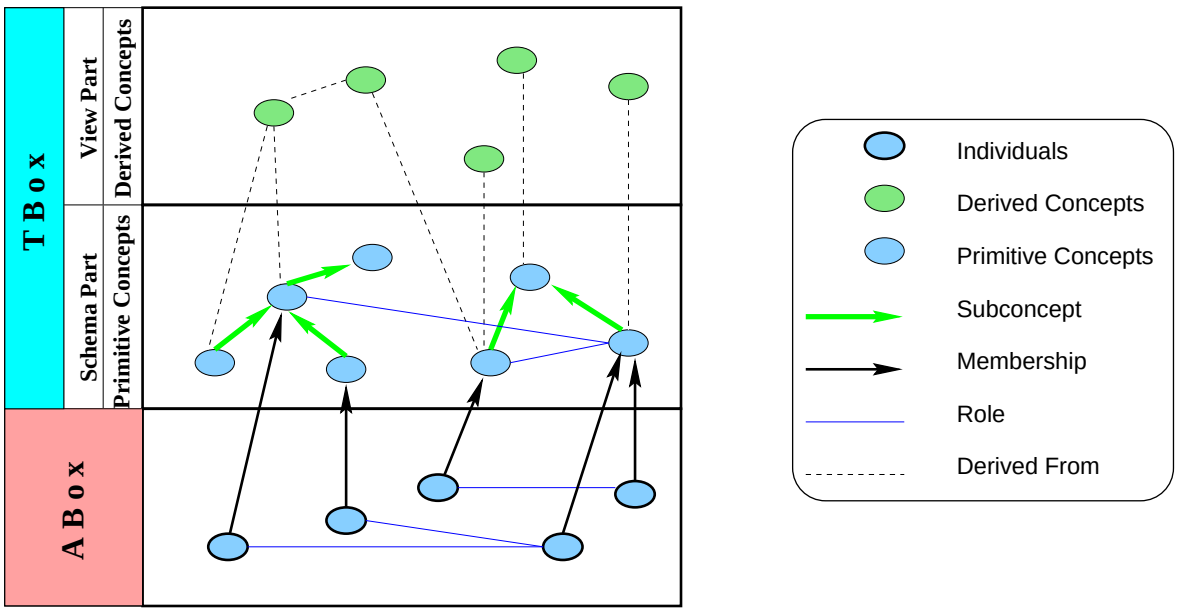
5.2.2 Το γενικό πλαίσιο της Περιγραφικής Λογικής

Μια βάση γνώσης (συνήθως θα παριστάνεται με το σύμβολο Σ), η οποία κατασκευάζεται με χρήση της ΠΛ, αποτελείται από δύο τμήματα. Το πρώτο τμήμα περιέχει τις περιγραφές εννοιών δηλαδή την εντασιακή γνώση (*Intensional Knowledge*) και ονομάζεται *TBox*. Το δεύτερο περιέχει τα πραγματικά δεδομένα δηλαδή την εκτασιακή γνώση (*Extensional Knowledge*) και ονομάζεται *ABox* [Bor95].

5.2.2.1 TBox - Εντασιακή (Intensional) γνώση

Ακολουθώντας την προσέγγιση που παρουσιάστηκε στο [BDNS94], χωρίζουμε το TBox σε δύο μέρη. Το πρώτο θα αποκαλείται *μέρος σχήματος* (scheme part) και περιέχει τις λεγόμενες *αρχικές έννοιες* ενώ το δεύτερο, αποκαλείται *τμήμα όψεως (view part)* και περιέχει τις λεγόμενες *παράγωγες έννοιες*. Η διαμέριση αυτή φαίνεται στο σχήμα 5.1.

Ορισμός 3 Αρχικές έννοιες (primitive concepts) *ονομάζονται οι βασικές έννοιες οι οποίες είναι ανεξάρτητες, αυτόνομες, δεν μπορούν να αναγνωριστούν από εξωτερικές ιδιότητές τους, ή να παραχθούν από άλλες. Για να γίνει αντιληπτή η ύπαρξη τους θα πρέπει να έχουν δηλωθεί ρητώς. Επίσης, τα αυτόνομα αντικείμενα (γεγονότα) θα πρέπει και αυτά με τη σειρά τους να εισαχθούν ρητώς ως μέλη τέτοιων εννοιών [Bor95].*



Σχήμα 5.1: Αρχικές έννοιες, παράγωγες, και αντικείμενα

Η ΠΛ, μέσω της εννοιολογικής γλώσσας, παρέχει τη δυνατότητα περιγραφής και ορισμού νέων εννοιών.

Ορισμός 4 *Δοθείσας μιας εννοιολογικής γλώσσας $\mathcal{L}$, οι έννοιες οι οποίες παράγονται μέσω των τελεστών της $\mathcal{L}$, και των αρχικών εννοιών, ονομάζονται* **παράγωγες (derived)**.

Ο ορισμός νέων εννοιών γίνεται, προσδιορίζοντας τις ικανές ή τις ικανές και αναγκαίες συνθήκες για τη συμμετοχή ενός αντικειμένου στις περιπτώσεις κάποιας έννοιας. Ανάλογος είναι και ο ορισμός των ρόλων.

Ο ορισμός του TBox γίνεται μέσω μιας σειράς αξιωματικών δηλώσεων οι οποίες μπορούν να πάρουν κάποια από τις παρακάτω μορφές:

1. **Προδιαγραφή αρχικής έννοιας (Primitive Concept Specification)**

$$A \;\dot{\leq}\; C$$

που δηλώνει ρητώς ότι η έννοια **A** αποτελεί υπάλληλη έννοια της **C**, δηλαδή αντικείμενα που είναι μέλη της **A** είναι αναγκαστικά και μέλη της **C**.

2. **Ορισμός παράγωγης έννοιας (Concept Definition)**

$$\mathrm{A} \;\doteq\; C$$

η οποία ορίζει την παράγωγη έννοια **A** και τον τρόπο παραγωγής της, μέσω της έννοιας **C**.

3. **Ορισμός Ρόλου (Role Definition)**

$$R \;\dot{\leq}\; C \times D$$

που δηλώνει ότι ο ρόλος **R** είναι μια αντιστοίχηση από αντικείμενα της έννοιας **C** σε αντικείμενα της **D**. Ο ορισμός του ρόλου είναι διαφοροποιημένος από τον ορισμό της καθιερωμένης ΠΛ ο οποίος θεωρεί ότι: $\mathbf{R} \dot{\leq} \Delta \times \Delta$ [Sch94].

Οι ακόλουθες παρατηρήσεις είναι αναγκαίες:

- Μια αρχική έννοια ορίζει τις ικανές συνθήκες ώστε ένα αντικείμενο να αποτελεί μέλος της. Οι έννοιες οι οποίες ορίζονται στις βάσεις δεδομένων, για παράδειγμα, είναι όλες αρχικές έννοιες. Αυτό γιατί για να ανήκει ένα αντικείμενο στην έννοια αυτή θα πρέπει να έχει εισαχθεί από το χρήστη και να έχει δηλωθεί ρητώς ότι ανήκει στην έννοια αυτή. Όταν για παράδειγμα ορίζεται η έννοια Date, η βάση δεδομένων δεν είναι σε θέση από μόνη της να αποφασίσει αν ένα αντικείμενο αποτελεί ή όχι μέλος της έννοιας αυτής, παρά μόνο αν κάτι τέτοιο έχει δηλωθεί ρητώς από το χρήστη.

  Από τις τρεις παραπάνω μορφές οι (1) και (3) περιγράφουν ικανές συνθήκες. Η πληροφορία ότι ένα αντικείμενο αποτελεί μέλος της έννοιας που περιγράφεται στο αριστερό μέλος, είναι αρκετή για την εξαγωγή του συμπεράσματος ότι το αντικείμενο αυτό ανήκει στην έννοια που περιγράφεται στο δεξιό μέλος. Δεν ισχύει όμως το αντίστροφο.

- Αντίθετα, μια παράγωγη έννοια ορίζει τις αναγκαίες και ικανές συνθήκες προκειμένου ένα αντικείμενο να ανήκει σε αυτή. Για παράδειγμα η έννοια DateOfCreation ορίζεται σαν η ημερομηνία η οποία περιγράφει το γεγονός μιας δημιουργίας. Αν δηλαδή ένα αντικείμενο ημερομηνίας περιγράφει ένα γεγονός δημιουργίας, το σύστημα, έχοντας υπόψη του τον ορισμό αυτό, είναι σε θέση να συμπεράνει από μόνο του ότι το αντικείμενο αυτό αποτελεί μέλος της έννοιας DateOfCreation.

  Σε αντίθεση με τις (1) και (3), η (2) περιγράφει ικανή και αναγκαία συνθήκη, δηλαδή η πληροφορία ότι ένα αντικείμενο ανήκει στην έννοια που περιγράφεται από κάποιο μέλος, συνεπάγεται ότι το αντικείμενο αυτό ανήκει και στην έννοια

που περιγράφεται από το άλλο μέλος. Αυτή η παρατήρηση αποτελεί τη βάση για την εκτέλεση συλλογισμών πάνω στο σχήμα και στα δεδομένα της βάσης γνώσης, όπως θα εξηγηθεί παρακάτω.

- Η πρώτη φορά που μια έννοια **A** θα εμφανιστεί στο αριστερό μέλος μιας δήλωσης του TBox (TBox-statement), θα αποκαλείται *εισαγωγή (introduction)* της **A**. Κάθε έννοια έχει το πολύ μία εισαγωγή.

- Το δεξιό μέλος κάποιας από τις (1), (2) και (3) μπορεί να είναι από μια αρχική έννοια έως μία σύνθετη εννοιολογική έκφραση της γλώσσας $\mathcal{L}$. Μια έννοια που εμφανίζεται στο αριστερό μέλος μιας έκφρασης προδιαγραφών (δηλ. της μορφής (1)) είναι πάντοτε μια *αρχική* έννοια. Η έκφραση προδιαγραφών στην οποία η έννοια του αριστερού της μέλους εμφανίζεται πρώτη φορά, θα θεωρείται ο *ορισμός της αρχικής έννοιας* αυτής. Οι αρχικές έννοιες οι οποίες δεν εμφανίζονται στο αριστερό μέλος καμίας έκφρασης του TBox, ονομάζονται *ατομικές* και θεωρείται ότι εισάγονται από μια έκφραση της μορφής[2]:

$$A \dot{\leq} \top$$

  Μια έννοια που έχει χρησιμοποιηθεί στο αριστερό μέλος κάποιου ορισμού θα αποκαλείται *ορισμένη (defined).*

- Το εννοιολογικό σχήμα (schema part) ορίζεται μέσω μιας σειράς προδιαγραφών-δηλώσεων της μορφής (1) και (3) μόνο, ενώ το μέρος όψεως (view part) ορίζεται μέσω δηλώσεων της (2) μόνο.

- *Κυκλικοί ορισμοί* στην παρούσα προσέγγιση δεν είναι επιτρεπτοί. Σύμφωνα με τον Nebel, όπως γράφει στο [Neb91], ένας *κύκλος* ορίζεται ως εξής:

  > *Μια έννοια A* χρησιμοποιεί άμεσα *μια άλλη έννοια B αν και μόνο αν η B εμφανίζεται στην εισαγωγή της A. Μια έννοια A_0 χρησιμοποιεί την έννοια A_n αν υπάρχει μια αλυσίδα $A_0, A_1, ..., A_n$ τέτοια ώστε η A_i να χρησιμοποιεί την A_{i+1}, $0 \leq i \leq n-1$. Τέλος, λέγεται ότι το TBox περιέχει ένα* κύκλο, *αν μια έννοια* χρησιμοποιεί *τον εαυτό της.*

- Ανάλογη σχέση με την (1) μπορεί να υπάρξει και μεταξύ ρόλων. Κάτι τέτοιο όμως δεν έχει ενδιαφέρον για την παρούσα εργασία και για το λόγο αυτό, θα ισχύει η θεώρηση ότι όλοι οι ρόλοι είναι ατομικοί.

[2] Η έννοια $\top$ θα παρουσιαστεί παρακάτω μαζί με τους υπόλοιπους τελεστές της ΠΛ

5.2.2.2 ABox - Εκτασιακή (Extensional) γνώση

Το ABox περιέχει τα γεγονότα της βάσης γνώσεων. Δύο είναι οι βασικοί τύποι δηλώσεων-αξιωμάτων για την κατασκευή του ABox. Οι τύποι αυτοί επιτρέπουν συσχέτιση αντικειμένων με έννοιες-κλάσεις του TBox όσο και συσχετίσεις αντικειμένων μεταξύ τους μέσω ρόλων. Οι δύο αυτοί τύποι δηλώσεων είναι οι εξής:

- **Δήλωση συμμετοχής αντικειμένου σε έννοια (Concept Membership Assertion)**: Συμβολίζεται **C(a)**, και δηλώνει ότι το αντικείμενο *a* αποτελεί μέλος της έννοιας *C*
- **Δήλωση συμμετοχής αντικειμένου σε ρόλο (Role Membership Assertion)**: συμβολίζεται **R(a,b)**, και δηλώνει ότι τα αντικείμενα *a* και *b* συνδέονται μέσω του ρόλου *R*.

Ένα ABox είναι μια πεπερασμένη σειρά από δηλώσεις (statements) των δύο παραπάνω τύπων [Sch94].

5.2.3 Ερμηνεία

Γενικά, σε οποιαδήποτε τυπική λογική, ο κόσμος αποτελείται από ένα σύνολο από αντικείμενα που αποκαλείται *πεδίο (domain)* ή *σύμπαν αναφοράς (universe of discourse)*, καθώς και μια σειρά από συσχετίσεις και λειτουργίες σε αυτό. Η σημασιολογία των εννοιών και των ρόλων στο σύμπαν αυτό, καθορίζεται με χρήση της έννοιας της *ερμηνείας (interpretation)* $\mathcal{I}$.

Ορισμός 5 *Σαν* **Ερμηνεία (Interpretation)** *[Bor96]* $\mathcal{I}$=<$\mathcal{I}(\Delta)$ $\mathcal{I}(\cdot)$>, *ορίζεται ένα σύνολο από τιμές* $\mathcal{I}(\Delta)$ *και μια συνάρτηση* $\mathcal{I}(\cdot)$ *η οποία αντιστοιχίζει:*

- *Κάθε έννοια σε ένα υποσύνολο τού* $\mathcal{I}(\Delta)$
- *Κάθε ρόλο σε μια δυαδική συνάρτηση που ανήκει στο* $\mathcal{I}(\Delta)\times\mathcal{I}(\Delta)$
- *Κάθε ανεξάρτητο αντικείμενο σε ένα στοιχείο του* $\mathcal{I}(\cdot)$ *[Wil95].*

Η ερμηνεία μιας εννοιολογικής έκφρασης ορίζεται με βάση την ερμηνεία των τελεστών και των εννοιών που αποτελούν την έκφραση.

Μια ερμηνεία $\mathcal{I}$= <$\mathcal{I}(\Delta)$, $\mathcal{I}(\cdot)$> ικανοποιεί τις παρακάτω δηλώσεις σύμφωνα με το [Sch94]:

- $C_1 \dot{\leq} C_2$ αν και μόνο αν $\mathcal{I}(C_1) \subseteq \mathcal{I}(C_2)$.

- $A \doteq C$ αν και μόνο αν $\mathcal{I}(A) = \mathcal{I}(C)$
- $R \dot{\leq} C_1 \times C_2$ αν και μόνο αν $\mathcal{I}(R) \subseteq \mathcal{I}(C_1) \times \mathcal{I}(C_2)$
- $C(a)$ αν και μόνο αν $\mathcal{I}$(a) $\subseteq \mathcal{I}$(C)
- $R(a, b)$ αν και μόνο αν ($\mathcal{I}$(a), $\mathcal{I}$(b))$\subseteq \mathcal{I}$(R)

5.2.4 Βάση Γνώσης

Ορισμός 6 *Δοθείσας μιας γλώσσας $\mathcal{L}$, μια βάση γνώσης εννοιών (Concept-Based Knowledge Base ή CBKB για συντομία), θα αποκαλείται η τριάδα $\Sigma = <\mathcal{T}, \mathcal{A}, \mathcal{L}>$, όπου $\mathcal{T}$ και $\mathcal{A}$, ένα TBox και ένα ABox αντίστοιχα.*

Ορισμός 7 *Μια ερμηνεία $\mathcal{I}$=<$\mathcal{I}(\Delta)$, $\mathcal{I}(\cdot)>$ λέγεται ότι αποτελεί* μοντέλο (model) *του Σ αν κάθε αξίωμα του Σ ικανοποιείται από το $\mathcal{I}$. Το Σ λέγεται* ικανοποιημένο *αν υπάρχει έστω και ένα μοντέλο αυτού.*

Ειδικότερα, σύμφωνα με το [Wil95], ισχύει ότι:

- Μια ερμηνεία $\mathcal{I}$ αποτελεί μοντέλο ενός TBox $\mathcal{T}$ αν και μόνο αν η $\mathcal{I}$ ικανοποιεί όλα τα εννοιολογικά αξιώματα στο $\mathcal{T}$.
- Μια ερμηνεία $\mathcal{I}$ αποτελεί μοντέλο ενός ABox $\mathcal{A}$, ως προς το TBox $\mathcal{T}$, αν και μόνο αν η $\mathcal{I}$ ικανοποιεί όλα τα δηλωτικά (assertion) αξιώματα στο $\mathcal{A}$, και όλα τα εννοιολογικά (terminological) αξιώματα στο $\mathcal{T}$.

5.2.5 Σχέση Υπαγωγής

Η σχέση $\dot{\leq}$ αποκαλείται *υπαγωγή (subsumption).* Στο [Wil95] η υπαγωγή ορίζεται ως:

Ορισμός 8 *Έστω δύο έννοιες C_1 και C_2, και έστω $\mathcal{T}$ ένα TBox. Λέγεται ότι η C_1 υπάγεται στην C_2, σύμφωνα με το $\mathcal{T}$ και συμβολίζεται $C_1 \dot{\leq} C_2$, αν και μόνο αν $\mathcal{I}(C_1) \subset \mathcal{I}(C_2)$ για κάθε μοντέλο του $\mathcal{T}$.*

Η σχέση $\dot{\leq}$ ορίζει μια σχέση μερικής διάταξης. Αν για δύο έννοιες, ισχύει το ότι η μία *υπάγει* την άλλη, τότε οι δύο αυτές έννοιες θα θεωρούνται *ισοδύναμες (equivalent)* Δηλαδή: $C_1 \equiv C_2 \Leftrightarrow C_1 \dot{\leq} C_2 \wedge C_2 \dot{\leq} C_1$ [BM95].

Η υπαγωγή για ρόλους μπορεί να οριστεί ανάλογα. Λέγεται επίσης ότι μια βάση γνώσης Σ λογικά συνεπάγεται μια δήλωση α του TBox ή του ABox, και γράφεται $\Sigma| = \alpha$ αν το α είναι αληθές για οποιοδήποτε μοντέλο του Σ.

5.2.6 Υποθέσεις της Περιγραφικής Λογικής

Οι βάσεις δεδομένων και τα συστήματα παράστασης γνώσεων, παριστάνουν και εξάγουν συμπεράσματα για γεγονότα του πραγματικού κόσμου. Τα ΣΔΒΔ παρέχουν μια γλώσσα που χρησιμοποιείται για τον ορισμό του σχήματος, και για λειτουργίες πάνω στα δεδομένα. Η γλώσσα αυτή ονομάζεται *γλώσσα χείρισμού δεδομένων (data manipulation language)*. Οι γλώσσες του είδους αυτού στηρίζουν τη λειτουργία τους, πάνω σε ορισμένες προϋποθέσεις σύμφωνα με το [Hus94]: **Στη μοναδικότητα του ονόματος (Unique name assumption)**, **στην Υπόθεση κλειστού κόσμου (Closed world assumption)**, και στην **Υπόθεση της κλειστού πεδίου (Closed-domain assumption)**

Από την άλλη μεριά, οι υποθέσεις πάνω στις οποίες στηρίζονται οι γλώσσες επερωτήσεων και διαχείρισης γεγονότων των Συστημάτων Παράστασης Γνώσης και Συλλογισμών (ΣΠΓΣ), συμπίπτουν μόνο στην υπόθεση της μοναδικότητας του ονόματος. Κατά τα άλλα, ισχύει η:

- **Υπόθεση του ανοικτού κόσμου (Open World Assumption)**,
  κατά την οποία μπορεί να υπάρχουν γεγονότα τα οποία να είναι αληθή και να μην περιέχονται στη βάση γνώσης, και η

- **Υπόθεση της ανοικτού πεδίου (Open domain assumption)**,
  σύμφωνα με την οποία μπορεί να υπάρξουν περισσότερα αντικείμενα στον κόσμο από όσα βρίσκονται αποθηκευμένα στη βάση.

Οι διαφορετικές αυτές υποθέσεις δηλώνουν ότι ακόμα και αν τα δεδομένα ενός ΣΔΒΔ και ενός ΣΠΓΣ συμπίπτουν, τα αποτελέσματα των ερωτήσεων σε αυτά θα είναι εντελώς διαφορετικά. Αναπόφευκτο γεγονός είναι ότι τόσο οι βάσεις δεδομένων όσο και τα συστήματα γνώσης, περιέχουν μόνο τμήμα της γνώσης του κόσμου, και έτσι δεν είναι ξεκάθαρο το τι είναι γνωστό και τι όχι.

Η δυνατότητα παράστασης μη πλήρους γνώσης, αποτελεί το βασικό χαρακτηριστικό που διαφοροποιεί τα συστήματα γνώσης από τις βάσεις δεδομένων. Η σημασιολογία των εννοιολογικών γλωσσών στηρίζεται στην υπόθεση του ανοικτού κόσμου. Έτσι η απάντηση σε μια επερώτηση Q θα είναι "Yes" αν η Q είναι αληθής για όλα τα μοντέλα του Σ, "No" αν είναι ψευδής για όλα τα μοντέλα, και "Unknown" στις υπόλοιπες περιπτώσεις. Στο [Bre95a] υποστηρίζεται ότι μια βάση γνώσης η οποία είναι πλήρης υπό την έννοια αυτή μπορεί να συμπεριφερθεί σαν μια βάση δεδομένων, και οι εννοιολογικές γλώσσες της ΠΛ να χρησιμοποιηθούν σαν γλώσσες επερωτήσεων.

Για να επιτευχθεί μια τέτοια δυνατότητα εισάγεται ένας νέος *επιστημολογικός τελεστής (epistemic operator)*, με τον τρόπο που ορίζεται στο [DLNN94], ο οποίος διαχωρίζει τη γνώση που αναφέρεται σε ολόκληρο τον κόσμο (world assumption), από τη γνώση που περιέχεται στη βάση (knowledge of the state of the base). Με προσεκτική χρήση του τελεστή αυτού, είναι δυνατόν να εκφράσει κανείς επερωτήσεις, η επεξεργασία των οποίων αναγκάζει το σύστημα να υποθέσει πλήρη γνώση για τη βάση γνώσης. Η παρατήρηση εδώ είναι ότι η προσέγγιση αυτή είναι διαφορετική από το να υποθέσει κανείς *κλειστή γνώση* για τη βάση. Η κλειστότητα δεν αφορά τη σημασιολογία της βάσης, αλλά κατά την εκτέλεση των επερωτήσεων, τα είδη των συλλογισμών που λαμβάνουν χώρα, έχουν τα ίδια αποτελέσματα με αυτά που θα είχαμε αν εάν ίσχυε η υπόθεση του κλειστού κόσμου. Αυτό πράγματι συμβαίνει στις κατανεμημένες, ετερογενείς και ομόσπονδες βάσεις. Κανείς δεν υποθέτει ότι κάποια βάση έχει πλήρη γνώση του κόσμου, αλλά κατά την εκτέλεση των επερωτήσεων, απαντήσεις της βάσης εξάγονται όπως εάν ίσχυε κάτι τέτοιο.

Ο *epistemic* αυτός τελεστής συμβολίζεται με το σύμβολο **"K"** (known) και εισάγεται μπροστά από έννοιες και ρόλους. Ο συμβολισμός $\mathbf{K}C$ δηλαδή, παριστάνει τα αντικείμενα τα οποία είναι γνωστό ότι ανήκουν στην C, δηλαδή όσα είναι αποθηκευμένα στη βάση μια συγκεκριμένη χρονική στιγμή, και ανήκουν στην *έκταση (extension)* της C. Η χρήση του τελεστή αυτού αποτελεί μια μορφή της υπόθεσης του κλειστού κόσμου.

Σύμφωνα με το [Sch94], όσον αφορά στις ερμηνείες των διαφόρων εννοιών, ισχύουν οι υποθέσεις:

- Κάθε ερμηνεία ορίζεται πάνω σε ένα σταθερό πεδίο, που ονομάζεται Δ.

- Για κάθε ερμηνεία, η αντιστοίχηση από τα ανεξάρτητα αντικείμενα στα αντικείμενα του Δ, είναι σταθερή και αποκαλείται γ.

Έτσι, το $\mathbf{K}C$ θα ερμηνεύεται σαν το σύνολο των αντικειμένων που είναι γνωστό ότι αποτελούν περιπτώσεις της C, σε οποιαδήποτε ερμηνεία. Αυτό έχει σαν αποτέλεσμα, για κάθε βάση Σ να θεωρείται η ύπαρξη ενός και μοναδικού μοντέλου. Από εδώ και στο εξής σαν *ερμηνεία* των διαφόρων εννοιών και ρόλων, θα θεωρείται πάντα αυτή που θα ήταν με τη χρήση του παραπάνω τελεστή, ο οποίος όμως, για λόγους ευκολίας, δεν θα αναφέρεται.

5.2.7 Σύνταξη και σημασιολογία των εννοιολογικών γλωσσών

Οι εννοιολογικές γλώσσες χρησιμοποιούνται στη δημιουργία εννοιολογικών εκφράσεων, προκειμένου να οριστούν νέες παράγωγες έννοιες, καθώς και οι συσχετίσεις τους. Μια εννοιολογική γλώσσα αποτελείται από τα παρακάτω βασικά συντακτικά σύμβολα όπως αναφέρονται στο [BMS+91]:

- **Έννοιες (concepts)**, το σύνολο των οποίων θα συμβολίζεται $\mathcal{C}$.
- **Ρόλους (roles)**, που συμβολίζονται με $\mathcal{RN}$.
- **Ανεξάρτητα αντικείμενα (individual objects)**, το σύνολο των οποίων είναι το Δ.
- **Τελεστές παραγωγής εννοιών και ρόλων (concept & role forming operators)**.

Από εδώ και στο εξής, εκτός και αν αναφέρεται ρητώς κάτι διαφορετικό, τα κεφαλαία γράμματα A, B, C, θα συμβολίζουν έννοιες, τα κεφαλαία R, P, ρόλους, και τα μικρά όπως i, j, k, ανεξάρτητα αντικείμενα.

Name	Concrete Form	Mathem. Repr/tion	Semantics
Concept Name	A	A	$\mathcal{I}(A) \subseteq \mathcal{I}(\Delta)$
Top	TOP	$\top$	Δ
Bottom	BOTTOM	$\bot$	$\emptyset$
Union	(OR C D)	$A \sqcup C$	$\{d_1 \mid d_1 \in \mathcal{I}(A) \cup \mathcal{I}(C)\}$
Intersect	(AND C D)	$A \sqcap C$	$\{d_1 \mid d_1 \in \mathcal{I}(A) \cap \mathcal{I}(C)\}$
Minus	(MINUS A C)	A - C	$\{d_1 \mid d_1 \in \mathcal{I}(A) \wedge d_1 \notin \mathcal{I}(C)\}$
Existential Quantification	(SOME R C)	$\exists R.C$	$\{d_1 \mid \exists\, d_2{:}(d_1,d_2) \in \mathcal{I}(R) \wedge d_2 \in \mathcal{I}(C)\}$
Universal Quantification	(ALL R C)	$\forall R.C$	$\{d_1 \mid \forall\, d_2{:}(d_1,d_2) \in \mathcal{I}(R) \rightarrow d_2 \in \mathcal{I}(C)\}$
At-Least Restriction	(ATLEAST n R)	$\leq n.R$	$\{d_1 \mid \#\{d_2 \mid (d_1, d_2) \in \mathcal{I}(R)\} \leq n\}$
At-Most Restriction	(ATMOST n R)	$\geq n.R$	$\{d_1 \mid \#\{d_2 \mid (d_1, d_2) \in \mathcal{I}(R)\} \geq n\}$
OneOf	(ONEOF i, j, ...)	$\{ i, j, ...\}$	$\{ i, j, ...\}$

Πίνακας 5.1: Τελεστές σύνθεσης εννοιών και η σημασιολογία τους

Στους πίνακες 5.1 και 5.2 περιγράφονται οι τελεστές εννοιών και ρόλων αντίστοιχα, καθώς και η ερμηνεία τους. Για τους πίνακες αυτούς θα πρέπει να σημειωθούν τα:

1. Ο τελεστής $\top$ χρησιμοποιείται για να δηλώσει την έννοια, η ερμηνεία της οποίας περιλαμβάνει όλα τα αντικείμενα που είναι αποθηκευμένα στη βάση, σε οποιοδήποτε στιγμιότυπό της.

Name	Concrete Form	Mathem. Repr/tion	Semantics
Role name	R	R	$\mathcal{I}(R) \subseteq \mathcal{I}(\Delta \times \Delta)$
Reverse	(REVERSE R)	R^{-1}	$\{(d_1, d_2) \mid (d_2, d_1) \in \mathcal{I}(R)$

Πίνακας 5.2: Τελεστές σύνθεσης ρόλων και η σημασιολογία τους

2. Η χρήση του τελεστή $\bot$ γίνεται σε περιπτώσεις εννοιών των οποίων η ερμηνεία σε οποιοδήποτε στιγμιότυπο της βάσης είναι το κενό σύνολο. Κάτι τέτοιο είναι χρήσιμο όπως θα φανεί παρακάτω στις περιπτώσεις όπου μια έννοια δεν παρίσταται στη βάση, που σημαίνει ότι η βάση δεν έχει καθόλου γνώση για την ερμηνεία της.

3. Τα σύμβολα A, C, C_1 και C_2 μπορεί να είναι τόσο αρχικές ή παράγωγες έννοιες, όσο και εννοιολογικές εκφράσεις.

4. Οι πράξεις ίσο, γνήσια μεγαλύτερο, ή γνήσια μικρότερο, δεν υποστηρίζονται άμεσα, μια και μπορούν να εξαχθούν από τους ήδη υπάρχοντες τελεστές $\leq$ και $\geq$ σε συνδυασμό με τους τελεστές τομής ή/και διαφοράς.

Παρακάτω περιγράφονται μερικά παραδείγματα για τον τρόπο χρήσης της ΠΛ και ειδικότερα των εννοιολογικών εκφράσεων στον ορισμό νέων εννοιών.

Παράδειγμα 1 *Χαρακτηριστικό παράδειγμα χρήσης του τελεστή Top είναι η περίπτωση της έννοιας "Any", η οποία χρησιμοποιείται σε περιπτώσεις που ο χρήστης χρειάζεται να απευθύνει μια ερώτηση στη βάση γενικά και όχι σε κάποιο συγκεκριμένο πεδίο. Οπότε η έννοια* Any *ορίζεται ως:*

$$Any \doteq \top$$

Παράδειγμα 2 *Έστω η βάση του Μουσείου Μπενάκη, στην οποία έχουν καταχωρηθεί τα αντικείμενα από τη συλλογή όπλων του μουσείου, και μόνο αυτά. Είναι προφανές ότι σε οποιοδήποτε στιγμιότυπο της βάσης, όλα τα αποθηκευμένα αντικείμενα ανήκουν στη συγκεκριμένη αυτή συλλογή. Έτσι η έννοια της* συλλογής *για τα αντικείμενα της βάσης αυτής θα έχει τη σταθερή τιμή "Benaki Museum Gun Collection". Αν όμως στις αρχικές έννοιες δεν υπάρχει ορισμένη η έννοια* Collection, *μπορεί να οριστεί δυναμικά μέσω του τελεστή* OneOf *ως εξής:*

$$Collection \doteq \{"Benaki\ Museum\ Gun\ Collection"\}$$

που σημαίνει ότι η ερμηνεία της έννοιας Collection*, έχει ένα και μόνο στοιχείο, το "Benaki Museum Gun Collection".*

Έστω τώρα η έννοια της προστασίας*. Μια τέτοια έννοια είναι για να χαρακτηρίζει ακίνητα μνημεία και όχι κινητά αντικείμενα. Έτσι, στην παραπάνω βάση μια τέτοια πληροφορία δεν θα είχε νόημα να παρίσταται. Η ερμηνεία της έννοιας* Protection Status *θα είναι πάντοτε το κενό σύνολο στο βαθμό που περιγράφει πληροφορία άσχετη με τα μουσειακά αντικείμενα που περιέχονται στη βάση. Μια τέτοια κατάσταση μπορεί να παρασταθεί στην ΠΛ ως εξής:*

$$ProtectionStatus \doteq \bot$$

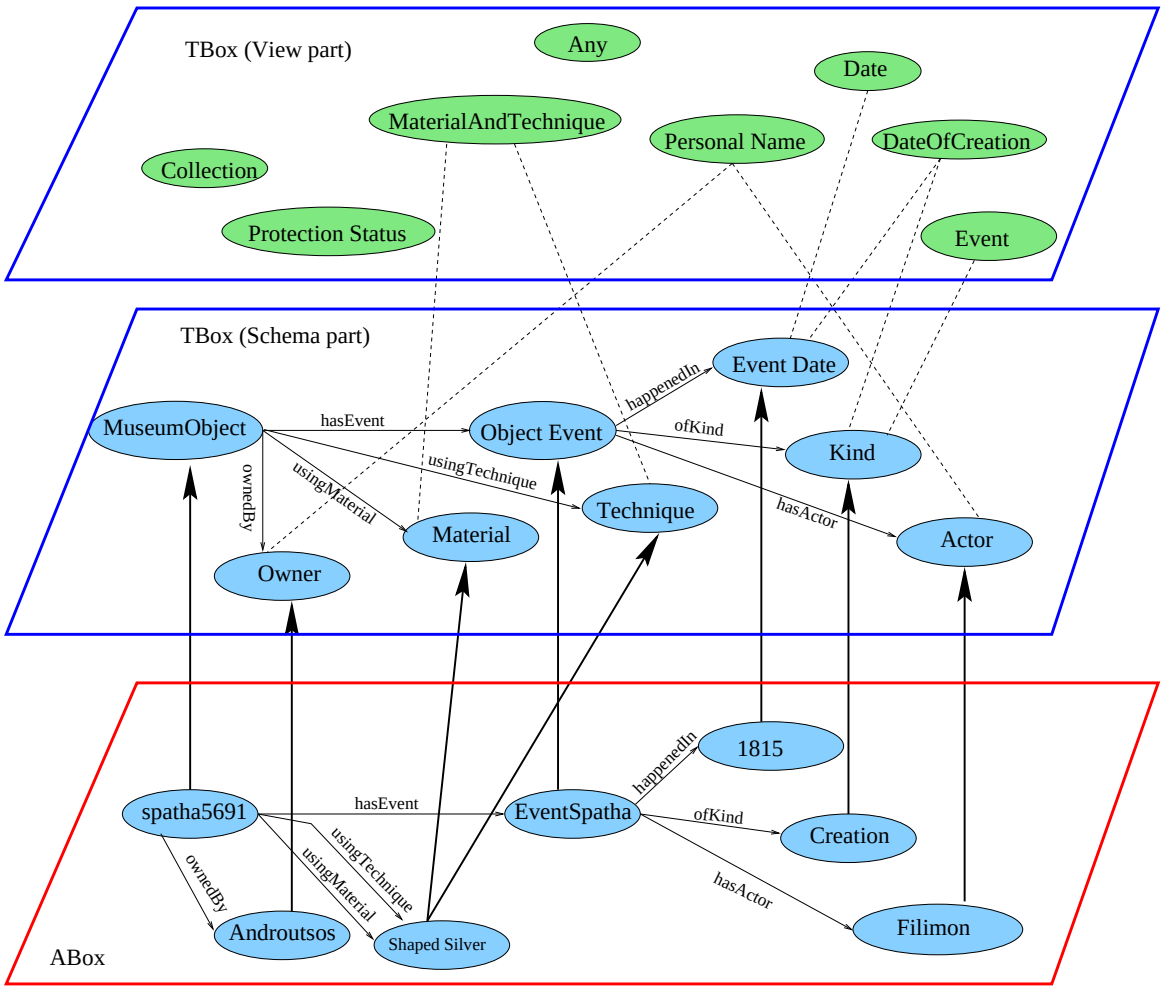
Ο τελεστής Bottom *αποτελεί ένα καλό τρόπο παράστασης ΣΠ τα οποία δεν υποστηρίζονται από την πηγή, αν και περιγράφονται στο προφίλ. Αντί να αποτυγχάνει μια ερώτηση, επειδή περιέχει ένα ΣΠ που δεν υποστηρίζεται, μια πηγή μπορεί να υποστηρίξει όλα τα ΣΠ του προφίλ, αντιστοιχώντας στην έννοια $\bot$ εκείνα για τα οποία δεν υπάρχει πληροφορία.*

Παράδειγμα 3 *Στο σχήμα 5.2 φαίνεται ένα τμήμα από μια μουσειακή βάση στην οποία περιέχονται πληροφορίες για γεγονότα τα οποία συσχετίζονται με κάποιο μουσειακό αντικείμενο, την τεχνοτροπία του, καθώς και το υλικό του. Το τμήμα σχήματος (schema part) του TBox που φαίνεται στο σχήμα, έχει προέλθει από τις παρακάτω δηλώσεις:*

$$\begin{aligned}
hasEvent &\dot{\leq} MuseumObject \times ObjEvent \\
ownedBy &\dot{\leq} MuseumObject \times Owner \\
happenedIn &\dot{\leq} ObjEvent \times EventDate \\
ofKind &\dot{\leq} ObjEvent \times Kind \\
hasActor &\dot{\leq} ObjEvent \times Actor \\
usingTechnique &\dot{\leq} MuseumObject \times Technique \\
usingMaterial &\dot{\leq} MuseumObject \times Material
\end{aligned}$$

Ορισμένες έννοιες του μέρους όψεως μπορούν να αντιστοιχιθούν κατευθείαν σε έννοιες του τμήματος σχήματος χωρίς χρήση άλλου τελεστή. Για το παράδειγμά μας, τέτοιες περιπτώσεις είναι:

$$\begin{aligned}
Event &\dot{\leq} Kind \\
Date &\dot{\leq} EventDate
\end{aligned}$$



Σχήμα 5.2: Ένα τμήμα μιας βάσης γνώσης κάποιου Μουσείου

Οι έννοιες Owner *και* Actor, *είναι οι μόνες της βάσης οι οποίες παριστάνουν πρόσωπα. Έτσι ο καλύτερος τρόπος για να παρασταθεί η έννοια* Personal Name *είναι να οριστεί σαν σύνθεση των δύο αυτών εννοιών και να κατασκευαστεί το τμήμα απόψεων (view part) του TBox που φαίνεται στο σχήμα, σύμφωνα με τη δήλωση:*

$$PersonalName \doteq Actor \sqcup Owner$$

Υπάρχουν περιπτώσεις ορισμένων όρων που περιγράφουν τόσο τεχνοτροπία, όσο και υλικό. Ένα τέτοιο παράδειγμα είναι η "χυτή τέμπερα" που αναφέρεται σε ένα πίνακα ζωγραφικής και δηλώνει παράλληλα με την τεχνική ("χυτή"), και το υλικό ("τέμπερα"). Στο παράδειγμά μας, περιγράφεται το σπαθί του Οδυσσέα Ανδρούτσου. Το ότι το σπαθί είναι "shaped silver", παριστάνει κι αυτό τόσο τεχνική, όσο και υλικό. Θα μπορούσε επομένως να παραστήσουμε την έννοια MaterialANDTechnique, *ως τα*

αντικείμενα εκείνα που ανήκουν ταυτόχρονα στις έννοιες Material *και* Technique. *Κάτι τέτοιο στην ΠΛ θα εκφράζονταν ως:*

$$MaterialAndTechnique \doteq Material \sqcap Technique$$

Παράδειγμα 4 *Έστω πάλι η βάση του σχήματος 5.2 στην οποία χρειάζεται να παραστήσουμε την έννοια "DateOfCreation". Αυτού του είδους οι ημερομηνίες είναι οι ημερομηνίες οι οποίες συνδέονται με κάποια γεγονότα τα οποία είναι τύπου "Creation".*

Η έκφραση $\{"Creation"\}$ *παριστάνει μια έννοια της οποίας η ερμηνεία περιέχει ένα και μοναδικό στοιχείο, το* "Creation", *και μπορεί να χρησιμοποιηθεί σε άλλες εννοιολογικές εκφράσεις στη θέση εννοιών, και έτσι να κατασκευαστούν νέες. Έτσι:*

$$\exists ofKind.\{"Creation"\}$$

θα παριστάνει όλα εκείνα τα αντικείμενα από τα οποία υπάρχει αρχή ρόλου με το όνομα ofKind *και πέρας το* "Creation". *Το παραπάνω κι αυτό με τη σειρά του μπορεί να χρησιμοποιηθεί σε μια νέα εννοιολογική έκφραση.*

Η έκφραση

$$\exists (happenedIn)^{-1}.\text{A}$$

περιέχει όλα εκείνα τα αντικείμενα στα οποία καταλήγει το πέρας κάποιου ρόλου (γι' αυτό εξάλλου και ο τελεστής της αντιστροφής), του οποίου η αρχή είναι αντικείμενο της έννοιας A. Συνθέτοντας τα παραπάνω μπορούμε να καταλήξουμε στην έκφραση:

$$DateOfCreation \doteq \exists (happenedIn)^{-1}.(\exists ofKind.\{"Creation"\})$$

5.2.8 Συλλογισμοί της Περιγραφικής Λογικής

Η γνώση η οποία περιέχεται σε ένα TBox και ABox είναι περισσότερη από αυτή που έχει ρητώς εισαχθεί, και μπορεί να εξαχθεί με χρήση της σημασιολογίας των εννοιών και των τελεστών, που ορίστηκε προηγουμένως. Για την εξόρυξη της γνώσης αυτής, έχουν αναπτυχθεί διάφοροι μηχανισμοί (reasoning services). Οι πιο βασικοί από αυτούς, όπως ορίζονται στα [DLNS96] και [DLNN95], είναι οι παρακάτω:

Δοθείσας μιας βάσης γνώσης $\Sigma = <\mathcal{T}, \mathcal{A}, \mathcal{L}>$, δύο εννοιών C και A, και ενός αντικειμένου α, ονομάζουμε:

- **Ικανοποίηση Εννοιών (Concept Satisfiability)**, που συμβολίζεται $\Sigma \not\models C \equiv \bot$, το πρόβλημα του ελέγχου της ικανοποίησης ή μη της έννοιας C στο Σ. Με άλλα λόγια αν η $\mathcal{I}(\mathcal{C}) \neq \emptyset$.

- **Έλεγχος συνέπειας (Consistency)**, που γράφεται ως $\Sigma \not\models$, τον έλεγχο αν υπάρχει έστω και ένα μοντέλο στο Σ ή όχι, αν δηλαδή όλες οι έννοιες καταλήγουν τελικά στην $\bot$.

- **Έλεγχος υπαγωγής (Subsumption)**, το οποίο γράφεται ως $\Sigma \models C \sqsubseteq D$, το πρόβλημα του ελέγχου αν η έννοια C εμπεριέχεται στην D, δηλαδή, αν $\mathcal{I}(C) \dot{\leq} \mathcal{I}(D)$ σε κάθε μοντέλο $\mathcal{I}$ του Σ.

- **Έλεγχος περιπτώσεως (Instance Checking)**, με συμβολισμό $\Sigma \models C(a)$, τον έλεγχο αν το $C(a)$ ικανοποιείται από κάθε μοντέλο του Σ.

- **Μη συσχέτιση και ισοδυναμία (Disjontness and Equivalence)**, η οποία συμβολίζεται $\Sigma \models C \not\equiv D$ και $\Sigma \models C \equiv D$ αντίστοιχα, τον έλεγχο αν η έννοια C είναι εντελώς ανεξάρτητη από την έννοια D, δηλαδή αν $\mathcal{I}(C) \cup \mathcal{I}(D) = \emptyset$, ή αν η C είναι ισοδύναμη με την D.

Ο έλεγχος για την ικανοποίηση μιας έννοιας είναι βασικός μια και μπορεί να χρησιμοποιηθεί για τον έλεγχο καταλληλότητας κάποιων ΣΠ του προφίλ για μια συγκεκριμένη εφαρμογή.

Ο έλεγχος συνέπειας αποτελεί βασική διαδικασία για τους μηχανικούς γνώσης που θέλουν να ελέγχουν τη συνέπεια των μοντέλων τους [BDNS94]. Η λειτουργία αυτή παρέχει, επίσης, έναν τρόπο ελέγχου της συνέπειας των προφίλ σε μια τοπική πηγή πληροφοριών.

Ο έλεγχος υπαγωγής θεωρείται η βασικότερη λειτουργία από αυτές που μπορεί να επιτελέσει η ΠΛ. Δοσμένης μιας έννοιας C και ενός TBox $\mathcal{T}$, μπορεί για κάθε έννοια D του $\mathcal{T}$ να ελεγχθεί αν $D \sqsubseteq C$ ή $C \sqsubseteq D$ ώστε η C να μπορεί να κατηγοριοποιηθεί σωστά στην ιεραρχία των εννοιών του $\mathcal{T}$ [DLNS96].

Η υπαγωγή είναι βασικό στοιχείο για πιο περίπλοκες δυνατότητες κατηγοριοποίησης. Με δεδομένο ότι τα ΣΠ ενός προφίλ αντιστοιχούν σε έννοιες του TBox, η ιεράρχηση των εννοιών συνεπάγεται και ιεράρχηση των αντίστοιχων ΣΠ. Σαν αποτέλεσμα, έχουμε την εννοιολογική δόμηση των ΣΠ του προφίλ.

Με τον έλεγχο ισοδυναμίας μπορεί κανείς να βρει ταυτόσημα ΣΠ σε διαφορετικά ή όχι προφίλ και να διαχειρίζεται ερωτήσεις σε αυτά ισοδύναμα. Η λειτουργία αυτή μπορεί να χρησιμοποιηθεί επίσης, και για κάποια μορφή αξιολόγησης του προφίλ

Τέλος, ο έλεγχος περιπτώσεως, ελέγχει αν σε μια βάση, υπάρχει ή όχι κάποιο αντικείμενο σαν περίπτωση κάποιας συγκεκριμένης έννοιας. Αποτελεί την κατεξοχήν

διαδικασία για την ανάκτηση πληροφοριών από βάσεις γνώσεις. Η ερώτηση για παράδειγμα "Δώσε όλους τους τίτλους των βιβλίων που υπάρχουν στη βάση" μεταφράζεται σε ένα επαναληπτικό έλεγχο για το αν ισχύει ή όχι το Title(a), για όλα τα a που υπάρχουν στο Abox, και επιστροφή όλων των a για τα οποία η απάντηση είναι θετική [DLNN94]. Την περίπτωση αυτή υλοποιήσαμε προκειμένου να μπορούν να απαντηθούν από τη βάση οι Z39.50 επερωτήσεις.

5.3 Υπολογιστική πολυπλοκότητα

Για την περιγραφή της υπολογιστικής πολυπλοκότητας οι έννοιες που χρησιμοποιούνται είναι σύμφωνες με τους ορισμούς που δίνονται στο [LP92].

Η πολυπλοκότητα ενός προβλήματος γενικά, μετριέται βάσει του μεγέθους των συνολικών δεδομένων εισόδου. Το αντίστοιχο θα ισχύει και κατά τη μέτρηση της πολυπλοκότητας των συλλογισμών στις εννοιολογικές γλώσσες. Στους συλλογισμούς αυτούς ως είσοδος δίνονται συνήθως περισσότερα από ένα πράγματα. Για παράδειγμα, για τον *έλεγχο περιπτώσεως*, που μας ενδιαφέρει, δίνονται το TBox, το ABox, και η έννοια για την οποία θα γίνει ο έλεγχος.

Όπως αποδεικνύεται στο [Sch94] ο *έλεγχος περιπτώσεως* απαιτεί πολυωνυμικό χρόνο σε γλώσσες όπου οι χρησιμοποιούμενοι τελεστές είναι οι: $\top$, $\bot$, $\sqcap$, $\forall$, και $\geq$. Αν στα παραπάνω προστεθεί και ο τελεστής $\sqcup$, το πρόβλημα γίνεται NP-complete. Τέλος, αν χρησιμοποιηθεί και ο τελεστής της άρνησης, το πρόβλημα μετατρέπεται σε PSPACE-hard.

Στο [DLNS96] αποδεικνύεται ότι η απάντηση επερωτήσεων με χρήση του τελεστή **K** δεν είναι υπολογιστικά περισσότερο πολύπλοκη από ότι στις γλώσσες χωρίς τον τελεστή αυτό. Άρα ο *επιστημολογικός* τελεστής δεν αυξάνει την πολυπλοκότητα της διαδικασίας της απάντησης επερωτήσεων, το οποίο είναι εξαιρετικά ενδιαφέρον ειδικά κάτω από φως της εκφραστικότητας η οποία επιτυγχάνεται. Η χρήση του τελεστή **K**, μπροστά από όλες τις έννοιες και τους ρόλους οδηγεί σε μια σημαντική μείωση της πολυπλοκότητας του *ελέγχου περιπτώσεως*. Στο [Sch94] μάλιστα, παρουσιάζεται ένας αλγόριθμος ο οποίος εκτελείται σε πολυωνυμικό χρόνο σε σχέση τόσο με την επερώτηση, όσο και με τα περιεχόμενα του ABox. Αυτό είναι ένα πολύ σημαντικό κέρδος σε σχέση με την PSPACE-hard πολυπλοκότητα που έχει ο έλεγχος περιπτώσεως σε ορισμένες γλώσσες.

Η πολυπλοκότητα του ελέγχου υπαγωγής, σύμφωνα με το [Bor95], είναι πολυωνυμική όταν χρησιμοποιούνται μόνο οι τελεστές $\sqcap$, $\forall$, $\geq$, $\leq$, και $oneof$. Η χρήση των - και $^{-1}$ κάνει την πολυπλοκότητα να είναι εκθετική.

5.4 Περιγραφική Λογική και Z39.50

5.4.1 Επεξεργασία επερωτήσεων Z39.50 μέσω Περιγραφικής Λογικής

Σύμφωνα με το [Bre95a] η ΠΛ μπορεί να χρησιμοποιηθεί σαν μια ισχυρή γλώσσα επερωτήσεων. Λόγω της σαφούς σημασιολογίας της, η μετάφραση από τις εννοιολογικές εκφράσεις σε επερωτήσεις της γλώσσας της πηγής είναι άμεση. Στο [Bor92a] αναφέρεται ότι αφού μια επερώτηση είναι απλά ένας ορισμός των ιδιοτήτων των οποίων η ικανοποίηση απαιτείται για τα αντικείμενα που θα περιέχονται στην απάντηση, είναι λογικό να μπορεί κανείς να διαχειρίζεται τις εννοιολογικές εκφράσεις σαν επερωτήσεις και αντιστρόφως. Οι έννοιες μπορούν να χρησιμοποιηθούν για ανάκτηση δεδομένων, επιστρέφοντας σαν σύνολο απαντήσεων το σύνολο των ανεξαρτήτων αντικειμένων τα οποία ικανοποιούν την περιγραφή τους. Το υποκεφάλαιο αυτό ασχολείται με το θέμα της παράστασης μιας επερώτησης Z39.50 σε ΠΛ.

Η βασική γλώσσα επερωτήσεων του Z39.50, η **type-1**, μπορεί να αποδειχθεί ότι αποτελεί περίπτωση των εννοιολογικών γλωσσών της ΠΛ. Οι τελεστές για παράδειγμα AND, OR, και AND-NOT, μπορούν να εκφραστούν μέσω εννοιολογικών εκφράσεων, με χρήση των τελεστών $\sqcap$, $\sqcup$, και - αντίστοιχα.

Προκειμένου να εμπλουτιστεί η εννοιολογική γλώσσα με λειτουργίες τελεστών, οι οποίες δεν υποστηρίζονται από τις μέχρι τώρα δυνατότητές της, εισάγεται ένας νέος τελεστής **TEST−C** για τη χρήση των λεγομένων *συναρτήσεων ελέγχου (test functions).* Οι συναρτήσεις ελέγχου είναι συναρτήσεις γραμμένες στη γλώσσα του συστήματος. Ο τρόπος χρήσης τους είναι αυτός που έχει οριστεί στην CLASSIC [BMS+91]. Οι συναρτήσεις ελέγχου αποτελούν ένα τρόπο διαφυγής από τα όρια εκφραστικότητας της εννοιολογικής γλώσσας. Παρέχουν στη γλώσσα προγραμματισμού του συστήματος, τη δυνατότητα να βλέπει και να επεξεργάζεται τις περιπτώσεις μιας έννοιας. Χειρίζονται δε από την ΠΛ σαν μαύρο κουτί και καλούνται μέσω του τελεστή **TEST−C**. Στις συναρτήσεις δίδεται σαν όρισμα ένα ανεξάρτητο αντικείμενο και επιστρέφεται η τιμή TRUE ή FALSE, ανάλογα με το αν το αντικείμενο αυτό ικανοποιεί ή όχι κάποιες προϋποθέσεις. Η ερμηνεία της έκφρασης **TEST−C**(f) είναι τα ανεξάρτητα αντικείμενα της πηγής τα οποία αν δοθούν σαν όρισμα στη συνάρτηση f, επιστρέφεται τιμή TRUE. Οι συναρτήσεις ελέγχου θα πρέπει να είναι *μονοτονικές*, δηλαδή να επιστρέφουν την ίδια τιμή για ένα συγκεκριμένο αντικείμενο, οποιαδήποτε χρονική στιγμή.

Οι ιδιότητες της *αποκοπής* και της *διάταξης* του Z39.50, μπορούν να υλοποιηθούν στην ΠΛ μέσω *συναρτήσεων ελέγχου.* Έτσι, για παράδειγμα τα ονόματα των προσώπων

τα οποία αρχίζουν από ”Andr”, είναι:

$$\{PersonalName\text{=}"Andr*"\} \quad = \quad PersonalName \sqcap TEST - C(rtrunc_{"Andr"})$$

όπου $rtrunc_{"A"}$ είναι μια συνάρτηση του συστήματος που ελέγχει αν το στοιχείο που της δίδεται σαν όρισμα, αρχίζει ή όχι από ”Andr”. Άλλες συναρτήσεις που θα πρέπει να υπάρχουν είναι οι: gt (γνήσια μεγαλύτερο), ge (μεγαλύτερο ή ίσο), lt (γνήσια μικρότερο), le (μεγαλύτερο ή ίσο), eq (ίσο), ltrunc (αριστερή αποκοπή) και lrtrunc (αριστερή και δεξιά αποκοπή).

Συμπεραίνεται επομένως, ότι η γλώσσα επερωτήσεων του Z39.50 μπορεί να παρασταθεί μέσω των εννοιολογικών γλωσσών της ΠΛ μια και όλες οι βασικές δομές της πρώτης, όπως μόλις παρουσιάστηκε, μπορούν να μοντελοποιηθούν στη δεύτερη.

Η απάντηση επερωτήσεων στο Z39.50, μπορεί να δειχθεί ότι αποτελεί μια ιδιαίτερη περίπτωση της απάντησης επερωτήσεων στην ΠΛ. Το μόνο είδος συλλογισμού που απαιτείται, είναι ο έλεγχος περιπτώσεως. Μια επερώτηση στην ΠΛ, σύμφωνα με το [DLNN94], ορίζεται ως εξής:

Ορισμός 9 *Δοθείσας μιας βάσης γνώσης* Σ*, μιας έννοιας* C *και ενός αντικειμένου* α*, η απάντηση στην επερώτηση* $C(\alpha)$ *στο* Σ *είναι YES αν* $\Sigma| \text{=} C(\alpha)$*, NO αν* $\Sigma| \not\text{=} C(\alpha)$*, και UNKNOWN διαφορετικά. Το αποτέλεσμα μιας τέτοιας ερώτησης είναι το σύνολο* $\{\alpha \in \mathcal{O}_\Sigma \mid \Sigma| \text{=} C(\alpha)\}$*, όπου* $\mathcal{O}_\Sigma$ *το σύνολο των ανεξάρτητων αντικειμένων που εμφανίζονται στο* Σ*.*

Όπως αναφέρθηκε σε προηγούμενη ενότητα, με τη χρήση του *epistemic* τελεστή, οι απαντήσεις στις επερωτήσεις, μπορούν να περιοριστούν στη γνώση η οποία περιέχεται στη βάση τη στιγμή που γίνεται η επερώτηση. Έτσι η απάντηση UNKNOWN μπορεί να παραληφθεί, οπότε η ερώτηση PersonalName="Androutsos" έχει σαν απάντηση Yes ή No, ανάλογα με το αν ο Androutsos ανήκει στην έννοια PersonalName ή όχι, και θα επιστρέφει σαν σύνολο απαντήσεων, το μονοσύνολο με στοιχείο το ”Androutsos” στην πρώτη περίπτωση, και το κενό σύνολο στη δεύτερη.

Η ιδιαιτερότητα που έχουν οι επερωτήσεις Z39.50 σε σχέση με τις επερωτήσεις των παραδοσιακών συστημάτων βάσεων είναι η συγκεντρωτική θεώρηση για την οποία μιλήσαμε στην 4.3.2. Η θεώρηση δηλαδή του Z39.50 είναι ότι σε κάθε βάση υπάρχει μια συγκεκριμένη κλάση-έννοια αντικειμένων στην οποία αναφέρονται όλες οι επερωτήσεις πάνω στα ΣΠ. Είναι κάτι αντίστοιχο με τις ρίζες διατηρητικότητας των οντοκεντρικών βάσεων δεδομένων. Η έννοια που παριστάνει την κεντρική αυτή κλάση, ονομάζεται

χαρακτηριστική έννοια και συμβολίζεται C_R. Για παράδειγμα, σε βάση με μουσειακά αντικείμενα, η χαρακτηριστική έννοια είναι τα *μουσειακά αντικείμενα* και η επερώτηση PersonalName="Androutsos" αναφέρεται στα *μουσειακά αντικείμενα* που έχουν ιδιοκτήτη τον "Androutsos". Κάτω από μια τέτοια θεώρηση, ο ορισμός που δόθηκε παραπάνω για την απάντηση επερωτήσεων είναι ελλιπής λόγω του ότι δεν μοντελοποιεί σωστά τα ως απάντηση επιστρεφόμενα αντικείμενα.

Έστω, C_{AP} η έννοια του TBox η οποία αντιστοιχεί στο ΣΠ AP, και έστω μια επερώτηση στο AP με κάποια τιμή "Value", π.χ. AP="Value". Για την απάντηση μιας τέτοιας ερώτησης ένα πρώτο βήμα είναι η εκτέλεση της: $\Sigma| = E_Q(\text{"}Value\text{"})$, όπου η E_Q είναι μια εννοιολογική έκφραση η οποία χρησιμοποιεί την έννοια C_{AP}. Σε περιπτώσεις που η επερώτηση του Z39.50 δεν περιέχει διάταξης ή αποκοπής ιδιότητες, τότε η E_Q είναι ίση με C_{AP}. Σε αντίθετη περίπτωση, η E_Q ισούται με $C_{AP} \sqcap TEST - C(f_{\text{"}Value\text{"}})$ όπου f συνάρτηση που εξαρτάται από τις πιθανές ιδιότητες διάταξης ή αποκοπής που έχουν ζητηθεί. Έστω C_{Res} το σύνολο απαντήσεως $\Sigma| = E_Q(Value)$. Για παράδειγμα, στην επερώτηση:

PersonalName="Andr" ***AND*** **Truncation**="Right"

η έννοια C_{AP} είναι η PersonalName, η έκφραση "Value" είναι η "Andr" και συνάρτηση f η rtrunc. Με δεδομένου του ότι η εννοιολογική έκφραση που ορίζει την PersonalName είναι η:

$$PersonalName \doteq Owner \sqcup Actor$$

η $E_Q(\text{"}Value\text{"})$ θα είναι: $(Owner \sqcup Actor) \sqcap TEST - C(rtrunc_{\text{"}Andr\text{"}})$. Η C_{Res} θα παριστάνει είναι όλα εκείνα τα πρόσωπα των οποίων το όνομα αρχίζει από "Andr".

Επόμενο βήμα για την απάντηση της επερώτησης Z39.50 είναι η εξαγωγή των ανεξάρτητων αντικειμένων που ανήκουν στην κεντρική έννοια C_R της πηγής, και τα οποία συσχετίζονται με τα αντικείμενα της έννοιας C_{Res}. Τα αντικείμενα αυτά μπορούν να παρασταθούν μέσω μιας εννοιολογικής έκφρασης, την E_P, η οποία εμπεριέχει την C_{Res}. Για το παράδειγμα της προηγούμενης παραγράφου και τη βάση του σχήματος 5.2, θα είναι:

$$E_P = (\exists ownedBy.C_{Res}) \sqcup (\exists hasActor.(\exists hasEvent.C_{Res}))$$

Το τελικό σύνολο απαντήσεων της ερώτησης θα είναι η έννοια που προκύπτει από το παραπάνω βήμα, αφού τμηθεί με την C_R ώστε να είναι εγγυημένο ότι το τελικό σύνολο

απαντήσεων C_Q περιέχει μόνο αντικείμενα της C_R. Ωστόσο, σύμφωνα με τον τρόπο με τον οποίο έχουμε θεωρήσει ότι ορίζονται οι ρόλοι, κάτι τέτοιο είναι εγγυημένο, οπότε:

$$C_Q = C_R \sqcap (E_P \circ C_{Res}) = E_P \circ C_{Res} = (\exists ownedBy.C_{Res}) \sqcup (\exists hasActor.(\exists hasEvent.C_{Res}))$$

Η διαδικασία που μόλις περιγράφηκε αναφέρεται στην απάντηση μιας επερώτησης η οποία περιέχει ένα και μόνο ΣΠ. Για την απάντηση επερώτησης που περιέχει πάνω από ένα ΣΠ, θα πρέπει σε πρώτο βήμα να χωριστεί σε τμήματα, ένα για κάθε ΣΠ, και τα οποία θα συνδέονται μεταξύ τους με τους τελεστές *AND*, *OR* και *AND_NOT* του Z39.50. Για κάθε τμήμα ακολουθείται η διαδικασία που περιγράφηκε παραπάνω η οποία θα επιστρέψει ένα σύνολο από αντικείμενα της C_R. Όπως θα δείξουμε παρακάτω, το σύνολο απαντήσεων μπορεί κι αυτό να παρασταθεί σαν μια έννοια (βλ. 5.4.2.4). Έτσι το τελικό σύνολο απαντήσεων θα προέρχεται από μια εννοιολογική έκφραση η οποία θα περιέχει τα ενδιάμεσα αποτελέσματα. Για παράδειγμα έστω η επερώτηση που αναζητεί τα αντικείμενα εκείνα που συσχετίζονται με πρόσωπα των οποίων το όνομα αρχίζει από "Andr" και χρονολογούνται σε έτη μεταγενέστερα του 1821. Η επερώτηση αυτή φαίνεται παρακάτω. Τα ΣΠ που περιέχει είναι δύο (Date και PeronalName) οπότε χωρίζεται σε δύο τμήματα, όπως φαίνεται από τις παρενθέσεις.

(**Date**=1821 ***AND*** **Relation**="GreaterThan" ***)*** ***AND*** ***(*** **PersonalName**="Andr" ***AND*** **Truncation**="Right" ***)***

Κάθε τμήμα που περικλείεται μέσα στην παρένθεση επεξεργάζεται σύμφωνα με τη διαδικασία που αναλύσαμε προηγουμένως. Έστω C_{Q1} η απάντηση της πρώτης παρένθεσης (τα αντικείμενα που χρονολογούνται μετά το 1821), και C_{Q2} η απάντηση της δεύτερης (τα αντικείμενα που σχετίζονται με πρόσωπα των οποίων το όνομα αρχίζει από "Andr"). Η τελική απάντηση στην επερώτηση θα είναι η έννοια:

$$C_R = C_{Q1} \sqcap C_{Q2}$$

Από όλα τα παραπάνω, γίνεται φανερό ότι για να είναι δυνατή η απάντηση σε μια επερώτηση θα πρέπει για κάθε ΣΠ που αναφέρει, να ισχύουν υποχρεωτικά τα παρακάτω:

1. Να υπάρχει μια (και μόνο) χαρακτηριστική κεντρική έννοια C_R της βάσης γνώσης.
2. Για κάθε σημείο πρόσβασης να υπάρχει μια αντίστοιχη παράγωγη έννοια C_{AP} η οποία να έχει όσο το δυνατό ίδια με αυτό σημασία, και
3. Για κάθε C_{AP}, να υπάρχει μια αντίστοιχη εννοιολογική έκφραση E_P η οποία να χρησιμοποιεί την C_{Res}.

Αν κάποια από τα παραπάνω δεν ικανοποιείται για κάποιο σημείο πρόσβασης, τότε επερωτήσεις πάνω σε αυτό δεν μπορούν να εκτελεστούν, και έτσι το ΣΠ αυτό, αναγκαστικά, δεν υποστηρίζεται.

5.4.2 Πλεονεκτήματα χρήσης της ΠΛ σε Z39.50 servers

5.4.2.1 Δόμηση του επιπέδου λεξιλογίου επερωτήσεων

Οι επερωτήσεις στην επίπεδη λίστα των ΣΠ του Z39.50 στηρίζονται κυρίως σε αναζητήσεις με βάση λέξεις κλειδιά και συνδυασμούς τους. Αυτού του είδους οι επερωτήσεις μπορούν να θεωρηθούν σαν επερωτήσεις πάνω σε μια καθολική σχέση [Ull82]. Ωστόσο, μια τέτοια προσέγγιση είναι κατάλληλη περισσότερο για αδόμητα ή ημιδομημένα δεδομένα, παρά για τις βάσεις δομημένων πληροφοριών όπως είναι για παράδειγμα οι πολιτισμικές. Το μεγαλύτερο μέρος της συντακτικής και εννοιολογικής δομής της πηγής, χάνεται κατά τη μετάβαση από το δομημένο σχήμα στο επίπεδο. Η εννοιολογική δόμηση των ΣΠ θα ήταν χρήσιμη γιατί θα συντελούσε στην:

- ευκολότερη αναζήτηση της πληροφορίας,
- έλεγχο καταλληλότητας ΣΠ ενός προφίλ,
- ευκολότερη διαδικασία απάντησης επερωτήσεων,
- βελτιστοποίηση επερωτήσεων
- εισαγωγή επιπλέον σημασιολογικής πληροφορίας στα ανεξάρτητα ΣΠ.

Η εννοιολογική δόμηση αυτή δόμηση μπορεί να γίνει μέσω της σχέσης *υπάγεται* της ΠΛ, η οποία οργανώνει τις έννοιες του TBox σε μια ιεραρχία. Η ιεραρχία αυτή προσδιορίζει αν μια έννοια είναι ευρύτερη ή στενότερη από κάποια άλλη. Στην ιεραρχία που δημιουργείται, περιέχονται και οι έννοιες οι οποίες αντιστοιχούν στα ΣΠ του προφίλ. Έτσι, η επίπεδη λίστα των ΣΠ αποκτά κάποια εννοιολογική δομή.

Η ταξινόμηση των εννοιών (και κατά συνέπεια και των ΣΠ) στην ιεραρχία αυτή, μπορεί να γίνει με δύο τρόπους. Ο πρώτος τρόπος είναι η ρητή δήλωση της σχέσης *υπάγεται* μεταξύ δύο εννοιών. Ο δεύτερος τρόπος είναι αυτόματος και γίνεται με χρήση των συλλογιστικών μηχανισμών της ΠΛ.

Έστω ένα παράδειγμα από το AQUARELLE profile [Ltd97]. Το ΣΠ Date παριστάνει ημερομηνίες, ενώ το DateOfCreation παριστάνει ημερομηνίες δημιουργίας αντικειμένων. Επίσης, το ΣΠ WHEN παριστάνει, γενικά, χρονικές στιγμές. Είναι προφανές ότι η WHEN

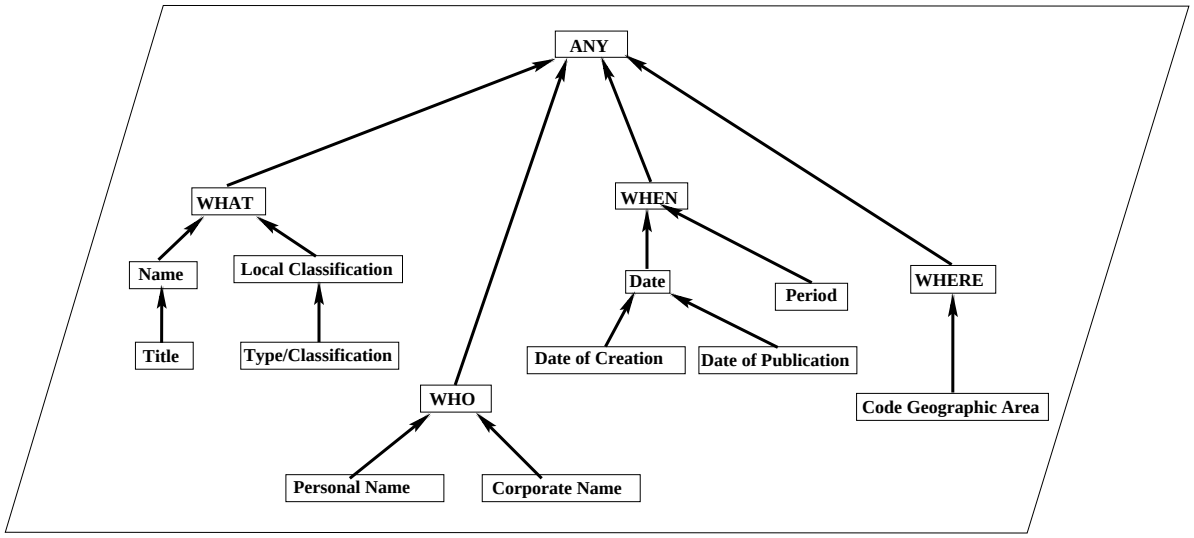
είναι μια ευρύτερη έννοια από την Date, η οποία με τη σειρά της είναι ευρύτερη της DateOfCreation. Υπενθυμίζουμε ότι οι ορισμοί των εννοιών αυτών είναι:

$$\begin{aligned} Date &\doteq EventDate \\ DateOfCreation &\doteq \exists(happenedIn)^{-1}.(\exists ofKind.\{\text{"}Creation\text{"}\}) \\ WHEN &\doteq Date \sqcup Period \end{aligned}$$

όπου το "$Creation$" αποτελεί περίπτωση της έννοιας $Kind$ του τμήματος σχήματος του TBox. Από τους ορισμούς αυτούς το σύστημα είναι σε θέση, χωρίς να έχει δηλωθεί ρητά, να συμπεράνει ότι:

$$\begin{aligned} DateOfCreation &\dot{\leq} Date \\ Date &\dot{\leq} WHEN \end{aligned} \tag{5.1}$$

Η ιεράρχηση που μόλις περιγράφηκε φαίνεται στο σχήμα 5.3 μαζί με κάποιες άλλες ιεραρχήσεις οι οποίες και αυτές παράγονται με όμοιο τρόπο. Η σημασιολογία των εννοιών που αναφέρονται, αν και προφανής, περιγράφεται στο [Ltd97].



Σχήμα 5.3: Δόμηση ενός συνόλου από ΣΠ, βάσει της σχέσης υπάγεται

5.4.2.2 Έλεγχος καταλληλότητας προφίλ

Ο έλεγχος του κατά πόσο ένα συγκεκριμένο προφίλ είναι κατάλληλο για να παραστήσει την πληροφορία κάποιας πηγής, είναι γενικά έξω από το σκοπό του Z39.50.

Μέχρι σήμερα, ένας τέτοιος έλεγχος γίνεται με χρήση άλλων εργαλείων η από το διαχειριστή. Με τη χρήση της ΠΛ, ένας τέτοιος έλεγχος γίνεται εφικτός μέσα στο πλαίσιο της κατασκευής ενός μεταφραστή Z39.50.

Η αντιστοίχιση ενός ΣΠ με την έννοια $\bot$ σημαίνει ότι η πηγή δεν έχει την απαραίτητη πληροφορία-γνώση για την έννοια, την οποία περιγράφει το ΣΠ. Η ύπαρξη πολλών ΣΠ αντιστοιχισμένων στην $\bot$ σημαίνει ότι πολλά από τα ΣΠ του προφίλ δεν μπορούν να παρασταθούν από δομές της συγκεκριμένης πηγής.

Με χρήση της συλλογιστικής δυνατότητας του *ελέγχου συνέπειας* και της *ικανοποίησης εννοιών* ο παραπάνω έλεγχος μπορεί να γίνει. Αν ο έλεγχος δείξει ότι όλες οι έννοιες που παριστάνουν τα ΣΠ καταλήγουν να είναι ισοδύναμες με την έννοια $\bot$, τότε το προφίλ αποδεικνύεται ακατάλληλο για να περιγράψει το είδος της πληροφορίας που βρίσκεται αποθηκευμένο στην πηγή.

5.4.2.3 Βελτίωση του τρόπου απάντησης επερωτήσεων

Η αντιστοίχιση των ΣΠ του Z39.50 στις σωστές δομές της πηγής είναι δύσκολη και πολύπλοκη διαδικασία, γιατί οι παράμετροι που θα πρέπει να ληφθούν υπόψη, είναι αρκετές. Η αντιστοίχιση μπορεί να είναι ένα προς ένα, ένα προς πολλά, πολλά προς ένα, και πολλά προς πολλά. Αν για κάποιο ΣΠ δεν μπορεί να υπάρξει αντιστοίχηση, τότε αυτό δεν υποστηρίζεται.

Σύμφωνα με τα όσα ορίζει το πρωτόκολλο, όταν στον server φτάσει μια επερώτηση στην οποία υπάρχει κάποιο μη υποστηριζόμενο ΣΠ, η επερώτηση αποτυγχάνει, και ένα διαγνωστικό μήνυμα επιστρέφεται στον client, προσδιορίζοντας την αιτία της αποτυχίας. Αυτό όπως έχει ήδη εξηγηθεί στην ενότητα 4.3.5 δεν είναι πολύ καλό καθώς η πιθανότητα αποτυχίας σε κατανεμημένο και ετερογενές περιβάλλον είναι μεγάλη. Όμως, ούτε η λύση που συχνά υιοθετείται, της παράλειψης των μη υποστηριζόμενων ΣΠ, είναι ικανοποιητική, μολονότι αποτρέπει τον κινδυνο αποτυχίας της επερώτησης.

Η ΠΛ παρέχει μια αρκετά ευέλικτη μέθοδο για την αντιμετώπιση του θέματος αυτού. Η βασική αρχή είναι ότι όλα τα ΣΠ θα πρέπει να υποστηρίζονται, οπότε εξαλείφεται η περίπτωση αποτυχίας μιας επερώτησης λόγω ανυποστήρικτων ΣΠ. Τα ΣΠ για τα οποία δεν υπάρχει κάποια εννοιολογικά αντίστοιχη δομή στην πηγή, ο διαχειριστής του μεταφραστή μπορεί να αποφασίσει να τα αντιστοιχίσει είτε

- στην έννοια $\top$, κάτι που θα έχει σαν αποτέλεσμα την αύξηση της ανάκλησης (recall) στις απαντήσεις των επερωτήσεων, είτε

- στην έννοια $\perp$, κάτι που θα έχει σαν αποτέλεσμα την αύξηση της ακρίβειας (precision).

Εξαίρεση στα παραπάνω αποτελεί η περίπτωση όπου ένα ΣΠ δεν έχει αντίστοιχη δομή στην πηγή, παριστάνει όμως πληροφορία η οποία είναι χαρακτηριστική στη βάση, περίπτωση όπως αυτή του παραδείγματος 2, όπου μπορεί να αντιστοιχιθεί σε μια έννοια η οποία παράγεται από τον τελεστή OneOf.

Μια άλλη προβληματική περίπτωση είναι αυτή κατά την οποία για κάποιο ΣΠ υπάρχει αντίστοιχη δομή στη πηγή, δεν έχει όμως κάποιο τρόπο συσχετισμού με την κεντρική έννοια C_R. Είναι η περίπτωση όπου το εννοιολογικό σχήμα της πηγής δεν ακολουθεί το σχήμα μιας καθολικής σχέσης. Για την αντιμετώπισή του, μέσω ΠΛ, προτείνεται η αντιστοίχηση του με την έννοια C_R ή $\perp$, ανάλογα αν απαιτείται μεγαλύτερη ανάκληση, ή ακρίβεια.

5.4.2.4 Ιεράρχηση επερωτήσεων

Το Z39.50 ως πρωτόκολλο διατηρητέων καταστάσεων υποστηρίζει την διατήρηση προηγουμένων αποτελεσμάτων επερωτήσεων και τη χρησιμοποίησή τους στην κατασκευή νέων. Από την πλευρά της η ΠΛ παρέχει ένα τρόπο με τον οποίο τα σύνολα απαντήσεων μπορούν και παριστάνονται και χειρίζονται ως έννοιες. Μια επερώτηση δεν είναι παρά μία περιγραφή ενός συνόλου αντικειμένων (αυτών της απάντησης) και επομένως σύμφωνα με το [BSV95], μπορεί να παρασταθεί σαν μια έννοια στο TBox. Για παράδειγμα, η επερώτηση Date=1821 σε μια μουσειακή βάση πληροφοριών είναι μια έννοια της οποίας η ερμηνεία είναι το σύνολο των μουσειακών αντικειμένων που χρονολογούνται στο έτος 1821. Ο server επομένως ο οποίος στηρίζει την λειτουργία του στην ΠΛ έχει ένα τρόπο για την παράσταση, αποθήκευση και διαχείριση των προηγουμένων επερωτήσεων και των αποτελεσμάτων τους, για την κατασκευή νέων επερωτήσεων. Οι επερωτήσεις, όπως αντίστοιχα και οι έννοιες, μπορούν να παρασταθούν και να ταξινομηθούν σε μια ιεραρχία βάσει της σχέσης *υπάγεται*.

Η ιεραρχία αυτή μπορεί να χρησιμοποιηθεί για πιο αποδοτική αποθήκευση των επερωτήσεων, ακόμα και όταν δεν χρησιμοποιούνται ρητώς προηγούμενα αποτελέσματα. Έστω για παράδειγμα οι παρακάτω **type-1** επερωτήσεις:

- Q1: **PersonalName**="Androutsos"
- Q2: **PersonalName**="Androutsos" ***AND*** **Date**=1821
- Q3: **Date**=1821

- Q4: **Person**="Androutsos" ***AND*** **Event**="Creation"

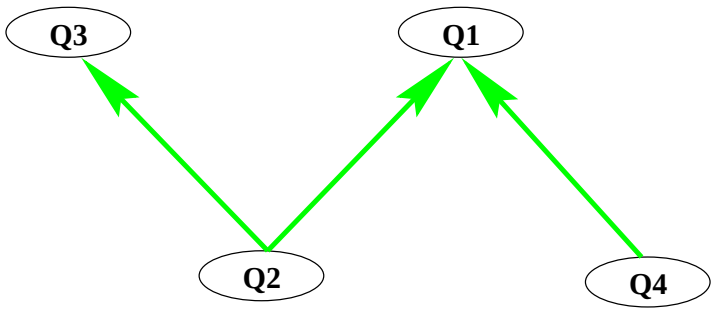
Παρατηρεί κανείς ότι το σύνολο αποτελεσμάτων της Q2 είναι αυτό της Q1 περιορισμένο στην ημερομηνία 1821. Άρα η έννοια που παριστάνει η Q1 είναι γενικότερη από αυτή της Q2. Η Q2 δεν χρησιμοποιεί ρητά την Q1. Σύμφωνα με το φυσιολογικό τρόπο εκτέλεσης, το σύστημα θα επέλεγε από όλα τα αντικείμενα της βασικής έννοιας εκείνα που σχετίζονται με το όνομα ''Androutsos''. Έπειτα θα ανακτούσε όλα όσα σχετίζονται με την ημερομηνία 1821. Τέλος, θα επέλεγε τα κοινά στοιχεία των δύο συνόλων. Κάτι τέτοιο μπορεί να προξενεί μεγάλη κατανάλωση σε πόρους του συστήματος, ιδιαίτερα σε περιπτώσεις όπου τα ενδιάμεσα αποτελέσματα-σύνολα είναι πολύ μεγάλα. Η κατηγοριοποίηση των επερωτήσεων και των περιγραφών τους στην ΠΛ μπορεί να συντελέσει σημαντικά στην άμβλυνση του παραπάνω προβλήματος. Όταν φτάσει στο σύστημα η επερώτηση Q2, το σύστημα θα μπορεί μέσω των συλλογιστικών μηχανισμών του να συμπεράνει ότι η Q2 είναι η Q1 με έναν επιπλέον περιορισμό. Άρα το σύνολο απαντήσεών της εμπεριέχεται σε αυτό της Q1. Κατά συνέπεια, δεν είναι απαραίτητο να αναζητηθούν τα αντικείμενα που ικανοποιούν τις συνθήκες της επερώτησης ανάμεσα σε όλα τα αντικείμενα της βάσης. Αρκεί να ελεγχθούν μόνο τα αντικείμενα της Q1 και να επιλεγούν όσα από αυτά σχετίζονται με την ημερομηνία 1821. Ο τρόπος αυτός βελτιστοποίησης της διαδικασίας απάντησης αποτελεί μέρος της λεγόμενης *εντασιακής βελτιστοποίησης επερωτήσεων (intensional query refinement)* [BSV95]. Οι τέσσερις παραπάνω επερωτήσεις ταξινομούνται σε μια ιεραρχία όπως αυτή του σχήματος 5.4.

Η ταξινόμηση αυτή είναι χρήσιμη όχι μόνο κατά τη διαδικασία απάντησης επερωτήσεων, αλλά και κατά τη διαδικασία αποθήκευσης αυτών. Έστω για παράδειγμα ότι έχουν γίνει οι επερωτήσεις Q2 και Q4, και το σύστημα έχει κρατήσει τα σύνολα των αντικειμένων που αποτελούν τα σύνολα απαντήσεών τους. Κάποια στιγμή γίνεται η επερώτηση Q1. Αν το σύστημα έχει ένα τρόπο να καταλάβει ότι η Q1 περιέχει την Q2 και Q4, τότε αντί να κρατήσει όλα τα αντικείμενα του συνόλου απαντήσεων της, θα μπορούσε να κρατήσει μόνο εκείνα τα οποία δεν περιέχονται στα ήδη κρατημένα σύνολα απαντήσεων των Q2 και Q4. Έτσι επιτυγχάνεται μια μεγάλη οικονομία στο χώρο αποθήκευσης.

Η ταξινόμηση των επερωτήσεων στην ιεραρχία μπορεί να χρησιμοποιηθεί προκειμένου να γίνει κάποια μορφή *εντασιακής απάντησης επερωτήσεων (intensional query answering)*. Με τον όρο αυτό δηλώνεται η απάντηση επερωτήσεων η οποία είναι γίνεται χωρίς πρόσβαση στα δεδομένα, αλλά μονάχα με χρήση του εννοιολογικού σχήματος[3].

[3]Ο τρόπος απάντησης επερωτήσεων με χρήση και των εκτάσεων (extensions) των εννοιών αποκαλείται

Όταν μια επερώτηση φτάσει στον server αυτός θα προσπαθήσει να την τοποθετήσει στην κατάλληλη θέση στην ιεραρχία. Αν η θέση αυτή είναι η θέση της έννοιας $\perp$, τότε χωρίς να την εκτελέσει μπορεί να επιστρέψει στον server το κενό σύνολο.



Σχήμα 5.4: Ταξινόμηση των επερωτήσεων στην ΠΛ

Θα ήταν πολύ χρήσιμο αν ο client θα μπορούσε να εκμεταλλευτεί την ιεραρχία αυτή των επερωτήσεων. Από την πλευρά του, στο βαθμό που έχει πρόσβαση στην ιεραρχία θα μπορούσε να έχει στη διάθεσή του ένα ιστορικό προηγουμένων επερωτήσεων και των αποτελεσμάτων τους, και να επιλέγει από αυτές προκειμένου να συνθέσει νέες. Επίσης, θα μπορούσαν οι επερωτήσεις να ομαδοποιούνται στην ιεραρχία κάτω από ομάδες εννοιών και έτσι ο χρήστης να μπορεί να βρίσκει προγενέστερες επερωτήσεις σχετικές με αυτό που θέλει να ανακτήσει [Bor94]. Επίσης, σε περιπτώσεις που το σύνολο απαντήσεων μιας επερώτησής είναι πολύ μεγάλο ή πολύ μικρό θα μπορούσε ο χρήστης να εξειδικεύσει (query refinement) ή να γενικεύσει (query generalization) κάποιους όρους βάσει της ιεραρχίας αυτής, ώστε να πετύχει ένα ικανοποιητικού μεγέθους σύνολο απαντήσεων [BDNS94]. Στο [BSV95] γίνεται μια πρόταση αυτή η γενίκευση/εξειδίκευση να γίνεται από τον server χωρίς να ειδοποιείται για κάτι τέτοιο ο χρήστης, επιλέγοντας αντίστοιχα από την ιεραρχία αυτή που είναι περισσότερο κοντά της *(Least Upper Bound ή LUB)* σε περίπτωση που η αρχική επερώτηση επιστρέψει κενό σύνολο απαντήσεων και αυτή που είναι περισσότερο κοντά σε αυτή που τέθηκε, και δεν είναι γενικότερη από αυτή *(Greatest Lower Bound ή GLB)* σε περίπτωση πολύ μεγάλου συνόλου απαντήσεων. Μια τέτοια προσέγγιση δεν είναι και η καλύτερη γιατί ο χρήστης δεν λαμβάνει γνώση του γεγονότος αυτού. Επίσης, με χρήση της ιεραρχίας ο χρήστης θα μπορούσε να κάνει πλοήγηση πάνω στις έννοιες, διεργασία ανάλογη με αυτή της πλοήγησης πάνω στα υπερκείμενα, ερευνώντας και ανακαλύπτοντας κρυμμένα γεγονότα της βάσης γνώσης. Δυστυχώς όμως οι διαδικασίες που περιγράφηκαν στην παράγραφο αυτή δεν μπορούν

εκτασιακός (extensional).

να υλοποιηθούν με την υπάρχουσα έκδοση του πρωτοκόλλου.

Θα πρέπει να σημειωθεί ότι η εννοιολογική βελτιστοποίηση των επερωτήσεων καθώς και η ταξινόμησή τους δεν μπορεί να γίνει σε περίπτωση που περιέχουν συναρτήσεις ελέγχου εφ'όσον ο τρόπος λειτουργίας τους και τα αποτελέσματα που δίνουν είναι έξω από τον μηχανισμό λειτουργίας ΠΛ.

5.4.2.5 Ενημέρωση για την ποιότητα των αποτελεσμάτων

Πολλές φορές, θα ήταν βοηθητικό για το χρήστη που κάνει μια επερώτηση και λαμβάνει κάποια αποτελέσματα, να γνωρίζει το αν και κατά πόσο τα αποτελέσματα αυτά είναι αυτό που πραγματικά ήθελε να ρωτήσει. Μια ιδέα για κάτι τέτοιο, η οποία βρίσκει εφαρμογή στο Z39.50, είναι η γνώση του τρόπου με τον οποίο έχουν γίνει η αντιστοιχίσεις των ΣΠ στις δομές της πηγής, καθώς και ο τρόπος με τον οποίο η επερώτηση εκτελέστηκε. Κάτι τέτοιο, όμως, δεν προβλέπεται από το πρωτόκολλο, και έτσι τέτοιου είδους γνώση δεν μπορεί να δοθεί στον client.

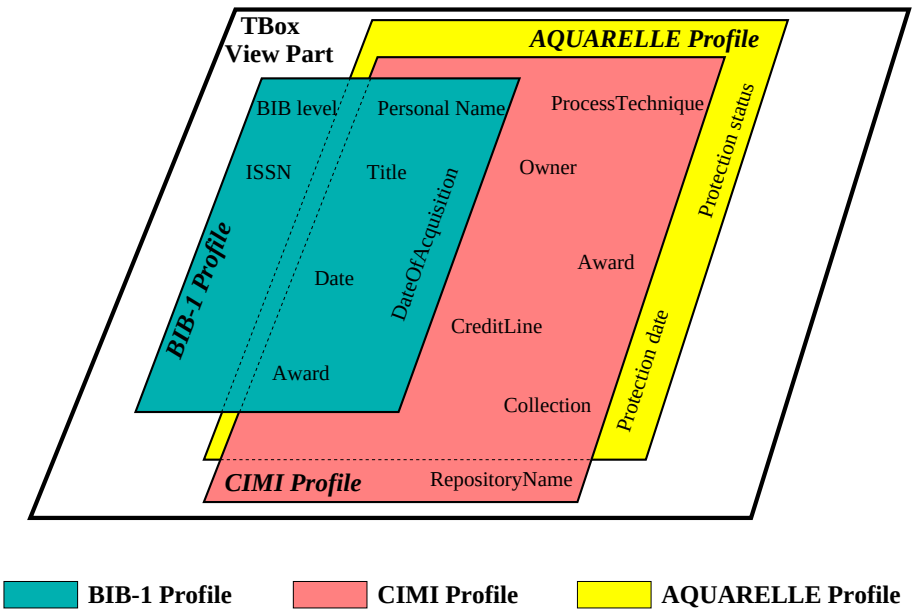
Αν χρησιμοποιούνταν η ΠΛ, θα μπορούσαν να ανακτηθούν οι εννοιολογικές εκφράσεις οι οποίες καθορίζουν τις αντιστοιχίσεις των ΣΠ στην πηγή. Από αυτή την αντιστοίχιση μπορεί να εξαχθούν συμπεράσματα για τη δομή και λειτουργία της βάσης, τη σημασιολογία που έχει αποδοθεί στα ΣΠ και ίσως τον τρόπο εκτέλεσης της επερώτησης. Η παροχή μιας τέτοιας δυνατότητας στο Z39.50 μπορεί να υλοποιηθεί με χρήση του *Explain Facility*, μια υπηρεσία η οποία μπορεί να χρησιμοποιηθεί από των client προκειμένου να ανακτήσει μετα-πληροφορίες από τη βάση. Επίσης, η γνώση των εννοιολογικών εκφράσεων θα μπορεί να βοηθήσει το χρήστη στην ευκολότερη και σωστότερη σύνθεση επερωτήσεων, δίνοντάς του μια ιδέα για το εννοιολογικό περιεχόμενο των στοιχείων που περιέχονται στην πηγή.

Η υπηρεσία *Explain* θα μπορούσε να χρησιμοποιηθεί για ανάκτηση των αντιστοιχίσεων, ακόμα κι αν οι αντιστοιχίσεις είχαν γίνει με χρήση της γλώσσας επερωτήσεων της πηγής, όπως για παράδειγμα έγινε στην υλοποίηση που περιγράφηκε στο κεφάλαιο 4, και όχι με εννοιολογικές εκφράσεις. Όμως, αυτό θα ανάγκαζε το χρήστη που συνθέτει την επερώτηση να έχει γνώση της γλώσσας αυτής. Αντίθετα, η εννοιολογική γλώσσα της ΠΛ, όντας περισσότερο κοντά στη φυσική, μπορεί πολύ εύκολα να αποτελέσει ένα κοινά κατανοητό και αποδεκτό μέσο επικοινωνίας για όλους τους clients και τους servers.

5.4.2.6 Υποστήριξη πολλαπλών προφίλ

Μέχρι σήμερα, συνήθως, κάθε Z39.50 server ακολουθεί ένα συγκεκριμένο προφίλ. Έτσι μια συγκεκριμένη πηγή πληροφοριών δεν θα μπορούσε να λάβει επερωτήσεις ελεύθερα από client οποιουδήποτε προφίλ, μια και η υποστήριξη κάτι τέτοιου ήταν μια χρονοβόρα και πολύπλοκη διεργασία.

Με τη χρήση της ΠΛ, κάτι τέτοιο απλοποιείται αρκετά. Οι έννοιες του TBox δεν εξαρτώνται από το είδος του προφίλ που υποστηρίζεται από τον server. Έτσι, ο διαχειριστής του μεταφραστή θα μπορεί να ορίσει στο TBox όλα τα ΣΠ που υπάρχουν ανεξαρτήτως προφίλ και να τα αντιστοιχίσει στις κατάλληλες δομές. Έπειτα από αυτό ο server, θα είναι υπεύθυνος για το ποια ΣΠ υποστηρίζονται και τη σωστή απάντηση σε επερωτήσεις που χρησιμοποιούν κάποιο προφίλ. Το χρησιμοποιούμενο προφίλ, επομένως, παύει πλέον να είναι χαρακτηριστικό του server, και γίνεται χαρακτηριστικό μόνο της επερώτησης. Στο σχήμα 5.5 φαίνεται ένα TBox στο οποίο έχουν οριστεί έννοιες από διάφορα προφίλ. Από αυτά, είναι ευθύνη του server να αποφασίσει ποια θα υποστηρίξει και ποια όχι ανάλογα με τις πληροφορίες της υποκείμενης πηγής.



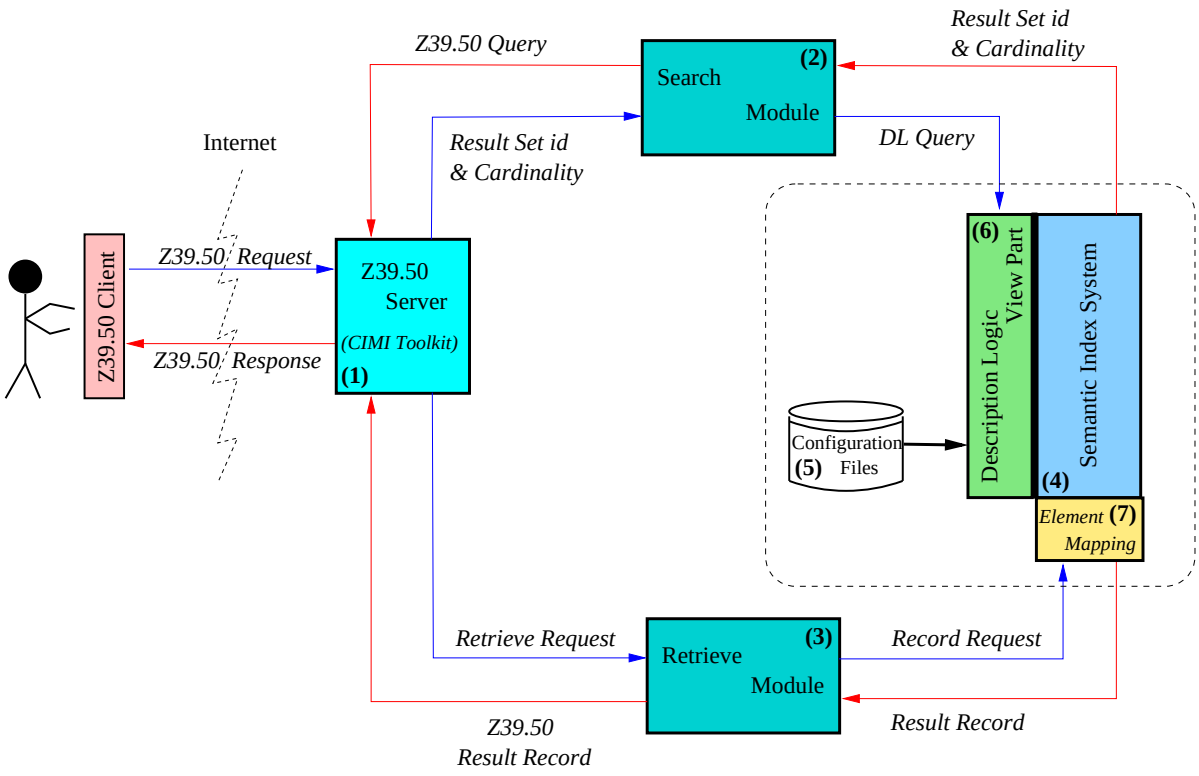
Σχήμα 5.5: TBox με υποστήριξη πολλαπλών προφίλ

5.5 Περιγραφή υλοποίησης Z39.50 server με χρήση ΠΛ

Στο κεφάλαιο αυτό παρουσιάζεται ο τρόπος με τον οποίο ο Z39.50 server ο οποίος παρουσιάστηκε στο κεφάλαιο 4 μπορεί να διαμορφωθεί και να λειτουργεί με χρήση της ΠΛ. Ο τρόπος μετάφρασης και γενικότερα της λειτουργίας του μεταφραστή θα καθορίζεται δυναμικά από το διαχειριστή ο οποίος τον κατασκευάζει. Οι αντιστοιχίσεις που γίνονται στα πεδία της βάσης, περιγράφονται σε ένα αρχείο με χρήση της εννοιολογικής γλώσσας. Ο μεταφραστής διαβάζοντας το αρχείο αυτό μαθαίνει τις αντιστοιχίσεις που πρέπει να γίνουν, και βάσει αυτών κάνει τις μεταφράσεις των επερωτήσεων.

5.5.1 Αρχιτεκτονική

Η αρχιτεκτονική που είχε παρουσιαστεί στο σχήμα 4.1 μεταβάλλεται ελαφρά και καταλήγει σε αυτή του σχήματος 5.6.



Σχήμα 5.6: Αρχιτεκτονική του Z39.50 server με χρήση ΠΛ

Όπως φαίνεται από το σχήμα έχουν προστεθεί δύο νέα τμήματα ((6), και (7)) και από το (2) και (3) έχουν αφαιρεθεί κάποιες αρμοδιότητες. Αναλυτικά:

Το τμήμα (2) πλέον δεν είναι υπεύθυνο για να κάνει την αντιστοίχιση των ΣΠ στα

πεδία της πηγής. Σκοπός του είναι να κάνει μια ανάλυση του συντακτικού δένδρου της επερώτησης και να τη χωρίσει σε τμήματα, ώστε κάθε ένα από αυτά να περιέχει ένα και μόνο ΣΠ. Έπειτα μεταφράζει την επερώτηση αυτή σε μια εννοιολογική έκφραση της ΠΛ και τη δίνει στο τμήμα (6). Έπειτα λαμβάνει από το ΣΣΕ (τμήμα (4)) τον μοναδικό κωδικό αναγνώρισης του συνόλου απαντήσεων της επερώτησης που έστειλε στο (6) καθώς και των πληθάριθμο του συνόλου αυτού. Τα στοιχεία αυτά τα στέλνει στο τμήμα (1). Είναι χαρακτηριστικό ότι το τμήμα αυτό δεν επικοινωνεί με το ΣΣΕ μια και δεν είναι πλέον αυτό υπεύθυνο για την αντιστοίχιση των ΣΠ στις δομές της πηγής.

Το τμήμα (4) με τη σειρά του όταν θα πάρει την εννοιολογική έκφραση από το (2) θα προσπαθήσει να την εκτελέσει. Το τμήμα αυτό είναι το σύστημα της ΠΛ και είναι αυτό που γνωρίζει τις αντιστοιχίσεις των ΣΠ με τις δομές της πηγής. Τις αντιστοιχίσεις αυτές τι μαθαίνει διαβάζοντας κατά την εκκίνηση της εφαρμογής ένα αρχείο στο οποίο περιγράφονται οι έννοιες αυτές (τμήμα (5)). Το τμήμα (6) είναι το *μέρος όψεως* και μόνο της ΠΛ. Σε αυτό δηλαδή ορίζονται οι παράγωγες έννοιες, κάθε μια από τις οποίες αντιστοιχεί και σε ένα ΣΠ. Οι αρχικές έννοιες και τα δεδομένα, εξακολουθούν να βρίσκονται αποθηκευμένα στο ΣΣΕ (τμήμα (4)). Το τμήμα (6) από τις εννοιολογικές εκφράσεις και σύμφωνα με όσα έχουν περιγραφεί σε προηγούμενες ενότητες θα κάνει την εκτέλεση της επερώτησης. Κάθε τελεστής μετατρέπεται σε μία ή περισσότερες επερωτήσεις PQI σύμφωνα με κάποιο πίνακα ο οποίος βρίσκεται ενσωματωμένος στο δυαδικό κώδικα. Ένα δείγμα του πίνακα αυτού που δείχνει για ορισμένους τελεστές της εννοιολογικής γλώσσας τις αντίστοιχες εκφράσεις PQI, φαίνεται στον πίνακα 5.3.

Operator	**PQI function(s)**
A	*get_all_instances(A)*
TOP	*set_curr_node(RootClass);* *get_all_instances(A);*
BOTTOM	*set_get_new();*
(OR C D)	*set_union(*C, D*);*
(AND C, D)	*set_intersect(*C, D*);*
(MINUS A)	*set_difference(*TOP, A*)*
.....	.....

Πίνακας 5.3: Τελεστές σύνθεσης εννοιών και οι αντίστοιχές συναρτήσεις του PQI

Οι εννοιολογικές εκφράσεις οι οποίες έχουν περιγραφεί στο αρχείο (τμήμα (5)) ακολουθούν την σύνταξη της δεύτερης στήλης των πινάκων 5.1 και 5.2. Για κάθε υποστηριζόμενο από τον server ΣΠ, ορίζονται στο αρχείο αυτό δύο διαφορετικές εν-

νοιολογικές εκφράσεις. Η πρώτη είναι η έκφραση ορισμού της παράγωγης έννοιας η οποία αντιστοιχεί στο ΣΠ και η δεύτερη είναι η εννοιολογική έκφραση E_P για την οποία μιλήσαμε στην ενότητα 5.4.1. Ένα δείγμα του αρχείου αυτού φαίνεται στο σχήμα 5.8, το οποίο ορίζει τις παράγωγες έννοιες που χρησιμοποιήθηκαν στα παραδείγματα αυτού του κεφαλαίου και οδηγούν στο *μέρος όψεως* του TBox του σχήματος 5.2.

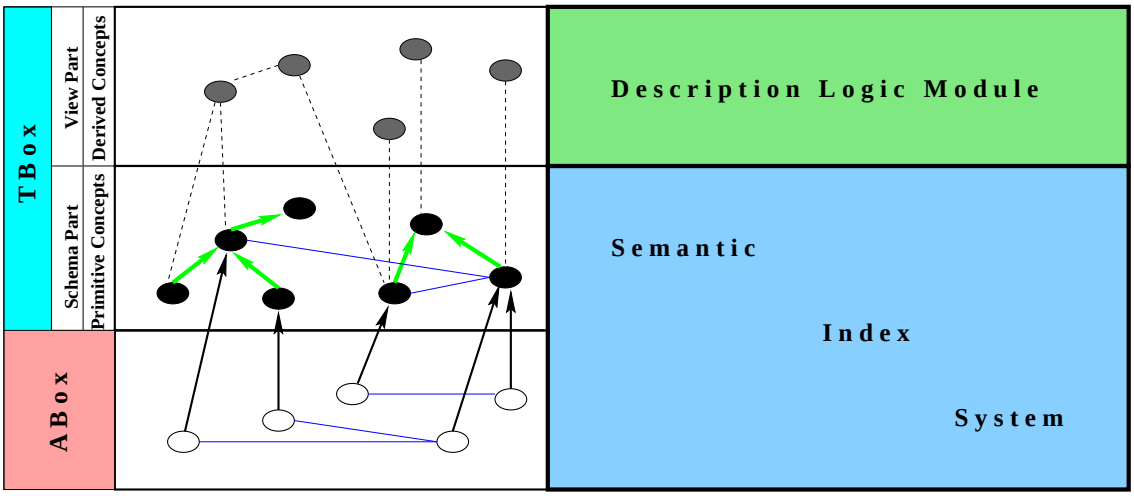
Μια παρατήρηση που μπορεί να κάνει κανείς στο αρχείο αυτό είναι ότι οι έννοιες Collection, ProtectionStatus και Any δεν έχουν αντίστοιχη έννοια E_P. Αυτό γιατί οι έννοιες τους αυτές αποτελούν ιδιαίτερες περιπτώσεις και έχουν συγκεκριμένη διαχείριση, οπότε δεν απαιτείται να ορίσει ο χρήστης κάτι γι' αυτές. Η έννοια Collection έχει αντιστοιχισθεί σε μια σταθερή τιμή. Όπως είχαμε πει η τιμή αυτή χαρακτηρίζει όλα τα αποθηκευμένα στη πηγή αντικείμενα της κεντρικής έννοιας. Έτσι αν το σύνολο C_{Res} είναι μη κενό, σημαίνει ότι η ερωτηθείσα για το ΣΠ τιμή υπάρχει και επομένως σαν έννοια E_P θα πρέπει να είναι η κεντρική έννοια C_R. Σε αντίθετη περίπτωση θα πρέπει να επιστραφεί ένα κενό σύνολο, οπότε η E_P είναι η κενή έννοια. Όσον αφορά στην ProtectionStatus, επειδή αυτή έχει αντιστοιχισθεί στην κενή έννοια, οποιαδήποτε επερώτηση σε αυτή θα δώσει σαν αποτέλεσμα το κενό σύνολο. Έτσι, η E_P των ΣΠ τα που δεν υποστηρίζονται μπορούν να αντιστοιχίζονται στην έννοια $\perp$ και η ερμηνεία τους θα είναι πάντα το κενό σύνολο. Τέλος, η έννοια Any έχει αντιστοιχηθεί στην έννοια $\top$. Η έννοια αυτή όπως έχουμε πει χρησιμοποιείται προκειμένου να γίνει μια επερώτηση σε όλα τα ΣΠ. Το σύνολο C_{Res} επομένως θα είναι ένα σύνολο από αντικείμενα από όλα πιθανώς τα ΣΠ. Έτσι, για να βρεθούν τα αντικείμενα της κεντρικής έννοιας τα οποία συσχετίζονται με αυτά σαν έκφραση E_P θα πρέπει να είναι η ένωση όλων των εκφράσεων E_P όλων των ΣΠ.

Στο σχήμα 5.7 φαίνονται τα μέρη του σχήματος 5.1, όπως έχουν υλοποιηθεί στην αρχιτεκτονική που παρουσιάστηκε.

Όσον αφορά στο τμήμα ανάκτησης δεδομένων, έχει κι αυτό μεταβληθεί. Το τμήμα αυτό (τμήμα (3)), είναι υπεύθυνο μονάχα για τη κωδικοποίηση των αποτελεσμάτων. Έχει πάψει κι αυτό να έχει άμεση επικοινωνία με το ΣΣΕ. Όταν φτάσει σε αυτό μια εντολή ανάκτησης κάποιας εγγραφής, τότε καλεί το τμήμα (7).

Το τμήμα (7) είναι υπεύθυνο για την επικοινωνία με το ΣΣΕ και την κατασκευή μιας εγγραφής. Μέσω μιας σειράς επερωτήσεων PQI ανακτά από το ΣΣΕ όλα εκείνα τα απαραίτητα δεδομένα που απαιτούνται, τα οποία τα συνθέτει και κατασκευάζει μια εγγραφή. Η αντιστοίχιση για το ποια δομή του ΣΣΕ θα μπει για κάθε στοιχείο (element) της εγγραφής είναι αποθηκευμένη στο δυαδικό κώδικα του τμήματος αυτού.

Σχετικά με την αρχιτεκτονική που παρουσιάστηκε θα μπορούσαμε να κάνουμε



Σχήμα 5.7: Τα τμήματα της ΒΓ της ΠΛ στην προτεινόμενη αρχιτεκτονική

κάποιες παρατηρήσεις. Η πρώτη είναι ότι η αντιστοίχιση των ΣΠ στις δομές της πηγής γίνεται από ένα τμήμα (το (6)), το οποίο είναι ισχυρά συνδεδεμένο με την πηγή. Μια δεύτερη παρατήρηση είναι ότι το τμήμα που κάνει την αντιστοίχιση των πεδίων της εγγραφής στις δομές της πηγής (τμήμα (7)), βρίσκεται και αυτό πάνω από το ΣΣΕ. Τέλος, θα πρέπει να σημειωθεί ότι τα τμήματα τα οποία εξαρτώνται από το είδος της πηγής και πρέπει να γνωρίζουν τον τρόπο, με τον οποίο πρέπει να επικοινωνήσουν με αυτή, είναι τα (6) και (7) μονάχα. Τα παραπάνω έχουν σαν αποτέλεσμα το τμήμα αναζήτησης να είναι εντελώς ανεξάρτητο από το είδος της πηγής, τον τρόπο επικοινωνίας με αυτήν και τον τρόπο αντιστοίχισης των ΣΠ στις δομές της, και το μόνο που έχει να κάνει είναι να μετατρέπει τις επερωτήσεις σε εννοιολογικές εκφράσεις. Το ίδιο συμβαίνει και για το τμήμα ανάκτησης όπου το μόνο που έχει να κάνει είναι να μετατρέπει τα εγγραφές σε GRS-1 μορφή. Η πηγή και τα εξαρτώμενα από αυτή τμήματα φαίνονται στο σχήμα 5.6 μέσα στη διακεκομμένη γραμμή. Η προσέγγιση αυτή παρέχει τη δυνατότητα στον server να μπορεί εύκολα να διαχειριστεί δύο ή περισσότερες, έστω και διαφορετικού είδους, πηγές πληροφορίας. Για κάθε πηγή πληροφορίας θα πρέπει να υπάρχουν τα τμήματα που βρίσκονται μέσα στη διακεκομμένη γραμμή. Το τμήμα ανάκτησης του server αφού μεταφράσει την επερώτηση σε εννοιολογική έκφραση θα πρέπει να τη στείλει σε όλες τις υποκείμενες πηγές. Κάθε πηγή θα έχει το δικό της τμήμα ΠΛ το οποίο θα κάνει τη μετάφραση σε μια επερώτηση της τοπικής πηγής και θα επιστρέψει τα αποτελέσματα. Ομοίως το τμήμα ανάκτησης, όταν λαμβάνει εγγραφές, ανεξάρτητα από που προήλθαν μπορεί και τις μετατρέπει σε GRS-1. Η αρχιτεκτονική αυτή επομένως λύνει ένα ακόμη

μεγάλο πρόβλημα του Z39.50: Το πρόβλημα της υποστήριξης πολλαπλών πηγών κάτω από έναν server, κάτι που μέχρι σήμερα ήταν δύσκολο να υλοποιηθεί δεδομένων των διαφορετικών μεθόδων πρόσβασης που έχουν οι πηγές ή των διαφορετικών τρόπων αντιστοιχίσεων ΣΠ στις δομές τους.

```
PersonalName            =  (OR Actor Owner)
EP_PersonalName         =  (OR (EXISTS ownedBy
                                       CRES)
                               (EXISTS hasEvent
                                       (EXISTS hasActor
                                               CRES)))

MaterialAndTechnique    =  (AND Material Technique)
EP_MaterialAndTechnique =  (OR (EXISTS usingTechnique
                                       CRES)
                               (EXISTS usingMaterial
                                       CRES)

Date                    =  EventDate
EP_Date                 =  (EXISTS hasEvent
                                   (EXISTS happenedIn
                                           CRES)))

DateOfCreation          =  (EXISTS (REVERSE happenedIn)
                                   (EXISTS ofKind
                                           {"Creation"})))
EP_DateOfCreation       =  EP_Date

Event                   =  EventKind
EP_Event                =  (EXISTS hasEvent
                                   (EXISTS ofKind
                                           CRES)))

Any                     =  TOP

Collection              =  {"Benaki Museum Gun Collection"}

ProtectionStatus        =  BOTTOM
```

Σχήμα 5.8: Αρχείο καθορισμού λειτουργίας μεταφραστή

5.5.2 Επεκτάσεις και σε άλλα συστήματα

Πολλές φορές είναι αναγκαία η πρόσβαση σε πληροφορίες και δεδομένα μιας πηγής (έστω βάσης δεδομένων) μέσω μιας διεπαφής ΠΛ. Σε μια τέτοια περίπτωση τίθεται το θέμα του σωστού τρόπου ολοκλήρωσης της ΒΓ με το ΣΔΒΔ. Η συνήθης προσέγγιση είναι η παράσταση του σχήματος της βάσης απευθείας σαν έννοιες στη ΒΓ. Κάτι τέτοιο είναι επιθυμητό μια και συχνά το σχήμα της βάσης μπορεί να μην ανταποκρίνεται στην εννοιολογική άποψη των χρηστών για τα δεδομένα (Το σχεσιακό μοντέλο για παράδειγμα δεν είναι πάντα και το εννοιολογικό). Για το λόγο αυτό προτείνεται η ολοκλήρωση των αρχικών εννοιών του εννοιολογικού σχήματος της βάσης με τους ορισμούς απόψεων των χρηστών [Bre96].

Η παράσταση του σχήματος μιας βάσης δεδομένων μέσω ΠΛ έχει και ορισμένα άλλα πλεονεκτήματα, όπως το ότι αποκτά την εκφραστική δυνατότητα της ΠΛ κάνοντας εφικτή των εξαγωγή και διαχείριση επιπλέον σημασιολογικών στοιχείων των δεδομένων [Bor95]. Επίσης, συντελεί στον έλεγχο συνέπειας του σχήματος, καθώς και σε όλα τα οφέλη που περιγράφηκαν στο κεφάλαιο αυτό.

Από την άλλη μεριά τα ΣΑΓ αποτυγχάνουν να διαχειριστούν μεγάλες ποσότητες δεδομένων, μια και ο σκοπός κατασκευής τους δεν είναι αυτός. Για το λόγο αυτό όταν χρειάζεται να διαχειριστεί κανείς μεγάλες ποσότητες δεδομένων είναι καλό αυτές να βρίσκονται σε άλλα συστήματα που είναι κατάλληλα για τη δουλειά της αποδοτικής αποθήκευσής τους όπως τα ΣΔΒΔ. Φαίνεται επομένως κι εδώ η ανάγκη για συνεργασία και ολοκλήρωση ΣΑΓ και ΣΔΒΔ, προκειμένου να διαχειρίζεται κανείς πολλά είδη πληροφορίας και για έναν ομοιόμορφο τρόπο ανάκτησης στοιχείων από ένα σύνθετο ολοκληρωμένο ΣΑΓ/ΣΔΒΔ.

Για την ολοκλήρωση ΣΑΓ και ΣΔΒΔ στο [Bre94] προτείνονται δύο διαφορετικοί τρόποι: Η *ασθενής διασύνδεση (loose coupling)*, σύμφωνα με την οποία τα δεδομένα ανακτούνται μαζικά από την βάση κατά την εκκίνηση της εφαρμογής, και έπειτα η διαδικασία συνεπαγωγών και απάντησης επερωτήσεων συνεχίζεται μονάχα στη ΒΓ. Το μειονέκτημα της μεθόδου αυτής είναι ότι είναι δύσκολος ο χειρισμός του μεγάλοι όγκοι πληροφοριών καθώς επίσης και το ότι η ΒΓ δεν γνωρίζει την εκάστοτε κατάσταση της βάσης δεδομένων παρά μόνο αυτή που είχε όταν φορτώθηκαν τα δεδομένα.

Από την άλλη μεριά υπάρχει η *ισχυρή διασύνδεση (tight coupling)* σύμφωνα με την οποία μόνο το εννοιολογικό σχήμα της βάσης αποθηκεύεται στη ΒΓ και τα δεδομένα ανακτώνται από την πηγή μόνο όταν χρειαστεί. Έτσι στο [Bre94] προτείνεται ένα νέο τμήμα το, DBox. Το DBox είναι ό,τι και το ABox με τη διαφορά ότι δεν βρίσκεται στη ΒΓ

αλλά σε μια εξωτερική πηγή δεδομένων.

Η βιβλιογραφία έχει να επιδείξει αρκετές αναφορές σε εφαρμογές όπου γίνεται μια προσπάθεια να εφαρμοστεί ένα επίπεδο ΠΛ, και γενικότερα βάσης γνώσης, πάνω από ένα ΣΔΒΔ. Παραδείγματα αποτελούν η COOCOON [RLNR95, NRL+94], η C3L++ [KSS+96, KSS95], η LOOM [Bre95b], και άλλες [GBBI96].

Η αρχιτεκτονική που προτάθηκε είναι μια ανοικτή αρχιτεκτονική. Τμήματα αυτής μπορούν να αντικαθίστανται εύκολα, ή να προστίθενται νέα. Η χρησιμοποίηση του ΣΣΕ ως αποθηκευτικού μέσου δεδομένων, δεν είναι περιοριστική. Τα δεδομένα θα μπορούσαν να βρίσκονται σε μια εξωτερική πηγή. Αυτό που θα απαιτούνταν θα ήταν μόνο η εισαγωγή και παράσταση του εννοιολογικού σχήματος της πηγής στο ΣΣΕ το οποίο θα έπαιζε τον ρόλο του *μέρους σχήματος* του TBox. Πάνω από αυτό θα συνέχιζε να υπάρχει το τμήμα της ΠΛ ως *μέρος όψεως* του TBox. Αντί τα δεδομένα να είναι σε μια μόνο πηγή θα μπορούσαν να είναι σε δύο ή περισσότερες. Η εφαρμογή (ΠΛ + ΣΣΕ) είναι σε θέση να χειριστεί μια τέτοια περίπτωση. Έχουμε επομένως μια εφαρμογή, η οποία γνωρίζει τα εννοιολογικά σχήματα διαφόρων πηγών, επομένως και το περιεχόμενό τους, μπορεί να επικοινωνεί και να ανακτά δεδομένα από αυτές και τέλος να δέχεται επερωτήσεις σε εννοιολογική γλώσσα και να τις μετατρέπει και να τις στέλνει στις κατάλληλες πηγές αφού κάνει κάποιους έξυπνους συλλογισμούς. Συνεπώς, ο μεταφραστής ο οποίος προτείνεται είναι περισσότερο ”έξυπνος” από άλλους μεταφραστές, και μπορεί να επιτελέσει και λειτουργίες *διαμεσολαβητή* όπως μόλις αποδείχθηκε.

Κεφάλαιο 6

Επίλογος

6.1 Συμπεράσματα

Το θέμα της εργασίας αυτής ήταν οι εξυπηρέτες Z39.50. Παρουσιάστηκε η υλοποίηση ενός Z39.50 εξυπηρέτη και εκτέθηκαν τα προβλήματα που αντιμετωπίστηκαν κατά την κατασκευή και εφαρμογή του πάνω από δύο βάσεις δεδομένων, μία με μουσειακά αντικείμενα και μια με κείμενα SGML.

Στο κεφάλαιο 2 έγινε μια συνοπτική παρουσίαση του Z39.50 πρωτοκόλλου και των λειτουργιών του.

Στο κεφάλαιο 3 εισήχθη το πρόβλημα της μετάφρασης και έγινε μια περιγραφή του τρόπου με τον οποίο διάφορα έργα αντιμετωπίζουν το πρόβλημα αυτό, είτε χρησιμοποιώντας το Z39.50 πρωτόκολλο είτε όχι, σε διάφορα συστήματα όπως βιβλιογραφικές βάσεις δεδομένων, ή Συστήματα Ολοκλήρωσης Πληροφοριών (π.χ. TSIMMIS), κλπ.

Στο κεφάλαιο 4 παρουσιάστηκε ο Z39.50 εξυπηρέτης που κατασκευάστηκε στα πλαίσια αυτής της εργασίας. Ο εξυπηρέτης αυτός χρησιμοποιήθηκε σε δύο μεγάλα έργα το AQUARELLE και το CIMIzit με επιτυχία. Παράλληλα όμως ανέδειξε μια σειρά προβλημάτων. Έγινε έτσι φανερή η ανάγκη για την ύπαρξη κάποιου ενδιάμεσου επιπέδου ανάμεσα στο τμήμα του εξυπηρέτη που επικοινωνεί με το δίκτυο και της υποκείμενης πηγής. Το τμήμα αυτό θα είναι υπεύθυνο για τη μετάφραση των επερωτήσεων από τις δομές και τους τελεστές του Z39.50 στις δομές και τελεστές της πηγής. Κρίθηκε επίσης πολύ χρήσιμη η ύπαρξη κάποιου τρόπου για το δηλωτικό προσδιορισμό της λειτουργίας του.

Στο κεφάλαιο 5 εισήχθη η ΠΛ, η οποία παρέχει μια εννοιολογική γλώσσα με την οποία θα μπορούσε κανείς να εκφράσει την εννοιολογική αντιστοίχιση μεταξύ των

δομών του Z39.50 και των δομών της πηγής. Έτσι παρουσιάστηκαν αρχικά τα βασικά χαρακτηριστικά της ΠΛ και έπειτα ο τρόπος με τον οποίο η θεωρία αυτή μπορεί να εφαρμοστεί στο Z39.50. Η εφαρμογή της θεωρίας αυτής έδωσε εκτός των άλλων και μια σειρά από επιπλέον δυνατότητες, οι οποίες συντελούν στην αντιμετώπιση ορισμένων προβλημάτων του Z39.50. Αν και, λόγω έλλειψης χρόνου, δεν έγινε υλοποίηση της ιδέας αυτής, φαίνεται να είναι αρκετά αποδοτική. Στο τέλος του κεφαλαίου έγινε μάλιστα μια πρόταση για τον τρόπο με τον οποίο το σύστημα που παρουσιάστηκε στο κεφάλαιο 3 μπορεί να αλλαχθεί ώστε να εφαρμοστεί σε αυτό η ΠΛ.

Η ΠΛ γενικά αποδείχθηκε αρκετά κατάλληλη για τον προσδιορισμό της λειτουργίας του εξυπηρέτη Z39.50. Αυτό γιατί παρέχει:

- μια σαφή σημασιολογία των τελεστών της, πράγμα που οδηγεί στην εύκολη μετάφραση από το μοντέλο του Z39.50 στο μοντέλο της ΠΛ,
- το δηλωτικό προσδιορισμό της λειτουργίας του μεταφραστή μέσω μιας γλώσσας η οποία είναι αρκετά εκφραστική και αρκετά εύκολη στη χρήση μια και είναι πολύ κοντά στη φυσική γλώσσα του χρήστη, και τέλος
- παρέχει ένα πλαίσιο για την αντιμετώπιση μιας σειράς προβλημάτων του Z39.50.

6.2 Μελλοντικές κατευθύνσεις

Για την πλήρη ολοκλήρωση της παρούσας δουλειάς υπάρχει ένας αριθμός από εργασίες οι οποίες θα μπορούσαν να γίνουν στο μέλλον.

Το πρώτο και βασικό που θα έπρέπε να γίνει είναι η μετατροπή του Z39.50 εξυπηρέτη ο οποίος περιγράφηκε στο κεφάλαιο 4 ώστε να χρησιμοποιεί την ΠΛ για τον δυναμικό καθορισμό της λειτουργίας του. Ο τρόπος με τον οποίο μπορεί να γίνει κάτι τέτοιο έχει περιγραφεί στην ενότητα 5.5.

Επόμενο βήμα θα ήταν η επέκταση της υλοποίησης ώστε να χρησιμοποιεί όλες τις δυνατότητες συλλογισμών της ΠΛ. Στην πρόταση που έγινε, μόνο ο έλεγχος περιπτώσεως είναι αυτός που χρησιμοποιείται μια και αυτό που ενδιέφερε είναι η απάντηση επερωτήσεων. Ωστόσο, και οι άλλοι συλλογισμοί μπορούν να χρησιμοποιηθούν για πιο αποδοτική ανάκτηση πληροφορίας. Για παράδειγμα, η ικανότητα της ΠΛ να καταλαβαίνει πότε μια έννοια υπάγεται σε μια άλλη καθώς και πότε έχει κενή ερμηνεία, μπορεί να χρησιμοποιηθεί για να γίνεται απάντηση επερωτήσεων με χρήση μόνο του σχήματος, χωρίς δηλαδή πρόσβαση στα δεδομένα (intensional query answering).

Σημαντικό ζήτημα επίσης, είναι μια θεωρητική μελέτη της πολυπλοκότητας των συλλογισμών. Θα πρέπει δηλαδή να μελετηθούν πιο συστηματικά οι δυνατότητες των εννοιολογικών γλωσσών καθώς και των συλλογισμών που μπορεί κανείς να επιτελέσει με αυτές. Κάτι τέτοιο θα αποτελεί ένα βασικό κριτήριο για τη χρησιμοποίησή τους ή όχι, καθώς και τον τρόπο με τον οποίο θα πρέπει να γίνει αυτό. Για παράδειγμα, με γνώση της πολυπλοκότητας των τελεστών και των συλλογισμών θα μπορεί κάποιος να αποφασίσει ποιους τελεστές να χρησιμοποιήσει και ποιους να αποφύγει. Θα πρέπει να σημειωθεί ότι η εκφραστικότητα και η πολυπλοκότητα είναι αλληλένδετες έννοιες στην ΠΛ. Όσο πιο πλούσιο είναι το σύνολο των τελεστών που χρησιμοποιεί η εννοιολογική γλώσσα τόσο μεγαλύτερη η εκφραστικότητά της, παράλληλα όμως μεγαλώνει και η πολυπλοκότητα των συλλογισμών.

Η παρούσα ή οι μελλοντικές υλοποιήσεις θα πρέπει να δοκιμαστούν και σε άλλα συστήματα αποθήκευσης πληροφοριών. Προς το παρόν σαν πηγή δεδομένων έχει χρησιμοποιηθεί μονάχα το ΣΣΕ. Θα ήταν αρκετά χρήσιμο να γίνουν ορισμένες δοκιμές της απόδοσης και γενικότερα της λειτουργικότητας του συστήματος σε περιπτώσεις ΣΑΓ, Βάσεων Δεδομένων (σχεσιακών ή οντοκεντρικών), καθώς και σε ΣΑΠ.

Τελειώνοντας, προτείνεται η ολοκλήρωση και ενσωμάτωση των μηχανισμών συλλογισμού και εξαγωγής συμπερασμάτων της ΠΛ στο ΣΣΕ ώστε να αποτελεί ένα ΣΠΓ με τις ικανότητες αυτές, πράγμα που θα του δώσει τη δυνατότητα να χρησιμοποιείται πάνω από άλλες πηγές πληροφοριών προκειμένου να παρέχει μια διεπαφή βάσης γνώσης προς αυτές. Μετά μια τέτοια ολοκλήρωση, το ΣΣΕ, θα ήταν δυνατό να αποτελέσει μια αυτόνομη εφαρμογή η οποία θα μπορεί να λειτουργήσει σαν διαμεσολαβητής σε ένα Σύστημα Ολοκλήρωσης Πληροφοριών (Information Intagration System).

Παράρτημα Α

Συντομογραφίες

ΠΛ	Περιγραφική Λογική
ΣΠ	Σημεία Πρόσβασης
ΣΣΕ	Σύστημα Σημασιολογικού Ευρετηριασμού
ΣΔΒΔ	Σύστημα Διαχείρισης Βάσεων Δεδομένων
ΣΑΠ	Σύστημα Ανάκτησης Πληροφοριών
ΣΟΠ	Σύστημα Ολοκλήρωσης Πληροφορίας
ΣΠΓ	Σύστημα Παράστασης Γνώσης
ΣΒΓ	Σύστημα Βάσης Γνώσης

Παράρτημα Β

Ευρετήριο όρων

Περιγραφική Λογική	Description Logic
Σημεία Πρόσβασης	Access Points
Σύστημα Ανάκτησης Πληροφοριών	Information Retrieval System
Σύστημα Βάσης Γνώσης	Knowledge Base System
Σύστημα Διαχείρισης Βάσεων Δεδομένων	DataBase Management System
Σύστημα Ολοκλήρωσης Πληροφορίας	Information Integration System
Σύστημα Παράστασης Γνώσης	Knowledge Representation System
Σύστημα Σημασιολογικού Ευρετηριασμού	Semantic Index System
Ψηφιακή Βιβλιοθήκη	Digital Library
ακρίβεια	precision
ανάκληση	recall
αποκοπή	truncation
αρχικές	primitive
ατομικός, -η, -ο	atomic
δήλωση	statement
δηλωτικός	declarative
διάταξη	relation
εγγραφή	record
εκτασιακός	extensional

ελεύθερες συμφραζομένων γραμματικές	context free grammars
εννοιολογική γλώσσα	concept language
εντασιακός	intensional
επέκταση	extension
επίπεδο εφαρμογών	application layer
ερμηνεία	interpretation
ετερογενείς πληροφορίες	heterogeneous information
ευρετηριασμός πλήρους κειμένου	full text indexing
ισοδύναμες	equivalent
καθολική σχέση	universal relation
μέρος όψεως	view part
μεταφραστής	wrapper
όψη	view
ορισμένη	defined
παράγωγος, -η, -ο	derived
πεδίο	domain
περίπτωση	instance
προφίλ	profile
συλλογισμός	reasoning
σύμπαν αναφοράς	universe of discourse
τμήμα σχήματος	schema part
υπάγεται	is subsumed by
υπαγωγή	subsumption
υπόδειγμα	template

Βιβλιογραφία

[ABN92] T. M. Anwar, H. W. Beck, and S. B. Navathe. Knowledge Mining by Imprecise Querying: A Classification-based System. In *Proceedings of the International Conference on Data Engineering*, pages 622--630, Tempe, AZ, February 1992.

[AK97] N. Ashish and C. Knoblock. Semi-automatic Wrapper Generation for Internet Information Sources. In *Proceedings of Second IFCIS International Conference on Cooperative Information Systems (CoopIS'97)*, South Carolina, USA, June 1997.

[ANS95] ANSI/NISO. Z39.50-1995 (Versions 2 and 3) Information Retrieval: Application Service Definition and Protocol Specification, 1995.

[AQU] Project AQUARELLE. http://aqua.inria.fr/. 1996-1998.

[BB93] A. Borgida and R. J. Brachman. Loading data into description reasoners. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 22(2):217--226, June 1993.

[BBH+91] F. Baader, H. Bürckert, J. Heinsohn, B. Hollunder, J. Müller, B. Nebel, W. Nutt, and H. Profitlich. Terminological Knowledge Representation: A proposal for a Terminological Logic. In Nebel B., Luck K. von, and Peltason C., editors, *International Workshop on Terminological Logics*, Dagstuhl, Germany, 1991. DFKI.

[BBS88] S. Bergamaschi, F. Bonfatti, and C. Sartori. ENTITY-SITUATION: A Model for the Knowledge Representation Module of a KBMS. In Missikoff M. Schmidt J. W., Ceri S. , editor, *Proceedings of the International Conference on Extending Database Technology (EDBT '88)*, volume 303 of *LNCS*, pages 578--582, Venice, Italy, March 1988. Springer.

[BDH97] V. Bouthors, J. Dupuis, and N. T. Huu. Z39.50 Gateway for Mistral and ARF DTD Specification. Aquarelle project, deliverable 5.2, INRIA, France, September 1997.

[BDNS94] M. Buchheit, F. Donini, W. Nutt, and A. Schaerf. Terminological Systems Revisited: Terminology=Schema + Views. In *Proceedings of KI'94 Workshop KRDB'94*, Saarbrócken, Germany,, September 20-22, 1994.

[BF98] R. Bull and S. Finnnigan. Z+SQL - Draft 5 Review. Proposal to the Z+SQL Meeting in Orlando, http://www.dstc.edu.au/DDU/projects/ZINC/zsql5_review, January 1998.

[BHN+93] F. Baader, B. Hollunder, B. Nebel, H. Profitlich, and E. Franconi. An Empirical Analysis of Optimization Techniques for Terminological Representation Systems. Research Report RR-93-03, Deutsches Forschungszentrum für Künstliche Intelligenz, Germany, 1993.

[BM95] U. Badertscher and R. Marti. Integrating Terminological and Deductive Reasoning. Technical Report 1995TR-229, Swiss Federal Institute of Technology, Zurich, March, 1995.

[BMS+91] R. Brachman, D. McGuinness, P. P. Schneider, L. A. Resnick, and A. Borgida. Living with CLASSIC: When and How to Use a KL-ONE-like Language. In John F. Sowa, editor, *Principles of Semantic Networks --- Explorations in the Representation of Knowledge*, pages 401--456. Morgan Kaufmann, 1991.

[Bor92a] A. Borgida. From tuple systems to knowledge representation: Natural semantics specification logics. Rutgers University, January 1992.

[Bor92b] A. Borgida. Towards the Systematic Development of Description Logic Reasoners: CLASP reconstructed. In Bernhard Nebel, Charles Rich, and William Swartout, editors, *KR'92: Principles of Knowledge Representation and Reasoning*, pages 259--269. Morgan Kaufmann, San Mateo, CA, 1992.

[Bor94] A. Borgida. Description Logics for Querying Databases. In *Proceedings of the International Workshop on Description Logics*, Bonn, Germany, 1994.

[Bor95] A. Borgida. Description Logics in Data Management. *IEEE Transactions on Knowledge and Data Engineering*, 7(5):671--682, October 1995.

[Bor96] A. Borgida. On the Relative Expressiveness of Description Logics and Predicate Logics. *AIJ: Artificial Intelligence*, 82(1-2):353--367, April 1996.

[Bre94] P. Bresciani. Uniformly Querying Knowledge Bases and Data Bases. In *Proceedings of KI'94 Workshop KRDB'94*, Saarbrócken, Germany, September 20-22, 1994.

[Bre95a] P. Bresciani. Querying Databases from Description Logics. In *Proceedings of 2nd Workshop KRDB'95*, Bielefeld, Germany, September 11-12, 1995.

[Bre95b] P. Bresciani. Querying Databases from Description Logics. In *Proceedings of 2nd Workshop KRDB'95*, Bielefeld, Germany, September11-12, 1995.

[Bre96] P. Bresciani. Some research trends in KR&DB. In *Proceedings of the 3rd Workshop KRDB'96*, Budapest, Hungary, August13 1996.

[BS85] R. J. Brachman and J. G. Schmolze. An Overview of the KL-ONE Knowledge Representation System. *Cognitive Science*, 9(2):171--216, March 1985.

[BSV95] S. Bergamasci, C. Sartori, and M. Vincini. DL Techniques for Intensional Query Answering in OODBs. In *Proceedings of 2nd Workshop KRDB'95*, Bielefeld, Germany, September11-12, 1995.

[CCP+97] V. Christophides, P. Carrara, Marco Padula, M. Scholl, and A. Vercoustre. Querying Heterogeneous Data in Aquarelle project. AQUARELLE Project. Deliverable 4.6, October 1997.

[CD93] P. Constantopoulos and M. Doerr. The SIS System: A Brief Presentation. ICS-FORTH, http//www.csi.forth.gr/isst, May 1993.

[CGL96] Diego Calvanese, Giuseppe De Giacomo, and Maurizio Lenzerini. Representing sgml documents in description logics. In *Proceedings of the 1996 Description Logic Workshop (DL-96)*, pages 102--106, 1996.

[CGMP96] C. K. Chang, H. García-Molina, and A. Paepcke. Boolean Query Mapping Across Heterogeneous Information Sources. *IEEE Transactions on Knowledge and Data Engineering*, 8(4):515--521, August 1996.

[Chr94] M. Christoforaki. Caltural documentation with system CLIO. Master science thesis, Computer Science Department, University of Crete, Greece, February 1994.

[CL93] T. Catarci and M. Lenzerini. Representing and using interchema knowledge in cooperative information systems. *International Journal of Intelligent and Cooperative Information Systems*, 2(4):375--398, 1993.

[con] The CIMI consortium. http://www.cimi.org/.

[DD95] H. Darwen and C. J. Date. The third manifesto (databases). *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 24(1):39--49, March 1995.

[DGS87] B. C. Desai, P. Goyal, and F. Sadri. Non-First Normal Form universal relations: an application to information retrieval systems. *Information Systems*, 12(1):49--55, 1987.

[DLNN94] F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. Queries, Rules and Definitions as Epistemic Sentences in Concept Languages. *Lecture Notes in Computer Science*, 810:113, 1994.

[DLNN95] F. M. Donini, M. Lenzerini, D. Nardi, and W. Nutt. The Complexity of Concept Languages. Research Report RR-95-07, Deutsches Forschungszentrum für Künstliche Intelligenz, Kaiserslautern, Germany, 1995.

[DLNS96] F. D. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasonig in Description Logics. In G. Brewka, editor, *Principles of Knowledge Representation and Reasoning*, Studies in Logic, Language and Information, pages 193--238. CLSI Publications, 1996.

[FC] S. Finnigan and R. Colomb. Protocol Support for Commencial Access to Complex Database Applications. Distributed Systems Technology Center Pty Ltd and School of Information Technology, The University of Queensland.

[GBBI96] A. Goni, J. Bermudez, J. M. Blanco, and A. Illarramendi. Using Reasoning of Description Logics for Query Processing in Multidatabase Systems. In *Proceedings of 3st Workshop KRDB'96*, Budapest, Hungary, August 13, 1996.

[GMHI+95] H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Ullman, , and J. Widom. Integrating and Accessing Heterogeneous Information Sources in TSIMMIS. In *Proceedings of the AAAAI Symposium on Information Gathering*, pages 61--64, Stanford, California, March 1995.

[GMPQ+97] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagiv, J. D. Ullman, V. Vassalos, and J. Widom. The TSIMMIS Approach to Mediation: Data Models and Languages. *Journal of Intelligent Information Systems (JIIS)*, 8(1):117--132, January 1997.

[GR90] C. F. Goldfarb and Y. Rubinsky. *The SGML handbook*. Clarendon Press, Oxford, UK, 1990.

[HKNP94] J. Heinsohn, D. Kudenko, B. Nebel, and H. Profitlich. An Empirical Analysis of Terminological Representation Systems. *Artificial Intelligence*, 68(2):367--397, 1994.

[Hus94] U. Hustadt. Do we need the closed world assumption in knowledge representation? In *Proceedings of KI'94 Workshop KRDB'94*, Saarbrócken, Germany, September 20-22, 1994.

[Ind97] Index Data, http://www.indexdata.dk/yaz. *Yaz User's Guide and Reference Manual*, version 1.4 edition, 1997.

[Ins97] Institute of Computer Science, Foundation of Research and Technology, Hellas (ICS/FORTH). *SIS - Programmatic Query Interface: Reference Manual*, v2.2 edition, 1997.

[Jeu94] M. A. Jeusfeld. Generating Queries from Complex Type Definitions. In *Proceedings of KI'94 Workshop KRDB'94*, Saarbrócken, Germany, September 20-22, 1994.

[KSS95] T. Kessel, M. Schlick, and O. Stern. Accessing Configuration-Databases by means of Description Logics. In *Proceedings of 2nd Workshop KRDB'95*, Bielefeld, Germany, September 11-12, 1995.

[KSS+96] T. Kessel, M. Schlick, H.-M. Speiser, U. Brinkschulte, and H. Vogelsang. C3L+++: Implementing a Description Logics System on Top of an Object-Oriented Database System. In *KRDB'96*, volume 4, Budapest, Hungary, August13 1996.

[LBG97] A. Ludwig, P. Becker, and U. Gntzer. Interfacing Online Bibliographic Databases with Z39.50. In *Proceedings of International Database Engineering and Applications Symposium (IDEAS)*, Montreal, Canada, 1997.

[LP92] H. R. Lewis and C. Papadimitriou. *Elements of the Theory of Computation*. Technical Chamber of Greece, Publications Bureau, 1992.

[LR96] A. Y. Levy and M. Rousset. Using Description Logics to Model and Reason About Views. In W. Wahlster, editor, *ECAI 96. 12th European Confere4nce in Artificial Intelligence*. John Wiley & Sons, Ltd., 1996.

[Ltd97] System Simulation Ltd. Aquarelle Z39.50 Profile. Working Draft 1.15, System Simulation Ltd., http://aqua.inria.fr, June 1997.

[Mad96] S. E. Madnick. Are We Moving Toward an Information SuperHighway or a Tower of Babel? The Challenge of Large-Scale Semantic Heterogeneity. In *Proceedings of the Twelfth International Conference on Data Engineering*, pages 2--8, New Orleans, Louisiana, February 1996.

[MBJK90] J. Mylopoulos, A. Borgida, M. Jarke, and M. Koubarakis. Telos: Representing Knowledge About Information Systems. *ACM Transactions on Information Systems*, 8(4):325--362, 1990.

[Moe95] W. E. Moen. A Guide to the ANSI/NISO Z39.50 Protocol: Information Retrieval in the Information Infrastructure. NISO Press, http://www.cni.org/pub/NISO/docs/Z39.50-brochure/50.brochure.toc.html, 1995.

[MS96] C. Meghini and U. Straccia. Information Retrieval: Foundations of a Description Logic Based Approach. Technical report, Istituto di Elaborazione della Informazione (IEI-CNR), December 20, 1996.

[MSS98] U. Meghini, C. Sebastiani, and F. Straccia. MIRLOG: A Logic for Multimedia Information Retrieval. Consiglio Nazionale delle Ricerche, Istituto di Elaborazione dell' Informazione, 1998.

[Neb91] B. Nebel. Terminological Cycles: Semantics and Computational Properties. In John F. Sowa, editor, *Principles of Semantic Networks*, pages 331--362. Morgan Kaufmann, San Mateo, 1991.

[NRL+94] M. C. Norrie, U. Reimer, P. Lippuner, Michael Rys, and Hans-J. Schek. Frames, Objects and Relations: Three Semantic Levels for Knowledge Base Systems. In *Proceedings of KI'94 Workshop KRDB'94*, Saarbrócken, Germany, September 20-22, 1994.

[oC] Library of Congress. Z39.50 Maintenance Agency Home Page. http://lcweb.loc.gov/z3950/agency.

[PGMGU95] Y. Papakonstantinou, H. Garcia-Molina, A. Gupta, and J. Ullman. A Query Translation Scheme for Rapid Implementation of Wrappers. In *Fourth International Conference on Deductive and Object-Oriented Databases*, pages 161--186, National University of Singapore(NUS), Singapore, 1995.

[proa] The CIMI profile. http://www.cimi.org/products/cimi_products.html. 1998.

[prob] The CIMI Interoperability Testbed project. http://www.cimi.org/projects/.

[PSS93] P. F. Patel-Schneider and B. Swartout. Description Logic Specification from the KRSS Effort. Working Version (Draft), June 1993.

[RLNR95] U. Reimer, P. Lippuner, M. Norrie, and M. Rys. Terminological Reasoning by Query Evaluation: A Formal Mapping of a Terminological Logic to an Object Data Model. Technical Report 1995PA-rlnr95-kruse95, Swiss Federal Institute of Technology, Zurich, August, 1995.

[Sch94] A. Schaerf. *Query Answering in Concept-Based Knowledge Representation Systems: Algorithms, Complexity, and Semantic Issues.* Ph.D. thesis, Dipartimento di Informatica e Sistemistica Univerita di Roma: La Sapienza, 1994.

[SL97] O. Signore and M. Loffredo. Z39.50-SQL gateways: Technical description. Aquarelle project, deliverable 5.1, CNR-CNUCE, Italy, April 1997.

[SSL97] SSL. Z39.50 version of Index+: Technical Description. Aquarelle project, deliverable 5.3, System Simulation Ltd, UK, October 1997.

[STM95] I. Savnik, Z. Tari, and T. Mohoric. Using Schema Information for Querying Databases. In *Proceedings of 2nd Workshop KRDB'95*, Bielefeld, Germany, September11-12, 1995.

[Ull82] J. D. Ullman. The Universal Relation as a User Interface. In *Principles of Database Systems*, pages 317--348. Computer Science Press, 1982.

[Wil95] M. H. P. Wilhelmus. *Selecting Knowledge Representation Systems.* PhD thesis, Universiteit Twente Enschede, 1995.